Expert Systems With Applications 298 (2026) 129853

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

CoSemiGNN: Blockchain fraud detection with dynamic graph neural networks based on co-association of semi-supervised

Yulong Wang [a], Qingxiao Zheng [a,b,c,d,*], Xuedong Li [a,c,d], Lingfeng Wang [a,c,d], Ling Lin [a,c,d]

[a] College of Artificial Intelligence(CUIT Shuangliu Industrial College), Chengdu University of Information Technology, Chengdu, 610225, China
[b] School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, 611130, China
[c] Advanced Cryptography and System Security Key Laboratory of Sichuan Province, CUIT Shuangliu Industrial College, Chengdu, 610225, China
[d] National Intelligent Society Comprehensive Governance Experimental Base, CUIT Shuangliu Industrial College, Chengdu, 610225, China

ARTICLE INFO

Keywords:
Blockchain abnormal transaction detection
Graph neural network
Semi-supervised learning
Ensemble learning

ABSTRACT

With the development of blockchain technology, the increasing number of cyber frauds has caused huge economic losses, prompting more and more researchers to focus on how to effectively detect criminal activities in the blockchain transaction environment. Currently, graph neural network (GNN)-based methods have made significant progress in the field of blockchain illegal transaction detection due to their advantages in extracting graph structure features. However, existing illegal transaction pattern detection methods usually rely on historical labeled data. In the blockchain transaction environment, transaction data changes over time, and it is often difficult to obtain transaction labels. As a result, the performance of these methods is often unsatisfactory when faced with newly distributed transaction data. To address this challenge, this paper proposes a dynamic graph neural network based on co-association of semi-supervised (CoSemiGNN) for more efficiently identifying illegal transactions in blockchain environments under conditions of dynamically changing transaction data. The model combines semi-supervised learning with a dynamic graph neural network, enabling it to effectively identify novel illegal transaction patterns from unlabeled data and adapt to the evolving blockchain network environment. Specifically, CoSemiGNN captures features of novel transactions by integrating semi supervised learning results. It utilizes co-occurrence relations of edges and co-occurrence feature aggregation of nodes to skillfully integrate semi-supervised methods into feature extraction of transaction graphs, enabling the model to extract novel illegal transaction patterns from unlabeled data. In addition, the model utilizes self attention recurrent neural networks (RNNs) to capture temporal information in transactions, ensuring the dynamics of CoSemiGNN. Finally, we theoretically analyze the model, and experiments on a real Bitcoin transaction dataset demonstrate that CoSemiGNN outperforms existing methods by as much as 30 % in terms of F1 scores for detecting illegal transactions when the transaction data undergoes distributional migration. This research compensates the problem that existing methods ignore the distributional changes of blockchain transaction data, and provides a new perspective and an effective solution for blockchain illegal transaction detection.

1. Introduction

Since Bitcoin was first proposed in 2008, the underlying blockchain technology has evolved into one of the most revolutionary innovations in fintech (Nakamoto, 2008). With characteristics such as decentralization, immutability, and transaction transparency, blockchain offers novel solutions for a variety of domains including digital currency transactions, smart contract execution, and supply chain management. However, the distinctive features of decentralization and transaction anonymity inherent in blockchain systems have also made them one of the most commonly used payment tools by criminals (Holub & O'Connor, 2018). In the past, anonymous darknet markets such as Silk Road extensively adopted Bitcoin as a means of payment, significantly fueling the proliferation of drug trafficking, money laundering, and other serious criminal activities (Christin, 2013). New-generation darknet platforms like Hydra and Black Market Reloaded remain active on the Tor

* Corresponding author at: College of Artificial Intelligence(CUIT Shuangliu Industrial College), Chengdu University of Information Technology, Chengdu, 610225, China.
E-mail addresses: 3231903007@stu.cuit.edu.cn (Y. Wang), zqx@cuit.edu.cn (Q. Zheng), xuedongl@cuit.edu.cn (X. Li), wanglf@cuit.edu.cn (L. Wang), linling@cuit.edu.cn (L. Lin).

https://doi.org/10.1016/j.eswa.2025.129853
Received 25 February 2025; Received in revised form 11 August 2025; Accepted 22 September 2025
Available online 27 September 2025

network, continuing to leverage crypto assets for illicit transactions. According to the 2024 Internet Crime Report published by the Internet Crime Complaint Center (IC3) of the Federal Bureau of Investigation (FBI), economic losses due to cryptocurrency fraud in the United States reached approximately USD 9.3 billion in 2024, representing a 66 % increase compared to 2023 (Investigation, 2025). The 2024 Crypto Crime Report by Chainalysis indicates that global crypto-related scams exceeded USD 14.9 billion in 2024 (Team, 2025). These phenomena underscore the inadequacy of existing regulatory measures and fund-tracking mechanisms in the blockchain transaction environment, due to the inherent decentralization, transactional anonymity, and cross-chain liquidity of such systems. they highlight the urgency of detecting and preventing phishing scams within the blockchain ecosystem. The United Nations Office on Drugs and Crime (UNODC), in its report, has called on national governments and regulatory authorities to strengthen cross-border regulatory collaboration over blockchain transactions (Drugs & Crime, 2025). As core nodes in the circulation of digital currencies, exchanges play a critical role in identifying suspicious transactions. Although many mainstream trading platforms-such as Binance, Coinbase, and OKX-have deployed on-chain analytics tools to assist in monitoring anomalous transactions, traditional tools generally rely on static blacklists or rule-based templates, which are insufficient in detecting emerging fraud patterns such as cross-chain laundering and multi-hop coin mixing. This situation underscores the urgent need for more effective automated detection methods (Elliptic, 2023; Motamed & Bahrak, 2019).

With the rapid development of blockchain technology and the widespread adoption of crypto-assets, related criminal activities have evolved to exhibit new features of intelligence and concealment. However, existing detection systems show a marked lack of adaptability when confronted with these emerging forms of criminal behavior, which further underscores the urgency of constructing intelligent and robust monitoring frameworks for illicit transactions. Against this backdrop, the detection of illicit transactions on blockchain has increasingly become a focal point of research in recent years, giving rise to a large number of modeling approaches and algorithmic frameworks tailored to this task. These studies have made notable progress in modeling paradigms, algorithmic performance, and the analysis of blockchain data, thus advancing the development of the field. Nevertheless, due to the highly complex structural characteristics of blockchain transaction data and its dynamic nature in real-world environments, the practical application of such methods still faces several key challenges:

1. **Graph-structured complexity of transaction data.** Blockchain transaction data naturally exhibits a complex graph structure, where nodes represent transaction addresses and edges denote fund flows, forming a typical transactional graph. This structure differs substantially from traditional tabular data or static social networks and is characterized by high diversity, interconnectivity, and hierarchy (Christin, 2013; Nakamoto, 2008). For instance, a single illicit transaction may involve multiple intermediary addresses with multi-hop fund transfers, or even construct a “money laundering path” to form an anonymous transaction chain (Bellei et al., 2024). These complexities render simple classification models based on isolated samples inadequate for capturing the intricate dependencies underlying transaction behaviors. There is thus a pressing need for graph-based modeling approaches capable of revealing suspicious behavioral patterns hidden within large-scale transaction networks.
2. **Dynamic evolution of blockchain transaction graphs.**

   Structural dynamics:In real-world scenarios, the structure of transaction graphs evolves continuously over time. New transaction addresses frequently appear, while transactional relationships are constantly being established or terminated, resulting in a constantly shifting graph topology. For example, new addresses participate in transactions daily, and the pathways of fund transfers change accordingly (Mohan et al., 2023; Pareja et al., 2020; Wang et al., 2024b; Xiao et al., 2023; Zheng, 2022). According to the existing research, the annual growth rate of edges and nodes in the Bitcoin environment is 11.20 % and 11.20 %, respectively. In Ethereum, the annual growth rate of edges is as high as 20.17 %, and the annual growth rate of nodes is as high as 19.42 % (Motamed & Bahrak, 2019).

   Feature dynamics:The attributes of nodes and edges, such as transaction frequency, amount, and interaction time-also exhibit significant temporal volatility. Events with a major impact on illicit transaction patterns (e.g., the takedown of darknet marketplaces) have been observed to often lead to abrupt shifts in the behavioral features of new transactions, thereby complicating the detection process (Pareja et al., 2020; Weber et al., 2019).
3. **Extreme class imbalance and label scarcity.** The effective training of models is severely constrained by the highly imbalanced nature of transaction data and the scarcity of labeled samples. Unlike traditional domains, blockchain data is highly anonymous, and illicit behaviors are both diverse and covert. As a result, labeled datasets tend to be limited in scope and infrequently updated, making it difficult to support the training of large-scale supervised models (Sanjalawe & Al-E’mari, 2023; Wang et al., 2024a). This also leads to models that overfit known patterns of illicit transactions and suffer from poor generalization to new or evolving forms of malicious behavior.

Given the limitations of existing methods, this study aims to explore a new approach for detecting illegal transactions, one that not only adapts to the temporal changes in blockchain transaction graphs but also identifies entirely new illicit patterns from unlabeled data. The goal is to more effectively detect potential criminal activities within the blockchain transaction network and reduce financial losses. Specifically, in response to the dynamic and rapidly evolving nature of blockchain transactions mentioned earlier, this paper proposes a novel dynamic graph neural network based on co-association of semi-supervised (CoSemiGNN). The model incorporates two innovative methods to integrate semi-supervised co-occurrence relationships into the blockchain transaction graph structure. This enables the model to retain powerful graph feature extraction capabilities while more effectively capturing unknown illegal transaction patterns from unlabeled transaction graphs. Consequently, the model effectively combines the advantages of semi-supervised learning and dynamic graph neural networks. By utilizing semi-supervised learning, the model avoids the over-reliance on labeled data that is typical in fully-supervised learning. Meanwhile, a small number of illegal anchor points ensures the high quality of the illegal transaction features extracted from unlabeled data.

The main contributions of this research include:

- We propose a novel dynamic graph neural network model based on semi-supervised co-association to detect illegal transactions in blockchain. As shown in Fig. 1, the model breaks through the bottleneck of the existing dynamic graph model in feature representation and time series modeling through the dual-module coupling design. The CMOS (Co-association Module of Semi-supervised) module uses the semi-supervised learning framework to capture the emerging illegal transaction patterns in the blockchain transaction network in real time. SDGM (Self-attention Dynamic Graph Module) module uses GNN (Graph Neural network) combined with self-attention mechanism to deeply mine the local details and global structure in the transaction graph. The memory retention ability of Recurrent Neural Network (RNN) was integrated to fully capture the dynamic changing trading characteristics. This design of dual-module collaborative work not only significantly improves the accuracy and robustness of the model in illegal trade detection tasks, but also effectively prevents the problem of detection failure in the face of new data distribution
- In CMOS (Co-association Module of Semi-supervised), we propose two innovative methods to deeply integrate semi-supervised co-association with node and side information of the transaction

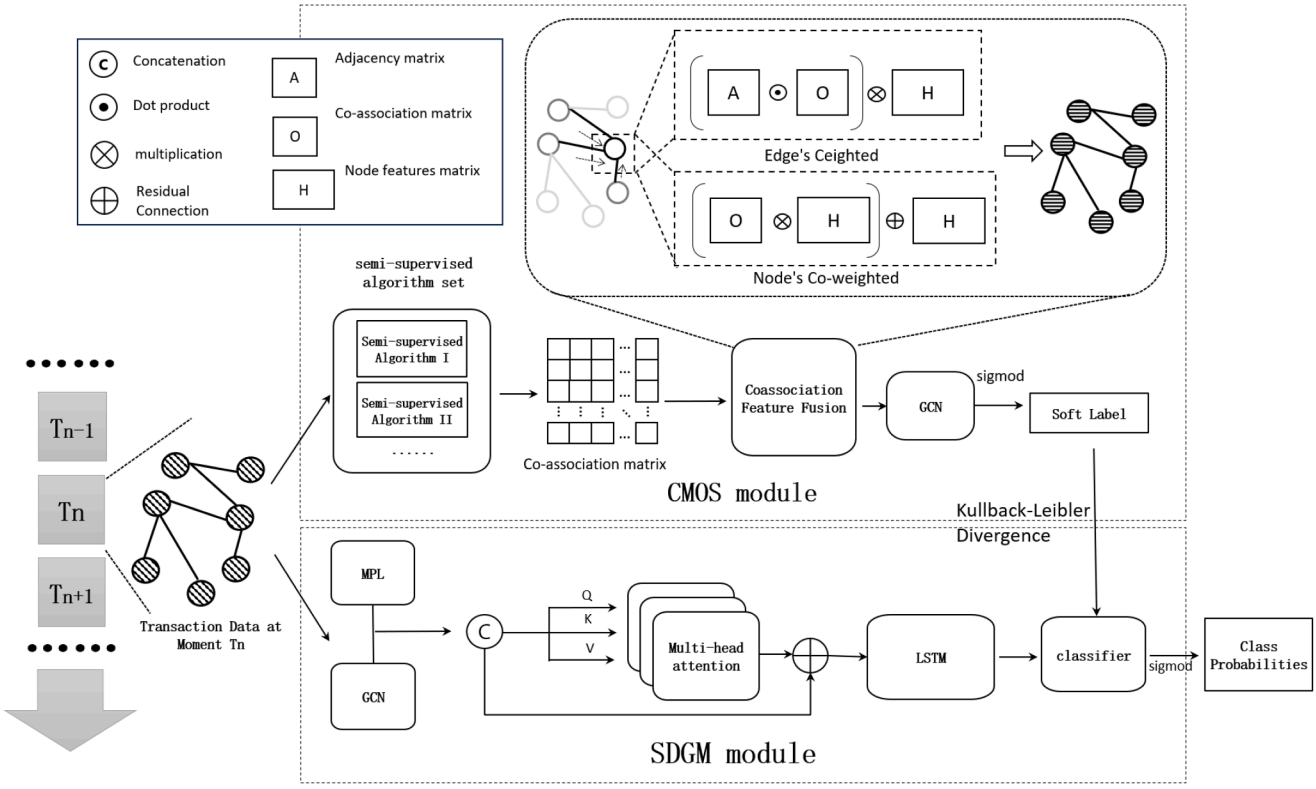

Fig. 1. CoSemiGNN Model Architecture. This diagram illustrates the two main components of the CoSemiGNN model: the Co-association Module of Semi-supervised (CMOS) and the Self-attention Dynamic Graph Module (SDGM), and how they interact to produce an illegal probability for each transaction.

graph. By constructing the node similarity matrix based on the co-association relationship, the limited labeled data and a large number of unlabeled data were effectively associated, and the characteristics of illegal transactions were accurately captured. It reduces the dependence of the model on manually labeled data, greatly improves the generalization ability of the model in the dynamic transaction environment, provides new theoretical support and solutions for solving the problem of illegal transaction detection in the dynamic environment, and also provides valuable reference for the blockchain and anomaly detection in other fields such as social networks and communication networks.

The structure of this paper is organized as follows:

Chapter 2 reviews related work and existing approaches for illicit transaction detection on blockchain.

Chapter 3 presents the architecture design of the proposed CoSemiGNN model and provides a theoretical analysis of the underlying algorithmic principles.

Chapter 4 introduces the dataset, descriptive data analysis, experimental setup, evaluation metrics, and result analysis.

Chapter 5 offers a comparative discussion between the proposed approach and existing techniques, objectively assessing its innovative advantages as well as potential limitations.

Chapter 6 concludes the study by summarizing the main contributions and proposing possible directions for future improvements and research, based on the identified limitations of the current work.

2. Related work

2.1. Static methods

We collectively refer to existing blockchain illicit transaction detection methods that do not consider temporal variations in transaction graphs as static methods. These methods are characterized by their simplicity, low computational overhead, and strong interpretability. XGBoost has been employed to detect illicit transactions on the Ethereum network (Farrugia et al., 2020). A variety of traditional machine learning classifiers, including Logistic Regression, Random Forest, Decision Tree, SVC, K-Nearest Neighbors, MLP, and AdaBoost, have been utilized to classify anomalous transactions (Feldman et al., 2021).

Graph Neural Networks (GNNs), a class of neural networks designed to process graph-structured data, propagate information and perform feature learning over graph topologies, thereby capturing complex interactions among nodes. This capability has led to their widespread success in domains such as social network analysis, recommender systems, and bioinformatics (Scarselli et al., 2009). In the context of illicit transaction detection on blockchain, the GNN's ability to aggregate features from related transaction nodes allows it to model intricate relationships within the transaction graph, thereby uncovering potential criminal activities.

The MIT-IBM Watson AI Lab explored the detection of illicit transactions in real blockchain environments. They found that Random Forest models exhibited notable advantages in handling binary classification problems for illicit transactions, and proposed the integration of Random Forest with GNN to enhance detection performance (Weber et al., 2019). They also introduced the Elliptic dataset, which remains the largest publicly available Bitcoin transaction graph dataset to date. This contribution has significantly advanced the application of Graph Convolutional Networks (GCN) in blockchain-based illicit transaction detection.

Building on existing research, challenges in applying GCN to illicit transaction detection were further examined (Alarab et al., 2020). A novel method combining GCN with linear layers was then proposed to enhance information flow and detect illicit activity on the Bitcoin blockchain (Alarab et al., 2020). In another study, GCN was utilized to extract behavioral and structural information from Ethereum

transactions, analyzing multiple networks derived from transaction behavior features (Tan et al., 2023). Similarly, a different approach combined Graph Attention Networks (GAT) with a subtree attention mechanism and two types of bootstrap aggregation to detect anomalous nodes, significantly improving model robustness through their ensemble learning strategy (Chang et al., 2024). An extension of this work also employed various graph neural network techniques, including GAT, GCN, Random Forest, and Struc2Vec, to detect abnormal transactions (Yu et al., 2021).

Collectively, these studies demonstrate that static methods possess clear advantages in structural modeling. However, their inability to capture the temporal evolution of transaction graphs limits their adaptability in real-world blockchain environments, thereby hindering further improvements in detection performance. Consequently, researchers have increasingly shifted their focus toward dynamic feature modeling, aiming to better align detection systems with practical application scenarios.

2.2. Dynamic method

To better adapt to the dynamic transactional environment of blockchain and improve the accuracy and robustness of illicit transaction detection, some studies have attempted to introduce temporal sequence modeling, integrating graph-structural information with temporal evolution characteristics. We collectively refer to such approaches as dynamic methods.

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specifically designed to process sequential data. They are capable of handling time series or any form of ordered datasets. The key feature of RNNs lies in their internal recurrent structure, which enables the network to consider not only the current input but also information from previous inputs. In recent years, researchers have increasingly incorporated RNN variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) into graph neural networks to handle data from dynamically evolving graph structures.

The MIT-IBM Watson AI Lab proposed EvolveGCN to extract temporal information from dynamic transaction graphs by updating GCN weights with LSTM and GRU (Pareja et al., 2020). Building on a prior idea (Weber et al., 2019), a Graph Convolutional Decision Forest was proposed, which combines dynamic graph convolutional networks with deep neural decision forests (Mohan et al., 2023). For path failure prediction, LRGCN was introduced; it treats the temporal dependencies between adjacent graph snapshots as a special memory mechanism and uses relational GCN to process both intra-temporal and inter-temporal relations (Li et al., 2019).

Other methods have also focused on combining spatial and temporal features. GCRN (Graph Convolutional Recurrent Network) was introduced for structured sequence prediction, capturing spatial structure with GCNs and temporal dynamics with RNNs (Seo et al., 2016). Similarly, GC-LSTM employs LSTM as the backbone for temporal modeling and GCNs at each time step to capture local node structures and inter-node relationships for dynamic network link prediction (Chen et al., 2022). DCRNN (Diffusion Convolutional Recurrent Neural Network) takes a different approach by modeling spatial dependencies via bidirectional random walks and capturing temporal dependencies using an encoder-decoder architecture (Li et al., 2017). More recently, a GRU-GAT model was proposed, integrating bidirectional recurrent neural networks with graph attention networks (Zheng, 2022).

These methods generally employ RNNs to capture temporal features and combine them with GNNs to extract structural patterns from graph data.

Overall, current dynamic methods have shown promising results in modeling the evolution of structural relationships within transaction graphs, effectively capturing structural dynamics. However, most of these methods still overlook the feature dynamics-that is, the temporal variability of node and edge attributes. In real-world blockchain environments, due to continuously evolving attack strategies, regulatory policy shifts, and the emergence of new fraud schemes, the attributes of illicit transactions may exhibit significantly different distributions across time periods. Such feature shifts often result in distribution drift, which can severely degrade a model's generalization performance and limit its ability to identify novel illicit behaviors, thereby reducing its effectiveness in practical detection tasks.

In order to more effectively maintain the security of blockchain, it is urgent to design a detection framework that can perceive and adapt to the change of feature distribution, so that the model has stronger temporal robustness and generalization ability.

2.3. Semi-supervised/unsupervised learning methods

Many researchers have focused on semi-supervised or unsupervised methods. For instance, a review of blockchain anomaly detection techniques emphasized that unsupervised algorithms are not merely tools, but should be seen as an integral part of broader frameworks (Cholevas et al., 2024).

This shift is also evident in the traditional financial domain. Researchers have proposed an unsupervised fraud detection model that uses attention-based autoencoders and GANs to treat fraud as an anomaly (Jiang et al., 2023). Similarly, another study combined GNNs with attention mechanisms and multi-view data to detect fraud, leveraging both labeled and unlabeled data by incorporating social relationships (Wang et al., 2019). Additionally, a semi-supervised adversarial framework was introduced to generate synthetic samples with a generator, which allows for effective learning with only a few labeled instances (Sanjalawe & Al-E'mari, 2023).

Building on these ideas, other work has focused on enhancing existing graph-based methods. One study enhanced traditional label propagation by applying minimal substitution theory to extend transaction labels (Wang et al., 2024a). Furthermore, graph learning models have been applied to anti-money laundering and phishing detection; one team used GNNs to identify laundering paths (Karim et al., 2024), while another improved GCNs through a process of important neighbor selection (Tang et al., 2022).

In the blockchain space, similar trends are emerging. A method was developed to predict illegal transactions by analyzing known Bitcoin transaction patterns (Xue et al., 2023). Additionally, a semi-supervised GNN was proposed that builds temporal transaction graphs and uses gated attention and risk propagation to model fraud behavior (Xiang et al., 2023).

Existing unsupervised or semi-supervised methods are basically based on static transaction network modeling, while focusing on capturing illegal transaction information from unlabeled data. However, few studies simultaneously consider the dynamic changes of the structure and sum characteristics of trading graphs in real-world trading environments. Moreover, current research often relies on a single technique to extract features from unlabeled data, which limits the robustness of the model and makes it difficult to apply the model effectively in real transaction environments.

To address the aforementioned challenges and enhance the model's adaptability to real-world transaction environments, we propose a novel approach for illicit transaction detection. Our method simultaneously leverages both the structural and temporal information of transaction graphs and captures emerging illicit transaction patterns from unlabeled data, as illustrated in the Figs. 1 and 2. Within the Co-association Feature Fusion module, we introduce two techniques to fuse the illicit transaction patterns extracted from unlabeled data with the transaction graph structure, thereby achieving an ensemble of multiple approaches. Subsequently, we employ a recurrent neural network (RNN) to capture the temporal dynamics of transactions and an attention mechanism to extract their global features. Finally, the newly identified illicit transaction patterns are incorporated into the model to further enhance its detection performance.

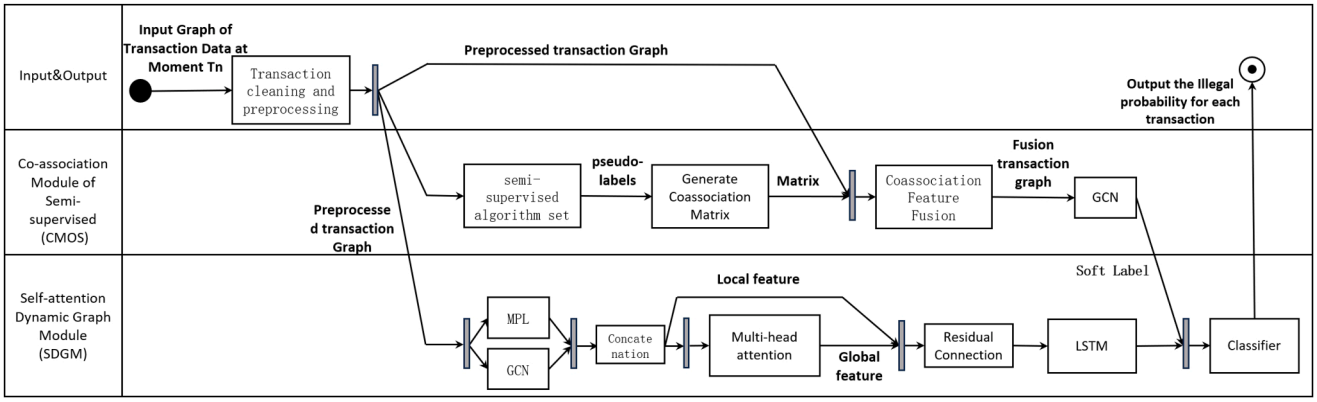

Fig. 2. UML diagram illustrating module interactions and data flow in CoSemiGNN. This figure details the interaction and data flow between the three main modules of the CoSemiGNN model: input/output layer, CMOS and SDGM. It shows how raw transaction data is fed into the model, processed separately in CMOS and SDGM modules after preprocessing, and finally the illegal probability of each transaction is obtained by fusing the output of the two modules.

3. Methods

To intuitively and clearly present the structure and operational workflow of the model, we first introduce the two primary modules of the model along with their internal mechanisms, followed by a UML flowchart that illustrates the overall architecture and the logic of inter-module interactions.

3.1. Module design

As shown in the Fig. 1, the model is divided into two core, collaboratively operating modules: the Co-association Module of Semi-supervised learning (CMOS) and the Self-attention Dynamic Graph Module (SDGM). These two modules respectively address the challenges of semi-supervised learning and dynamic graph modeling, working in synergy to enhance the model's representational capacity and predictive performance in financial fraud detection.

The model takes a time-series of transaction graph data as input, which first undergoes data cleaning and preprocessing. The preprocessed transaction graph data is then fed into both the CMOS and SDGM modules. The CMOS module is responsible for extracting information from unlabeled data and integrating it into the transaction graph. Specifically, it applies a series of semi-supervised learning algorithms to process the unlabeled data and aggregates their outputs to construct a co-association matrix (O). This matrix quantifies potential relationships between any pair of transaction nodes-even those not directly connected in the graph. Based on this co-association matrix, we propose two mechanisms: co-associative edge feature aggregation and co-associative edge feature weighting, which inject semi-supervised information into the node feature matrix (H), as shown in the upper-right section of the Fig. 1. These processed features are subsequently passed through a co-association feature fusion module and a GCN, with the output activated by a Sigmoid function to generate soft labels. These soft labels enable the model to capture emerging illicit transaction patterns using only a small set of anchor labels.

The SDGM module focuses on modeling the structural dynamics and global dependencies of transaction data. Initially, it simultaneously embeds raw transaction features into a hidden space via an MLP, while also extracting local structural features using a GCN. The concatenated output of these two branches is then passed into a multi-head attention mechanism, which adaptively captures global features in the transaction graph. The attention outputs are further passed into an LSTM network to capture the temporal dynamics of transaction data. During training, KL divergence is used as a weighted loss term to guide the SDGM module, while the soft labels are used to continuously fine-tune the classifier throughout the prediction phase. The classifier outputs the final illicit probability for each transaction after activation by a Sigmoid function.

3.2. Model architecture workflow

In the following, As shown in the Fig. 2, we detail the specific information flow and interaction processes among model components to better understand the overall working mechanism. The model begins with the input transaction graph data at time step T_n, which is first processed by the Transaction Cleaning and Preprocessing module to produce a Preprocessed Transaction Graph that serves as the common input for both CMOS and SDGM modules.

The CMOS module receives the preprocessed transaction graph and first sends it into a semi-supervised algorithm set to generate pseudo-labels. These pseudo-labels are then used to construct the Co-association Matrix. This matrix is fused with the features from the preprocessed transaction graph in the Coassociation Feature Fusion module to generate a Fusion Transaction Graph. The fusion graph is then passed into a GCN, whose output, after activation by a Sigmoid function, yields the Soft Labels-representing CMOS's prediction of emerging illicit transaction patterns.

Simultaneously, the SDGM module also receives the preprocessed transaction graph. It processes the input through two parallel branches: an MLP for embedding the raw features and a GCN for extracting local features. The outputs of these two branches are merged via Concatenation, then fed into the Multi-head Attention module to extract global features. These global features are passed through a Residual Connection and then into an LSTM network to capture temporal dependencies of transaction behavior. The output of the LSTM is passed into a Classifier, which also takes the Soft Labels from the CMOS module as supervisory signals. Finally, the classifier produces the illicit probability for each transaction.

In the subsequent sections, we provide a more detailed theoretical analysis and technical exposition of the CMOS and SDGM modules.

3.3. CMOS

3.3.1. Set of semi-supervised algorithms

For convenience of description, we define the semi-supervised algorithm set as in Eq. (1), where ω represents a specific semi-supervised learning algorithm, and m denotes the number of base algorithms.

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}. \tag{1}$$

Next, we define the dynamic transaction graph structure data set as in Eq. (2), where ε refers to a certain intermediate transaction moment, and ζ represents the final transaction moment.

$$\mathbf{D} = \{(V_t, E_t) \mid t = 1, 2, \ldots, \varepsilon, \ldots, \zeta\}. \tag{2}$$

Finally, we denote the transaction set at time $t = \varepsilon$ as in Eq. (3), where v represents a specific transaction node in the transaction set $V_{t=\varepsilon}$

at time $t = \varepsilon$. The flow of funds between transaction nodes is defined in Eq. (4).

$$V_{t=\varepsilon} = \{v_1, v_2, \ldots, v_n\}. \tag{3}$$

$$E_{t=\varepsilon} = \{< v_i, v_j > | \; v_i, v_j \in V \text{ and } v_i \neq v_j\}. \tag{4}$$

To ensure the generality of the semi-supervised algorithm, we only use simple and commonly used base algorithms such as selfTrain and sim-kernelKmeans in our approach.

1. SelfTrain aims to utilize a small amount of labeled data to train an initial model, which is then used to predict the labels of a large number of unlabeled data points. The predicted labels are referred to as pseudo labels. Next, the data with high-confidence pseudo labels is added to the training set, and the model is retrained using the updated set. This process is iterated until the model performance no longer improves or a pre-defined number of iterations is reached. The algorithmic steps can be described as follows:

1) At time t, take the partially labeled transaction set $V_0 = \{(v_i, y_i)\}$ and the rest of the unlabeled transaction data set $V_1 = \{v'_j\}$, and train the initial classifier f_0 using V_0.

2) At each iteration $t = 0, 1, 2, \ldots$, for each $v_j \in V_1$, compute $\hat{y}_j^t = f_t(v_j)$. Select high-confidence pseudo-labeled samples and add them to V_0. The confidence function is defined as:

$$S = \{(v_j, \hat{y}_j) \mid \text{confidence}(v_j, \hat{y}_j) > \kappa\}, \tag{5}$$

where κ is the pre-set confidence threshold.

3) Use the updated training set $V_0 = V_0 \cup S$ to retrain the model f_{t+1}. This process continues until either all transactions have confidence greater than κ or the maximum number of iterations is reached.

2. The K-means algorithm aims to partition data by minimizing the intra-cluster distance, ensuring that points within the same cluster are as close as possible, while points in different clusters are as far apart as possible. However, in our experiments, we found that relying solely on the unsupervised K-means method to partition transaction clusters is unrealistic. This is likely because in a real-world blockchain environment, the number of illegal transactions is much smaller than that of legal transactions, and illegal transactions often disguise their nature. As a result, only a few features of illegal transactions exhibit their fraudulent characteristics. In this case, unsupervised K-means is almost incapable of extracting useful information.

To address this issue, we adjusted the K-means algorithm by introducing a small amount of labeled data, allowing the algorithm to start from labeled data and derive the features of illegal transactions. By using the sim-Kmeans algorithm, we preserve the unsupervised feature extraction characteristics of K-means while improving clustering performance and algorithm stability. Furthermore, to enhance the feature extraction ability of the K-means algorithm and adapt to the non-linear structure of blockchain transaction data, we employed kernel methods to map the original transaction data to a higher-dimensional space, where the data might exhibit clearer clustering structures. This approach allows us to effectively extract features from unlabeled data and handle non-linearly separable transaction data.

Specifically, for the transaction data set $V = \{v_1, v_2, \ldots, v_n\}$, first, we map each data point v into the high-dimensional space to obtain $\phi(v_i)$ via a nonlinear mapping function ϕ. Then, we compute a small number of centers with labeled data as the initial cluster center m_c. As the algorithm continues to optimize Eq. (6), the cluster centers m_c are iteratively updated until the position of the center of mass no longer changes or a preset number of iterations is reached. At this point the algorithm converges and the final cluster division is obtained.

$$m_c = \frac{1}{N_c} \sum_{v \in C_c} \phi(v), J = \sum_{c=1}^{2} \sum_{v \in C_c} \|\phi(v) m_c\|^2, \tag{6}$$

where N_c is the number of points in cluster C_c, and $\|\|^2$ denotes the distance metric function.

In practical experiments, by adjusting the initial self-training algorithm and using different distance metrics, we can obtain semi-supervised results from different perspectives.

Finally, we define the results of the semi-supervised algorithm set as follows:

$$R = \{r_1, r_2, \ldots, r_m\}, \tag{7}$$

where $r_1, r_2, \ldots, r_m$ represent the prediction results of different semi-supervised algorithms for transactions at the same moment.

3.3.2. Co-association feature fusion and theoretical analysis

By using different semi-supervised algorithms, we can obtain a multi-dimensional, multi-angle pseudo-label dataset R. However, effectively integrating the results from different classifiers is a key challenge. Common methods for integration include voting, averaging, and stacking. Yet, none of these existing methods are able to integrate the structural information of graphs during the ensemble process, and this structural information is crucial for transaction analysis. Therefore, we propose a method that integrates multiple semi-supervised approaches with Graph Neural Networks (GNNs).

Firstly, inspired by the progress made in using Co-association Matrices in ensemble clustering, we construct the semi-supervised co-association matrix between different algorithms, as shown in Eq. (8). Based on this, we further analyze the structural similarity between the transaction graph and the co-association matrix O. We propose two methods to integrate semi-supervised co-association information into the feature extraction process: 1. Edge-based co-association weighting Eq. (9), which treats the semi-supervised co-association relations as the weight matrix of the original adjacency matrix. 2. Node-based co-association feature aggregation Eq. (10), where the aggregation result of the co-association matrix is added to the original transaction features.

The co-association matrix O is defined as:

$$O_{ij} = \frac{1}{m} \sum_{s=1}^{m} \mathrm{LI}\left(\omega^s(v_i), \omega^s(v_j)\right),$$
$$\mathrm{LI}\left(\omega^s(v_i), \omega^s(v_j)\right) = \begin{cases} 1, & \text{if } \omega^s(v_i) = \omega^s(v_j), \\ 0, & \text{if } \omega^s(v_i) \neq \omega^s(v_j), \end{cases} \tag{8}$$

where ω^s represents the semi-supervised algorithm used previously, and O_{ij} represents the average similarity of transactions v_i and v_j in the base classifiers at a given time.

Next, we describe the feature extraction process in the CosimGNN model, based on the co-association information.

In Eq. (9), the co-association matrix O is combined with the adjacency matrix A via the Hadamard product to generate a new weight matrix. This matrix can map the semi-supervised co-association relations to the actual transaction edges. Then, the new weight matrix is combined with the identity matrix I, which adds self-loops to nodes to ensure that the aggregation process retains the node's own transaction features. Multiplying by the inverse degree matrix D^{-1} ensures that edges with different degrees are treated in the same metric space. Finally, we multiply by the learnable parameters and apply a nonlinear transformation to update the transaction feature $H^{(l)}$:

$$H^{(l+1)} = \sigma\left(D^{-1}(A \odot O + I) H^{(l)} W^{(l)}\right). \tag{9}$$

In Eq. (10), we first multiply the co-association matrix O with the node feature representation $H^{(l)}$ to obtain the aggregated co-association features. We then add this result to the original features $H^{(l)}$, which enables the incorporation of additional co-association features while retaining the original node features. These co-association features capture global patterns related to similar relationships with the node. We then multiply by the normalized adjacency matrix $D^{-1}(A + I)$ to further aggregate the local features of nodes using the graph structure. Finally, we update the node feature representation $H^{(l)}$ by multiplying with the learnable parameters and applying a nonlinear activation function:

$$H^{(l+1)} = \sigma\left(D^{-1}(A + I)\left(O H^{(l)} + H^{(l)}\right) W^{(l)}\right). \tag{10}$$

We name the model obtained from Eq. (9) as CosimGNN-E, and the model derived from Eq. (10) as CosimGNN-V.

This approach allows us to seamlessly integrate semi-supervised information and graph structural data, providing a more robust and effective solution for detecting abnormal transactions in blockchain networks.

3.3.3. Temporal feature extraction

Using the method described above, we successfully introduced semi-supervised co-association relations into the feature extraction process for transactions. To accommodate the dynamic nature of blockchain transaction networks in the temporal dimension, we further integrate Recurrent Neural Networks (RNNs) to extract temporal sequence information. Specifically, we employ Long Short-Term Memory (LSTM) networks as the memory module.

The computation of the graph convolution weights at time step i is given by:

$$\begin{aligned} I_t &= \sigma\left(W_I V_t + U_I H_{t-1} + B_I\right), \\ O_t &= \sigma\left(W_O V_t + U_O H_{t-1} + B_O\right), \\ \widetilde{C}_t &= \tanh\left(W_C V_t + U_C H_{t-1} + B_C\right), \\ C_t &= F_t \odot C_{t-1} + I_t \odot \widetilde{C}_t, \\ W_t^{LSTM} &= O_t \odot \tanh\left(C_t\right), \end{aligned} \tag{11}$$

where:σ denotes the sigmoid activation function, used for computing gate signals. tanh denotes the hyperbolic tangent activation function, used to compute the candidate cell state. $\odot$ represents the Hadamard product (element-wise multiplication). W, U, B represent weight matrices and bias terms, with subscript I, F, O, and C denoting the input gate, forget gate, output gate, and cell state, respectively. V_t is the input feature vector at the current time step. H_{t-1} is the hidden state at the previous time step. C_{t-1} is the cell state at the previous time step. I_t, F_t, O_t are the activation values of the input gate, forget gate, and output gate, respectively. $\widetilde{C}_t$ is the candidate cell state. C_t is the current time step's cell state.

The LSTM-based model helps capture temporal dependencies in transaction data, which is crucial for analyzing blockchain transaction sequences and detecting anomalies or illegal activities over time. This integration of LSTM with graph convolution enables our model to consider both the structural information of the transaction graph and the temporal dynamics, significantly improving the detection of abnormal transactions.

3.3.4. CMOS Algorithm

As shown in Algorithm 1:

1. Use multiple semi-supervised algorithms to extract features from different dimensions of the transaction graph.
2. Aggregate these features through the co-association matrix O.
3. Integrate semi-supervised features with the structural information of the transaction graph.
4. Use Long Short-Term Memory (LSTM) to model the temporal sequence and generate the final output P.

3.4. SDGM

In the SDGM architecture, we first adopt a hybrid model framework that combines the advantages of Multi-Layer Perceptrons (MLP) and Graph Neural Networks (GNN) to fully capture both the raw features and structural features of transaction data. Specifically, we first use the Multi-Layer Perceptron (MLP) to map the raw transaction features into a hidden-dimensional space, extracting deeper feature representations. Subsequently, to further leverage the structural features of the transaction data, we introduce Graph Neural Networks (GNN) to aggregate the neighbor features of each transaction node, thereby obtaining a more comprehensive representation of the transaction nodes.

Algorithm 1 CMOS algorithm.

Input: Transaction graph at time t (V_t, E_t), semi-supervised algorithm set Ω
Output: co-association module of Semi-supervised P
Compute the semi supervised co-association matrix
for $i \in \{1, 2, \ldots, m\}$ **do**
  $r_i \leftarrow \omega_i(V)$
  Compute co-association, where $\oplus$ indicates OR
  $O_{ij} \leftarrow \frac{1}{m} \sum_{\substack{e_j, e_k \in E \\ i \neq j}} r_i(e_j) \oplus r_i(e_k)$
end for
Use Eq. 9 or Eq. 10 to fuse semi-supervised co-association features and graph structure to obtain $H_t^{(l)}$
$P \leftarrow \mathrm{LSTM}\left(H_t^{(l)}, W_t^{(l)}\right)$
return P

$$H_t = \sigma(W, V_t) \oplus GNN(V_t, E_t, W), \tag{12}$$

where V_t represents the transaction set at time t, and E_t represents the edge set at time t. Here, σ denotes a nonlinear activation function, and W represents the weight matrix, which is used in the neural network to transform the input features into hidden layer representations.

3.4.1. Global feature extraction

In real-world transaction data, there are often significant differences between the same feature of different transactions, and these differences are often key to distinguishing the legality of transactions. Traditional Graph Neural Networks (GNNs) usually focus on aggregating local features, which lacks consideration of global features. On the other hand, attention mechanisms have gained prominence due to their global perspective on feature extraction and their unique ability to selectively focus on important aspects. Therefore, we attempt to supplement the model with an attention mechanism to further capture the global features of transactions. The specific formula is as follows:

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{13}$$

where Q, K, and V represent the query, key, and value matrices, respectively. These matrices are derived from the co-association matrix through linear transformations. Specifically:

$$Q = O_t W^Q, \quad K = O_t W^K, \quad V = O_t W^V, \tag{14}$$

here, W^Q, W^K, and W^V are learnable parameter matrices, and O is the co-association matrix.

During the computation, the following steps are performed: first, the dot product between the query matrix Q and the key matrix K is computed to obtain the similarity scores. Then, these scores are scaled by the square root of the dimension of the key vectors, d_k, to prevent the dot product results from becoming excessively large. Next, the Softmax function is applied to normalize the scores, producing a score matrix. Finally, this score matrix is multiplied by the value matrix V to obtain the final attention output.

3.4.2. Loss function

At this point, the two components of our model are capable of extracting information from unlabeled data and extracting important transaction features at multiple scales. Therefore, we need an effective way to fuse these two components, transmitting the semi-supervised information into the supervised model. Specifically, we use the Kullback-Leibler (KL) divergence between the pseudo-label distribution from the semi-supervised methods and the true label distribution from the supervised methods as a regularization term. This enables the model to not

fully fit the real labels during training, but instead fuse the structural information extracted by the semi-supervised methods, thus improving the model's generalization ability and its capacity to adapt to the dynamic changes of data in the blockchain transaction environment. The mathematical formulation is as follows:

$$\arg\min_{\theta}\left[(1-\alpha)\mathrm{KL}(P\|Q_\theta)+\alpha\mathrm{BCE}(Y,Q_\theta)\right], \tag{15}$$

where $\mathrm{KL}(P\|Q_\theta)$ represents the Kullback-Leibler (KL) divergence, $\mathrm{BCE}(Y,Q_\theta)$ is the binary cross-entropy (BCE) loss, α is a weight parameter, P is the pseudo-label distribution obtained by the semi-supervised methods, Q_θ is the final output distribution of the model, and Y is the true label.

3.4.3. SDGM Algorithm

Algorithm 2 SDGM algorithm.

Input: Dynamic transaction graph set $\mathbf{D}=\{(V_t,E_t)\mid t=1,2,\ldots,\varepsilon,\ldots,\zeta\}$, semi-supervised algorithm set Ω
Output: Model weights W
for $t\in\{1,2,\ldots,\varepsilon\}$ **do**
  Use CMOS algorithm to fuse semi-supervised co-occurrence features
  $P\leftarrow CMOS((V_t,E_t),\Omega)$
  Embed original features into hidden layers
  $H1_t\leftarrow\sigma(v_t,W_t)$
  Aggregate node neighbor features via graph convolution
  $H2_t\leftarrow\mathrm{GCONV}\left((V_t,E_t),W_t\right)$
  $H\leftarrow H1_t\oplus H2_t$
  Extract global features using attention mechanism
  $H_t\leftarrow H\oplus Atte(H,W_t)$
  $H\leftarrow\mathrm{RNN}\left(H_t,W_t\right)$
  Train the model using equation 15 to obtain model weights W
end for
return W

The specific steps of the SDGM algorithm are shown in Algorithm 2. The SCDG inputs the dynamic transaction graph and the set of semi-supervised algorithms, and fuses the semi-supervised co-occurring features using the CMOS algorithm in each time step. Then the original node features are embedded in the hidden layer and aggregated with neighbor information through graph convolution to splice to get the fused features; the global features are extracted using the attention mechanism and the timing information is integrated using RNN, and finally the model is trained through the predetermined loss function, and the final model weights are updated and output.

4. Experiment

4.1. Dataset

The Elliptic dataset, released by Weber et al. (2019), contains transaction data extracted from the real Bitcoin blockchain network. It stands as one of the largest and most representative open-source graph-structured datasets to date, specifically curated for classifying illicit activities in Bitcoin transactions. The dataset comprises 203,769 transactions, forming a directed graph based on the relationships between transactions and addresses.

The transaction labels were provided by Elliptic Ltd., relying on its collaboration with multiple international law enforcement agencies (e.g., Europol, FBI, etc.). Labels were assigned by identifying wallet addresses directly associated with illicit activities such as money laundering, fraud, and drug trafficking, thereby tagging the corresponding transactions as illicit, licit, or unknown. The labeling process adopts a semi-automated approach, combining human expert verification with algorithmic assistance, ensuring high levels of accuracy and credibility. A portion of the illicit addresses can be traced

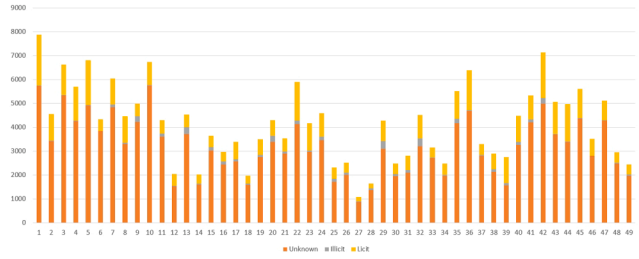

Fig. 3. The proportion of various types of transactions in the transaction graph of each time slice of the Bitcoin network, the horizontal axis represents the time line, and the vertical axis represents the quantity. Yellow denotes legitimate transactions, orange denotes unknown transactions, and blue denotes illegal transactions.

to transactions linked darknet marketplaces such as Silk Road and Hydra.

The Elliptic dataset provides a total of 166 features to describe each transaction. The first 94 features are local transaction features, covering attributes such as temporal steps (49 time intervals spanning approximately two weeks), input/output counts, transaction fees, and transaction amounts. The remaining 72 features are neighbor-aggregated statistics, representing summary metrics (e.g., maximum, minimum, standard deviation, correlation, etc.) from neighboring transactions.It is important to note that, due to privacy and compliance considerations, the dataset does not publicly disclose detailed semantic annotations for these 166 features. As such, the exact definitions of individual features remain unavailable. In 2023, Elmougy and Liu (2023) released an extended version known as Elliptic++, which introduces 17 enhanced features on top of the original dataset. These features are derived via deanonymization techniques and on-chain analytical methods, and are automatically extracted based on UTXO structures, transaction sizes, and address interaction patterns. Detailed descriptions of these enhanced features are provided in Table 1.

Moreover, although the Elliptic and Elliptic++ datasets offer high-quality labeled resources, challenges related to label incompleteness, timeliness, and bias are inherent to the blockchain's highly anonymous, open, and dynamic environment. As shown in the Fig. 3. only approximately 2.23 % of the transactions are labeled as illicit, while 20.62 % are labeled as licit. This substantial class imbalance may introduce significant bias during model training and evaluation. Therefore, in our model design and experimental setup, in addition to adopting semi-supervised learning algorithms, we also incorporate strategies such as weighted loss functions to mitigate the impact of class imbalance on classification performance.

4.2. Analysis of blockchain transaction characteristics

To comprehensively understand the challenges of illicit transaction detection in blockchain environments, we conducted a systematic analysis of the temporal evolution characteristics of real transaction data. Compared to traditional financial systems, blockchain's decentralization and anonymity render illicit transactions not only more covert but also subject to continuous evolution in behavioral patterns and feature manifestations over time. This evolution primarily manifests in two aspects:

Structural Dynamics: Refers to temporal changes in the transaction graph structure, reflected by fluctuations in structural statistical features such as the number of nodes, edges, and graph density.

Characteristic Dynamics: Refers to significant variations in feature distribution, combination patterns, and key attributes of illicit transactions over different time periods, even when their labels (e.g., illicit) remain unchanged.

To quantitatively assess the potential impact of these dynamics on model performance, we performed exploratory analyses based on real transaction data:

Table 1
Description of Features in the Elliptic++ Dataset. The left side shows the feature name and the right side shows the detailed description.

Feature Name	Description
Feature1-Feature94	Local transaction features
Feature95-Feature166	Aggregated neighborhood features
In-txs-degree	Number of incoming transactions (in-degree)
Out-txs-degree	Number of outgoing transactions (out-degree)
Total-BTC	Total amount of Bitcoin involved in the transaction (input or output)
Fees	Transaction fee, calculated as input amount minus output amount
Size	Transaction size in bytes
Num-input-addresses	Number of input addresses involved
Num-output-addresses	Number of output addresses involved
In-BTC-min	Minimum of all input amounts
In-BTC-max	Maximum of all input amounts
In-BTC-mean	Mean of all input amounts
In-BTC-median	Median of all input amounts
In-BTC-total	Total sum of all input amounts
Out-BTC-min	Minimum of all output amounts
Out-BTC-max	Maximum of all output amounts
Out-BTC-mean	Mean of all output amounts
Out-BTC-median	Median of all output amounts
Out-BTC-total	Total sum of all output amounts

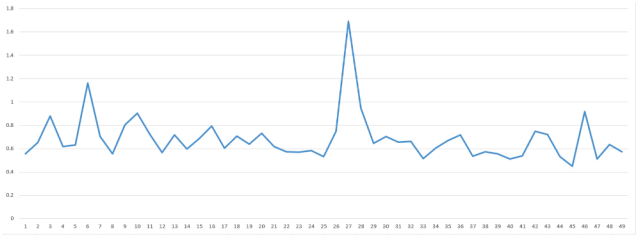

Fig. 4. Sparsity of Bitcoin transaction graphs in different time slices. The horizontal axis represents time and the vertical axis represents sparsity ratio.

4.2.1. Structural dynamics analysis

We employed graph sparsity, measured as the ratio of nodes to edges, to quantify temporal changes in graph structure. This metric provides a simple yet effective evaluation of the evolving intensity of transaction activities and network topology across different time intervals.

The experimental results in Fig. 4 show that the transaction graphs exhibit clear sparsity at all time points. The node-to-edge ratio fluctuates primarily between 0.5 and 0.9, indicating significant differences in transaction network activity levels and connection patterns across time slices. This temporal variation in structural sparsity reflects pronounced changes in network activity and behavioral modes at different stages, evidencing strong structural dynamics in blockchain transaction graphs over time.

4.2.2. Feature dynamics analysis

Focusing on characteristic dynamics, we selected time slice 43, corresponding to the real-world shutdown of a darknet market, serving as a typical case to observe illicit transaction behavior evolution.

We extracted illicit transaction samples from adjacent periods before and after the event and evaluated changes in feature importance using a decision tree model. As shown in Fig. 5, the blue curve (pre-event) highlights that features 53, 54, and 55 predominantly contributed to illicit transaction identification, whereas the yellow curve (post-event) reveals a significant decrease in their importance and a marked increase in the relevance of features 1, 169, and 155. This shift indicates that under external regulatory pressure, criminals rapidly adapted their transaction behaviors and obfuscation strategies, causing key feature migration.

This phenomenon illustrates the pronounced dynamic and non-stationary nature of illicit transaction features. Most existing dynamic graph modeling approaches focus primarily on structural dynamics and exhibit limited adaptability to characteristic dynamics, leading to performance degradation during abrupt feature distribution shifts (Pareja et al., 2020; Weber et al., 2019).

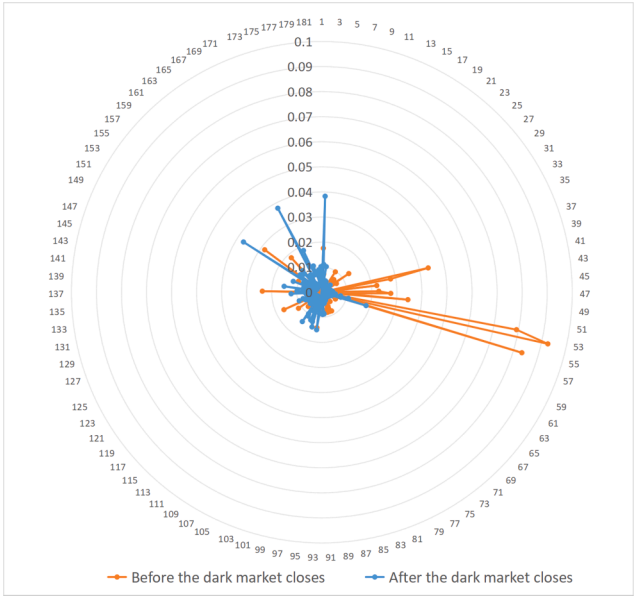

Fig. 5. Bitcoin illegal trading data at different moments using feature importance scores from decision forests. The outer circle shows the transaction features in the Elliptic++ dataset.The orange curve represents before the dark market closes, and the blue curve represents after the dark market closes.

Therefore, in subsequent experiments, we specifically select time intervals exhibiting significant characteristic dynamics to rigorously evaluate the robustness and adaptability of the proposed method under feature distribution shifts, aiming to closely emulate the complex challenges faced in real-world illicit transaction detection scenarios.

4.3. Experimental setup

4.3.1. Data preprocessing

During data preprocessing, to remove the influence of differing feature scales, we standardly scaled node initial features. Considering the common presence of extreme values or long-tailed distributions in financial data, directly applying Min-Max scaling may compress most data into a very small interval (Elmougy & Liu, 2023).

To address this issue and improve model robustness, we adopt Quantile Scaling. This nonlinear transformation maps feature values to a uniform distribution over the interval [0,1], is insensitive to outliers, and better preserves the original distributional structure. Based on empirical settings, we set the lower and upper quantile bounds to 0.01 and 0.99, respectively. Finally, following the settings in Weber et al. (2019) and Pareja et al. (2020), we split each dataset chronologically into a training set (the first 34 time steps) and a test set (the last 15 time steps).

4.3.2. Experimental environment

All experiments were conducted in the following environment:

Hardware environment: GPU is an NVIDIA RTX 3060 GPU (12 GB memory), CPU is an Intel(R) Xeon(R) E5-2673, with 32 GB system memory. **Software environment**: programming language Python 3.8; core frameworks are PyTorch 1.10.0 (cu113) and PyG (torch-geometric) 2.0.0.

4.3.3. Baseline model

To comprehensively evaluate the performance of our proposed CosemiGNN model, we selected multiple state-of-the-art and mainstream dynamic graph neural network models as baselines for horizontal comparison. To ensure fairness and reproducibility, all baseline models were implemented directly using their publicly released codebases, with only data input interfaces adapted as needed, without any other modifications, and executed under the same experimental environment. In addition, for the hyperparameter settings of each baseline model, we prioritized following the configurations recommended in their original papers. A brief description of each baseline model and its corresponding hyperparameters is provided as follows (More detailed experimental process can get from https://github.com/sunflower110/CoSemiGNN):

1. **evolveGCN-o** (Pareja et al., 2020): updates GNN weight parameters using an LSTM.
2. **evolveGCN-h** (Pareja et al., 2020): updates GNN weight parameters using a GRU.
3. **LRGCN** (Li et al., 2019): employs a gated mechanism for long-term memory and integrates an attention mechanism for path representation; implemented with num of bases = 3 and num of relation = 3.
4. **GConvGRU** (Seo et al., 2016): uses a CNN to identify spatial structures and combines an RNN to capture dynamic patterns; implemented with Chebyshev filter size = 2.
5. **GC-LSTM** (Chen et al., 2022): applies graph convolution to capture graph structural information and integrates it into an LSTM; implemented with Chebyshev filter size = 2.
6. **DCRNN** (Li et al., 2017): captures spatial dependencies via bidirectional random walks and temporal dependencies using an encoder-decoder architecture with scheduled sampling; diffusion convolution order = 2.
7. **A3T-GCN** (Bai et al., 2021): uses gated recurrent units to capture short-term trends in time series and applies graph convolutional networks to capture spatial dependencies, introducing an attention mechanism to adjust the importance of different time points; number of temporal periods = 2.
8. **AGCRN** (Bai et al., 2020): enhances graph convolutional networks through two adaptive modules; filter size = 2.
9. **DyGrEncoder** (Taheri et al., 2019): employs an encoder-decoder framework that projects the dynamic graph at each time step into a d-dimensional space, using an autoencoder to learn node representations, a gated graph neural network (GGCN) to capture per-time-step graph structure, and an LSTM to capture temporal information; implemented with GGCN depth = 2 and neighbor mean aggregation.

It is worth noting that most existing semi-supervised graph learning methods are primarily designed for static graphs, making them difficult to apply directly to dynamic graph scenarios. To ensure the validity and fairness of our evaluation, such methods were not included in the comparison.

4.3.4. Parameter settings

Regarding hyperparameter settings, all comparative models were trained using a unified configuration: the learning rate was fixed at 1×10^{-4}, batch size was set to 1, and the Adam optimizer was used. The number of training epochs was adaptively adjusted between 500 and 1000 based on model convergence. The loss function was uniformly set as weighted binary cross-entropy to mitigate the impact of class imbalance on the training process.

To ensure that models possess sufficient representational capacity and achieve a fair comparison under the same hardware conditions, certain core structural parameters (such as hidden layer dimensions) were uniformly set to 256. More detailed model configurations can be found in the accompanying code repository.

For the proposed CosemiGNN, key hyperparameters were determined through empirical settings and experimentation. Specifically, to maintain consistency with common practices in semi-supervised baseline algorithms (Sohn et al., 2020; Xiang et al., 2023), and to balance the quality and quantity of pseudo-labels, the confidence threshold κ was set to 0.85, while the initial anchor ratio was set to 0.3. For more sensitive hyperparameters in our method, the number of semi-supervised base classifiers m was set to 5, and the loss function balance coefficient α was set to 0.3. A parameter sensitivity analysis will be detailed in the subsequent experimental section to ensure the interpretability and reproducibility of the method's performance.

4.3.5. Evaluation metrics

In existing studies, some researchers ignore the severe class imbalance present in blockchain transaction data, where legitimate transactions vastly outnumber illicit transactions. Under such circumstances, overall accuracy and similar metrics can be misleading because a model that predicts all samples as the majority class can still achieve a very high score.

For a more meaningful evaluation, besides computing overall metrics, we separately compute precision, recall, and F1-score for legitimate (majority class) and illicit (minority class) transactions. Since the primary goal of the experiments is to identify illicit activity, in subsequent experiments we focus more on the performance metrics for the illicit class, especially the F1-score which balances precision and recall.

4.4. Comparative experiment

Table 2 presents a detailed performance comparison between CosemiGNN and all baseline models. As expected, overall evaluation metrics (e.g., Total F1) are generally high due to the dominance of legitimate transactions, but these do not truly reflect a model's core capability of detecting illicit transactions.We therefore shift focus to the more revealing illicit-class metrics. Fig. 6 visually compares the F1-scores, accuracy and recall of each model for illicit transaction detection. Our model CosemiGNN demonstrates significant superiority on this key metric, achieving an F1 score of approximately 0.74, surpassing all baseline models and realizing about a 30 % relative performance improvement. This provides strong evidence of CosemiGNN's superior ability to precisely identify illicit transactions. Further analysis indicates that this improvement is mainly attributable to the model's adaptability to changes in feature distributions. To that end, we perform a chronological analysis of each model's adaptability to feature dynamics.

4.5. Experimental of robustness to feature dynamics

To thoroughly analyze the robustness of the model when faced with changes in data distribution (i.e., Feature dynamics), we plotted the F1 score variation curves for illegal category detection of each model at every time step on the test set, as shown in Fig. 7.

Table 2
Performance Comparison of CoSemiGNN with Baseline Models. This table presents a comprehensive performance evaluation of the proposed CoSimGNN-V and CoSimGNN-E against various Baseline models. The comparison is based on a range of metrics, including overall accuracy, F1 scores, precision, and recall for both illegal and legal transactions, as well as the micro F1 score. The highest values for each metric are highlighted in bold.

Model	Accuracy	Illegal F1	Legal F1	Micro F1	Illegal Precision	Legal Precision	Illegal Recall	Legal Recall
CoSimGNN-V	0. 9814	0. 7446	0. 9899	0. 8673	0. 7757	0. 9863	**0. 7302**	0. 9936
CoSimGNN-E	0. 9847	0. 8016	0. 9917	0. 8966	0. 9643	0. 9860	0. 7111	0. 9974
GAT	0. 8841	0. 3674	0. 9312	0. 6493	0. 3523	0. 9724	0. 5299	0. 9312
GCN	0. 9316	0. 3845	0. 9633	0. 6739	0. 3855	0. 9679	0. 4045	0. 9633
EvolveGCN-H	0. 9454	0. 4056	0. 9702	0. 6879	0. 4602	0. 9668	0. 3917	0. 9702
EvolveGCN-O	0. 9510	0. 4383	0. 9732	0. 7058	0. 4425	0. 9788	0. 4948	0. 9732
LRGCN	0. 8990	0. 3813	0. 9437	0. 6625	0. 3380	0. 9733	0. 5338	0. 9180
GConvGRU	0. 9212	0. 4112	0. 9567	0. 6839	0. 4086	0. 9716	0. 5139	0. 9437
GCLSTM	0. 9064	0. 3743	0. 9479	0. 6611	0. 3437	0. 9730	0. 5063	0. 9263
DCRNN	0. 9014	0. 3578	0. 9454	0. 6516	0. 3271	0. 9696	0. 4825	0. 9241
A3TGCN	0. 9184	0. 3928	0. 9548	0. 6738	0. 3761	0. 9739	0. 4826	0. 9381
AGCRN	0. 8490	0. 3375	0. 9131	0. 6253	0. 2830	0. 9741	0. 5646	0. 8651
DyGrEncoder	0. 9206	0. 4182	0. 9561	0. 6872	0. 4260	0. 9749	0. 5231	0. 9395

Fig. 6. Performance Comparison of CoSemiGNN and Other Baseline Models on a Fraud Detection Dataset. This bar chart presents a comprehensive performance evaluation of the CoSemiGNN-V and CoSemiGNN-E models against a range of benchmark graph neural network models. The comparison is based on three key metrics for illegal transactions: F1 score, Accuracy, and Recall.

Fig. 7 illustrates the changes in F1 scores for negative class samples of each model over time. From the experimental results, we observe that CoSemiGNN consistently outperforms baseline models in illegal transaction detection F1 scores. The F1 scores of GCN and GAT models fluctuate drastically over time with high variance. Although these two models effectively process graph-structured data by aggregating neighbor information, they are based on static network modeling. In real blockchain environments, transaction data continuously evolves dynamically. Over time, the gap between static training data and the illegal transaction data that the model needs to detect widens, leading to a sharp decline in model performance. It is especially noteworthy that the 43rd time slice in the Elliptic++ dataset marks the shutdown of the dark web market. This represents a completely new event not present in the training data, making it extremely difficult for static models to accurately identify illegal activities from entirely unfamiliar data.

To address such temporal changes, dynamic models like EvolveGCN, LRGCN, and GC-LSTM leverage recurrent neural networks to capture the temporal information in transaction data. However, our study found that even these dynamic models experience significant drops in F1 scores when confronted with completely new events. This is because these models typically rely on a core assumption: the training and testing data distributions are similar. They excel at handling transactions consistent with the training distribution, but when data distribution shifts significantly (e.g., changes in feature or label distributions), their generalization ability weakens, resulting in poor performance.

The core advantage of CoSemiGNN lies in its clever integration of semi-supervised learning and dynamic graph networks. The model not only retains the ability to understand time-series data but, more importantly, can learn and extract illegal features from unlabeled data using only a small amount of labeled data, thereby updating the model. This mechanism greatly enhances the model's adaptability and detection capability for novel events. The semi-supervised feature extraction mechanism enables CoSemiGNN to autonomously recognize illegal patterns in the absence of explicit labels, providing a powerful solution for handling sudden, unprecedented events such as the dark web shutdown.

4.6. Ablation experiment

We designed ablation experiments to explore the specific impact of each key component of the model on overall performance. Specifically:

1. **CoSimGNN-V**$_{cosim}$: Ablation of the semi-supervised co-association fusion based on transaction nodes.
2. **CoSimGNN-V**$_{rnn}$: Ablation of the recurrent neural network (RNN).

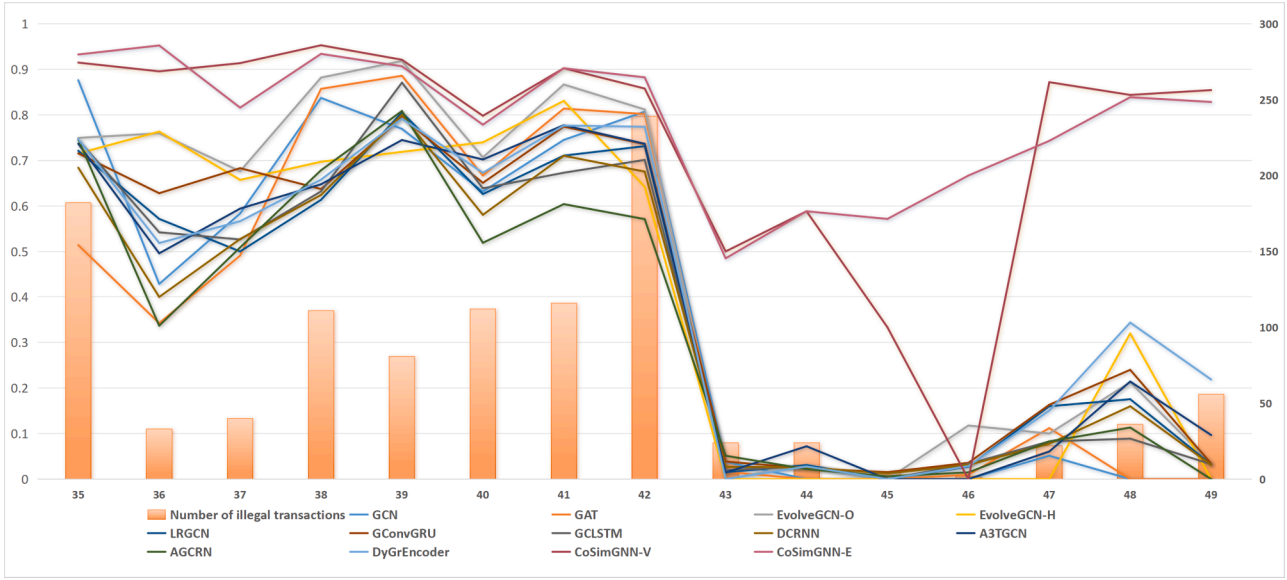

Fig. 7. F1 Scores and Number of Illicit Transactions Over Time. This plot combines a bar chart and a line graph to show the performance of different models in detecting illicit transactions over a series of time steps (35 to 49). The orange bars represent the number of illicit transactions at each time step. The various colored lines show the F1 scores of the CoSemiGNN variants and other Baseline models.

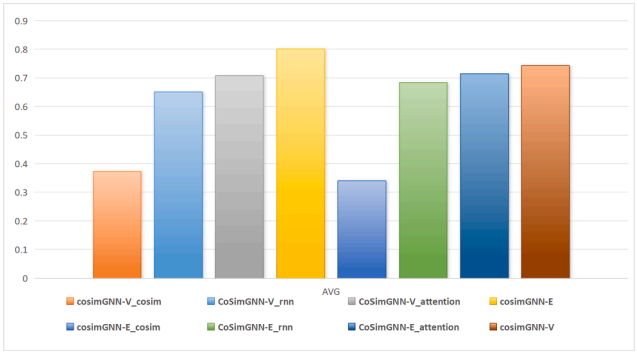

Fig. 8. Ablation Study of the CoSemiGNN Model. This bar chart shows the results of an ablation experiment to evaluate the contribution of different components to the overall performance of the CoSemiGNN model. The results demonstrate the importance of each module to the model's effectiveness.

3. **CoSimGNN-V**$_{attention}$: Ablation of global attention.
4. **CoSimGNN-E**$_{cosim}$: Ablation of the semi-supervised co-association fusion based on transaction edges.
5. **CoSimGNN-E**$_{rnn}$: Ablation of the recurrent neural network (RNN).
6. **CoSimGNN-E**$_{attention}$: Ablation of global attention.

Here, CoSimGNN-V and CoSimGNN-E refer to the co-association weighted aggregation of edges and the co-association feature aggregation of nodes, as previously described. The subscript $_{cosim}$ indicates the ablation of the semi-supervised mechanism; $_{rnn}$ indicates the ablation of the RNN mechanism in the model; $_{atte}$ refers to the ablation of the self-attention global feature extraction mechanism. The detailed experimental results are shown in the table below.

The experimental results are presented in Fig. 8. Observing the results, we find that the model performance significantly decreases after the ablation of $_{cosim}$, which clearly indicates that $_{cosim}$ plays a crucial role in improving the model's performance. This is because the CoSimGNN model integrates multiple semi-supervised methods, effectively incorporating semi-supervised information into the graph structure through co-association features. This strengthens the model's generalization ability to new data distributions, improving its prediction performance for emerging illicit transactions during time slices 43–49. Without $_{cosim}$, the model struggles to effectively predict illicit transaction patterns from new data distributions.

Further analyzing the results of the model after ablation of $_{rnn}$, we observe a moderate decline in performance, but it still significantly outperforms the model after $_{cosim}$ ablation. This is because the Recurrent Neural Network (RNN) plays a critical role in capturing the temporal dependencies of sequential data. Although the ablation of $_{rnn}$ affects the model's ability to capture temporal features, retaining $_{cosim}$ allows the model to still capture emerging illicit transaction patterns, somewhat mitigating the dynamic changes in illicit transaction patterns.

Finally, we observed an interesting phenomenon: the performance of the model after the ablation of $_{atte}$ is similar to that of the "CoSimGNN-V" and "CoSimGNN-E" models without ablation. The purpose of $_{atte}$ is to compute the global attention coefficients of transactions and apply self-attention to the global graph nodes. However, the experimental results show that while the self-attention mechanism has some effect on global feature extraction in transaction graphs, its impact is not significant. We propose a hypothesis: In blockchain transaction networks, illicit transactions actively disguise themselves as legitimate transactions, and the extreme imbalance between legitimate and illicit transactions makes it challenging for simple self-attention mechanisms to extract meaningful global features from the transaction graph.

Overall, the "CoSimGNN-V" and "CoSimGNN-E" models without ablation show the best performance in the charts, consistent with our expectations for the model. This further confirms the effectiveness of edge co-association weighting and node co-association feature aggregation in integrating the results of semi-supervised algorithms into the graph structure feature extraction process.

4.7. Hyperparameter sensitivity analysis

To evaluate the robustness and adaptability of the proposed model to key hyperparameters, we conducted sensitivity experiments on two important hyperparameters: (1) the number of base classifiers m in the semi-supervised module, and (2) the loss weighting coefficient α. The experimental results are shown in Figure X.

1. **Effect of the number of bases** m

Under the condition that other parameters remain unchanged, we gradually adjusted the number of basis functions m from 2 to 6. The bar chart on the left side of Fig. 9 shows that the F1 score steadily improves

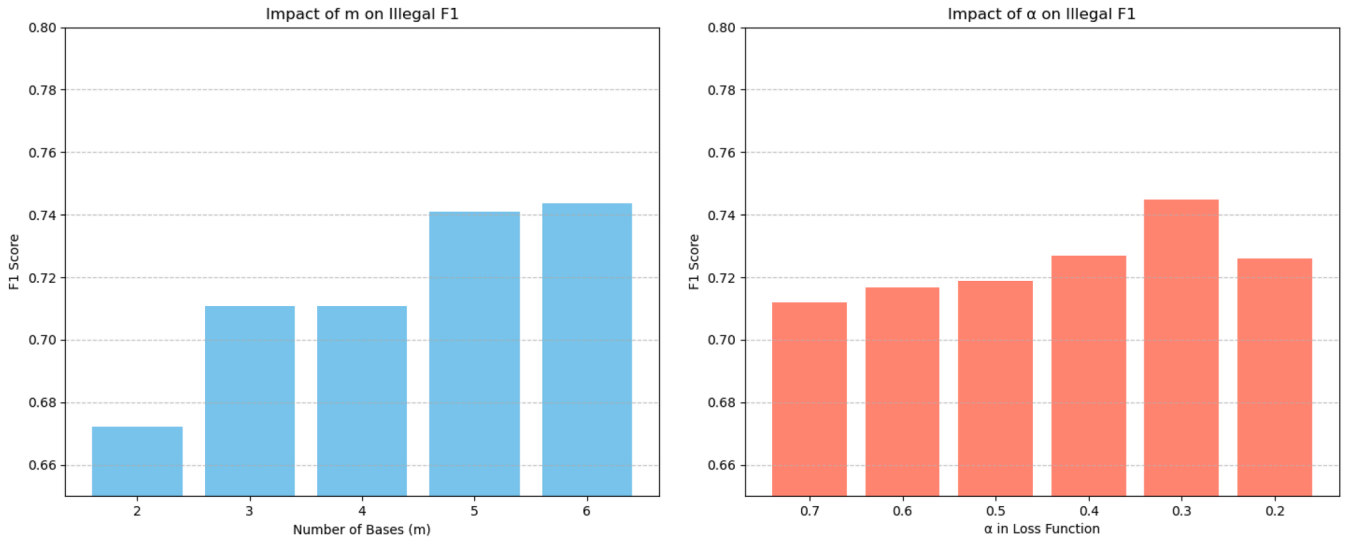

Fig. 9. The results of the hyperparameter sensitivity analysis are shown. The left graph represents the number of semi-supervised base class algorithms, and the right graph represents the value of the parameter *alpha* in the loss function.

as m increases, rising from 0.67 at $m = 2$ to 0.74 at $m = 6$. This indicates that appropriately increasing the number of basis functions helps the model better capture the latent relationships among semi-supervised nodes, thereby enhancing detection performance. However, a too-large m increases computational cost and may also lead to overfitting risks. We further observed that as m continues to increase, the model performance follows a logarithmic growth trend with a noticeably slowing rate of improvement. Based on the above analysis, it is recommended to set the initial value of m within the range $[4, 6]$, and flexibly adjust according to the specific task complexity and computational resources to achieve the best balance between performance and efficiency. In the experiments of this paper, $m = 5$ was ultimately chosen, achieving a good compromise between detection performance and computational efficiency.

2. **Effect of loss weighting coefficient** α

The coefficient α in the loss function (Eq. (15)) is used to balance the traditional supervised loss (BCE) and the semi-supervised consistency regularization loss (KL divergence) that we introduced. The value of α directly affects how the model utilizes information from labeled and unlabeled data. As shown in the bar chart on the right side of Fig. 9, the F1 score gradually increases as α decreases, reaching a peak of 0.7347 at $\alpha = 0.3$, and then slightly declines afterward. This indicates that giving a higher weight to the KL divergence term in the loss function (i.e., reducing α) helps the model learn a more robust intrinsic data distribution, thereby improving generalization. However, if the weight is too high (i.e., α is too small), the model may focus excessively on consistency of the unlabeled data and neglect the supervisory signal from labeled data. Therefore, we recommend tuning α within the range [0.2, 0.5]. In practical applications, starting from $\alpha = 0.3$ and adjusting based on performance on the validation set is advised.

3. **Confidence threshold** κ

The confidence threshold κ is used to filter semi-supervised pseudo-labeled samples, ensuring the quality of training data. It has been shown that if κ is set too low, the pseudo-label noise increases, leading to degraded model performance; if set too high, the sample size becomes insufficient, negatively affecting training effectiveness. Thanks to our ensemble learning architecture, we set κ to 0.85 to balance sample quality and quantity. In practical applications, we recommend dynamically adjusting κ within the range of 0.7 to 0.9 to adapt to different data conditions.

In summary, we conducted an in-depth analysis of the model's key hyperparameters. Experimental results clearly demonstrated the specific impact of each hyperparameter on model performance. Considering model complexity and computational resources, we proposed reasonable value ranges and adjustment recommendations. This work provides robust practical guidance for achieving optimal performance through systematic hyperparameter tuning across different datasets and application scenarios.

Table 3
Comparison of training time, parameters, and memory usage.

Model	Time/Epoch (s)	Params	Memory (MB)
EvolveGCN-O	0.2674	507,393	3,318.53
EvolveGCN-H	0.5365	507,649	1,580.89
LRGCN	1.4201	441,225	2,062.88
GConvGRU	0.9415	342,657	1,962.88
GC-LSTM	0.6356	244,609	2,818.29
DCRNN	0.6189	342,273	2,876.15
A3T-GCN	0.5052	244,354	1,136.88
AGCRN	**8.4500**	146,721	**11,539.82**
DyGrEncoder	0.2699	704,897	3,262.89
CMOS (Ours)	0.3313	561,409	1,318.88
SDGM (Ours)	1.5640	**1,255,297**	1,450.88

4.8. Computing resource analysis

We posit that, while pursuing high module accuracy, consideration of computational cost and resource consumption is crucial, as it directly affects a module's practicality and scalability. To analyze the performance of our proposed method in this respect, we compared it against a set of baseline modules on three core metrics: training time per epoch, total parameter count, and memory usage. The detailed results are shown in Table 3.

The CMOS module has a training time per epoch of 0.3313 s, a speed comparable to modules such as EvolveGCN-O (0.2674 s) and A3T-GCN (0.5052 s), and faster than LRGCN (1.4201 s) and AGCRN (8.4500 s). Its memory usage is 1,318.88 MB. The total number of parameters is 561,409, at a moderate level. Taken together, these figures indicate that the CMOS module achieves high efficiency while demanding relatively low computational resources.

For the SDGM module, although SDGM's parameter count (1,255,297) is the largest among all modules, its training time per epoch is 1.5640 s-longer than CMOS but still faster than AGCRN. This is primarily due to the multi-head attention mechanism in SDGM, which brings a substantial increase in parameter count; future work could consider

Table 4
Comparison of AML Methods in Terms of Dynamics, Supervision, and Evaluation Metrics. ✓ = Supported, × = Not Supported. FS = Fully Supervised, SS = Semi-Supervised, US = Unsupervised. Acc = Accuracy, P = Precision, R = Recall, F1 = F1-score.

Method	Graph Structure	Graph Structure Dynamics	Feature Dynamics	Supervision Type	Primary Evaluation Metrics
Farrugia et al. (2020)	×	×	×	FS	Acc, P, R, F1
Feldman et al. (2021)	×	×	×	FS	Acc, P, R
Weber et al. (2019)	✓	×	×	FS	Acc,Illegal-F1, F1,
Alarab et al. (2020)	✓	×	×	FS	Acc, P, R, F1, Illegal-F1
Tan et al. (2023)	✓	×	×	FS	Acc, P, R, F1, Illegal-F1
Chang et al. (2024)	✓	×	×	FS	Acc, P, R, F1
Pareja et al. (2020)	✓	✓	×	FS	Acc, F1
Mohan et al. (2023)	✓	✓	×	FS	Acc, P, R, F1
Li et al. (2019)	✓	✓	×	FS	Acc, P, R, F1
Seo et al. (2016)	✓	✓	×	FS	Acc, MSE
Chen et al. (2022)	✓	✓	×	FS	Acc, AUC
Li et al. (2017)	✓	✓	×	FS	MSE, MAE
Zheng (2022)	✓	✓	×	FS	Acc, F1
Jiang et al. (2023)	×	×	×	US	AUC, R
Wang et al. (2019)	✓	×	×	SS	AUC, F1, P, R
Sanjalawe and Al-E'mari (2023)	×	×	×	SS	P, R, F1, AUC
Wang et al. (2024a)	✓	×	×	SS	P, R, F1, AUC
Karim et al. (2024)	✓	×	×	SS	P, R, F1, AUC
Tang et al. (2022)	✓	×	×	SS	Acc, P, R, F1
Xue et al. (2023)	✓	×	×	SS	Acc, F1
Xiang et al. (2023)	✓	×	×	SS	F1, AUC
Our method	✓	✓	✓	**SS**	Illegal-F1, Acc, P, R, F1,

adopting more advanced attention mechanisms to capture global features while reducing resource consumption.

In summary, the difference in computational resource consumption between the CMOS and SDGM modules stems from their distinct architectural designs, but their resource requirements remain within an acceptable range for most application scenarios.

5. Discussion

To more comprehensively understand the experimental results of this study and their engineering implications, this section provides an in-depth discussion of the performance of CoSemiGNN in blockchain illicit transaction detection. Through a systematic analysis of the model architecture, observed experimental phenomena, and differences from baseline methods, we aim to uncover the fundamental factors underlying its performance advantages.

5.1. Comparison of methods

To systematically evaluate the practical contributions of this study, we conducted a comparative analysis between CoSemiGNN and a range of static and dynamic graph modules (see Table 4).

From the perspective of structural modeling, existing studies have widely recognized and successfully applied graph structures to characterize the complex relationships among entities. This abstraction of data into nodes and edges effectively captures the topological relationships between entities, thereby laying a solid foundation for in-depth analysis of relational data and the discovery of latent patterns.

From the perspective of dynamic modeling, real-world systems are continuously evolving. Some pioneering studies (e.g., Pareja et al. (2020), Mohan et al. (2023), Li et al. (2019)) have acknowledged this fact and have taken the lead in modeling the dynamic evolution of graph structures, achieving significant progress. However, the limitation of current research lies in the fact that most existing work focuses solely on the temporal evolution of structures, while generally neglecting the dynamics of node features, thus failing to realize joint dynamic modeling of both structures and features.

From the perspective of supervision paradigms, fully supervised learning, as the mainstream paradigm, demonstrates strong performance when sufficient labeled data are available. Meanwhile, to address the challenge of label scarcity in real-world scenarios, semi-supervised methods have emerged and proven their value. Nevertheless, the design of existing semi-supervised graph models has yet to effectively integrate and jointly model the dynamic evolution information of both structures and features, which limits their applicability in dynamic scenarios.

From the perspective of evaluation metrics, the vast majority of studies have adopted general-purpose metrics such as Accuracy, Precision, Recall, and F1-score (or regression metrics such as MSE and MAE), which can, to a certain extent, effectively evaluate model performance. Some domain-specific studies (e.g., Alarab et al. (2020), Tan et al. (2023)) have introduced customized metrics such as Illegal-F1 to enhance the precision of evaluation. In this study, such customized metrics are employed to more faithfully reflect the model's effectiveness in specific tasks.

We identify the following key characteristics that distinguish CoSemiGNN from existing methods:

1. **Problem setting closer to real-world scenarios**: Most approaches are based on a fully supervised setting, rendering them unable to capture illicit patterns from unlabeled data. In contrast, CoSemiGNN is built upon a semi-supervised assumption, enabling it to detect emerging illicit transaction patterns with only a small number of labels.
2. **More comprehensive dynamic modeling capability**: CoSemiGNN jointly models the dynamic evolution of graph structures and the temporal variation of node features, and employs a collaborative attention mechanism to integrate temporal contextual information. This contrasts with most semi-supervised methods (e.g., Wang et al., 2019), which do not fully exploit both structural and feature temporal dynamics.
3. **Targeted architectural design**: The model incorporates a temporal-awareness module and a dynamic correlation modeling module to enhance the detection of time-varying illicit behaviors.
4. **More fine-grained evaluation metrics**: Unlike most methods that rely solely on general-purpose metrics, CoSemiGNN adopts the task-specific Illegal-F1 metric for a more targeted performance evaluation of illicit behavior detection.

5. **Clear sources of performance improvement**: Co-SemiGNN achieves an average increase of 30 % in illicit transaction F1-score, primarily attributable to the introduction of feature dynamics modeling on top of existing transaction structure modeling, thereby enhancing model robustness.

Overall, CoSemiGNN is the only method in the table that simultaneously possesses structural dynamics, feature dynamics, and semi-supervised capability, offering higher adaptability and generalization potential in scenarios of dynamic blockchain transactions and label-scarce illicit detection tasks.

5.2. Analysis of key modules

We next analyze the specific roles of the semi-supervised Co-association mechanism, dynamic graph modeling structure, and attention mechanism within transaction graphs, thereby revealing how they jointly enhance the model's expressive capacity and generalization performance.

In the CMOS module, a key innovation lies in the strategic integration of multiple semi-supervised algorithms to capture emerging illicit transaction patterns. Traditional supervised models face severe challenges caused by shifts in feature distributions, while the decentralized and anonymous nature of blockchain transaction environments provides fertile ground for criminal activities. This is strongly evidenced by the failure of existing methods in the experimental results in Fig. 7. To adapt to the dynamic changes of illicit transactions, it is particularly necessary to effectively leverage unsupervised or semi-supervised methods to identify emerging transaction behaviors. Malicious actors often evade monitoring through disguising legitimate transactions or employing multi-hop transaction strategies, making it difficult for purely unsupervised approaches to accurately capture novel illicit patterns. However, we find that using a small number of known illicit transaction "anchors" can effectively guide the discovery of new illicit patterns-an ability lacking in existing dynamic graph neural networks. To better integrate semi-supervised signals into graph neural networks, CMOS constructs a collaborative relation node similarity matrix, naturally embedding Co-association relationships into the node aggregation process, thereby achieving effective adaptation to dynamic changes in transaction features.

The SDGM module inherits and extends the ideas of existing dynamic graph neural networks by employing an RNN to evolve GCN weights, enabling dynamic adaptation to temporal transaction data. Meanwhile, it incorporates a multi-head attention mechanism to capture global transaction features, compensating for the limitations of traditional GCNs in global information perception. Although this design increases model complexity, it significantly enhances overall performance. The emerging illicit patterns captured by CMOS and the persistent illicit patterns identified by SDGM work synergistically to form a continuously adaptive dynamic classifier, substantially improving the model's adaptability and generalization capability in illicit transaction detection.

Overall, the combination of CMOS and SDGM not only provides new theoretical support and practical solutions for illicit transaction detection in dynamic trading environments but also offers valuable references for the blockchain domain and other application scenarios involving complex dynamic graph anomaly detection, such as social networks and communication networks.

5.3. Limitations and practical challenges

Although CoSemiGNN demonstrated strong robustness on real blockchain transaction data, several notable limitations were identified in the study:

1. Scalability of large-scale graphs is a highly relevant and challenging task. In the computation process, the coassociation matrix computes the similarity between any two nodes, and its space complexity is $O(n^2)$ (where n is the number of nodes), which leads to significant space resource consumption when dealing with large-scale graphs. Future work could explore more efficient approximation algorithms or chunking computation strategies to reduce computational complexity and improve scalability for large graphs.

2. Noise in pseudo-labels remains an inherent issue, despite our use of higher confidence thresholds to improve pseudo-label quality when capturing emerging illicit transaction patterns. Therefore, developing more robust pseudo-label optimization mechanisms before constructing the Co-association matrix remains an urgent direction for future research.

3. Significant imbalance in transaction data categories. CMOS can capture illicit transaction patterns in unlabeled data using limited labeled information, alleviating the need for extensive labeled datasets to some extent. However, it has not fully addressed the prevalent class imbalance in blockchain transaction data (i.e., illicit transactions are far fewer than legal ones). Future work will focus on advanced imbalanced learning techniques to enhance the detection of minority classes (illicit transactions).

4. Computational overhead from global attention: While common GNN models tend to focus on local feature capture, introducing multi-head attention can capture global features but inevitably incurs additional computation due to full-graph attention. This increase in complexity may lead to higher demands on computational resources (e.g., GPU memory, processing time) during training and inference, which could pose practical challenges for real-time deployment in very large blockchain networks or resource-constrained environments. Balancing performance and computational efficiency will be a key direction for practical applications in the future.

6. Conclusion

This paper proposes a dynamic graph neural network based on co-association of semi-supervise (CoSemiGNN) for effectively detecting illicit transactions in blockchain systems under conditions of label scarcity and dynamic transactional changes. The model leverages semi-supervised learning to identify previously unseen illicit transaction patterns from unlabeled data, integrates semi-supervised signals into the transaction graph structure via Co-association relationships, and employs dynamic graph neural networks to capture temporal information of transactions, thereby adapting to evolving blockchain network environments. Experiments conducted on a Bitcoin transaction dataset demonstrate that CoSemiGNN outperforms conventional supervised learning methods in terms of precision, recall, and F1-score. Future work may explore complexity optimization for large-scale transaction graphs and data imbalance handling strategies, aiming to further advance illicit transaction detection techniques in blockchain systems.

CRediT authorship contribution statement

Yulong Wang: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing; **Qingxiao Zheng:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing; **Xuedong Li:** Methodology, Formal analysis, Writing – review & editing; **Lingfeng Wang:** Methodology, Formal analysis, Writing – review & editing; **Ling Lin:** Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This document is the results of the research project funded by the National Experimental Base for Intelligent Social Governance.

References

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020). Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In *Proceedings of the 2020 5th international conference on machine learning technologies* ICMLT '20 (p. 23-27). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3409073.3409080

Bai, J., Zhu, J., Song, Y., Zhao, L., Hou, Z., Du, R., & Li, H. (2021). A3t-GCN: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, *10*(7), 485.

Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, *33*, 17804–17815.

Bellei, C., Xu, M., Phillips, R., Robinson, T., Weber, M., Kaler, T., Leiserson, C. E., Chen, J. et al. (2024). The shape of money laundering: Subgraph representation learning on the blockchain with the elliptic2 dataset. arXiv preprint arXiv:2404.19109.

Chang, Z., Cai, Y., Liu, X. F., Xie, Z., Liu, Y., & Zhan, Q. (2024). Anomalous node detection in blockchain networks based on graph neural networks. *Sensors*, *25*(1), 1.

Chen, J., Wang, X., & Xu, X. (2022). GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. *Applied Intelligence*, *52*(7), 7513-7528. https://doi.org/10.1007/s10489-021-02518-9

Cholevas, C., Angeli, E., Sereti, Z., Mavrikos, E., & Tsekouras, G. E. (2024). Anomaly detection in blockchain networks using unsupervised learning: A survey. *Algorithms*, *17*(5), 201.

Christin, N. (2013). Traveling the silk road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd international conference on world wide web* (pp. 213–224).

Drugs, U. N. O. o., & Crime (2025). Inflection point: Global implications of scam centres, underground banking, and illicit online marketplaces in Southeast Asia. Technical Report UN.Office on Drugs and Crime. https://www.unodc.org/roseap/en/2025/04/inflection-point.html.

Elliptic (2023). What is blockchain analytics? https://www.elliptic.co/blockchain-basics/what-is-blockchain-analytics.

Elmougy, Y., & Liu, L. (2023). Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* KDD '23 (p. 3979-3990). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3580305.3599803

Farrugia, S., Ellul, J., & Azzopardi, G. (2020). Detection of illicit accounts over the ethereum blockchain. *Expert Systems with Applications*, *150*, 113318.

Feldman, E. V., Ruchay, A. N., Matveeva, V. K., & Samsonova, V. D. (2021). Bitcoin abnormal transaction detection based on machine learning. In *Recent trends in analysis of images, social networks and texts: 9th international conference, AIST 2020, Skolkovo, Moscow, Russia, october 15–16, 2020 revised supplementary proceedings 9* (pp. 205–215). Springer.

Holub, A., & O'Connor, J. (2018). Coinhoarder: Tracking a Ukrainian bitcoin phishing ring DNS style. In *2018 APWG symposium on electronic crime research (ecrime)* (pp. 1–5). IEEE.

Investigation, F. B. o. (2025). 2024 Internet crime report. Accessed: 2025-08-04 https://www.ic3.gov/AnnualReport/Reports/2024_IC3Report.pdf.

Jiang, S., Dong, R., Wang, J., & Xia, M. (2023). Credit card fraud detection based on unsupervised attentional anomaly detection network. *Systems*, *11*(6), 305.

Karim, M. R., Hermsen, F., Chala, S. A., De Perthuis, P., & Mandal, A. (2024). Scalable semi-supervised graph learning techniques for anti money laundering. *IEEE Access*, *12*, 50012–50029. https://doi.org/10.1109/ACCESS.2024.3383784

Li, J., Han, Z., Cheng, H., Su, J., Wang, P., Zhang, J., & Pan, L. (2019). Predicting path failure in time-evolving graphs. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* KDD '19 (p. 1279-1289). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3292500.3330847

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *Learning*, https://api.semanticscholar.org/CorpusID:3508727.

Mohan, A., PV, K., Sankar, P., Maya Manohar, K., & Peter, A. (2023). Improving anti-money laundering in bitcoin using evolving graph convolutions and deep neural decision forest. *Data Technologies and Applications*, *57*(3), 313–329.

Motamed, A. P., & Bahrak, B. (2019). Quantitative analysis of cryptocurrencies transaction graph. *Applied Network Science*, *4*(1), 131.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the thirty-fourth AAAI conference on artificial intelligence*.

Sanjalawe, Y. K., & Al-E'mari, S. R. (2023). Abnormal transactions detection in the ethereum network using semi-supervised generative adversarial networks. *IEEE Access*, *11*, 98516–98531. https://doi.org/10.1109/ACCESS.2023.3313630

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2016). Structured sequence modeling with graph convolutional recurrent networks. In *International conference on neural information processing*. https://api.semanticscholar.org/CorpusID:2687749.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., & Li, C.-L. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, *33*, 596–608.

Taheri, A., Gimpel, K., & Berger-Wolf, T. (2019). Learning to represent the evolution of dynamic graphs with recurrent models. In *Companion proceedings of the 2019 world wide web conference* WWW '19 (p. 301-307). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3308560.3316581

Tan, R., Tan, Q., Zhang, Q., Zhang, P., Xie, Y., & Li, Z. (2023). Ethereum fraud behavior detection based on graph neural networks. *Computing*, *105*(10), 2143-2170. https://doi.org/10.1007/s00607-023-01177-7

Tang, J., Zhao, G., & Zou, B. (2022). Semi-supervised graph convolutional network for ethereum phishing scam recognition. In *Third international conference on electronics and communication; network and computer technology (ECNCT 2021)* (pp. 369–375). SPIE (*vol. 12167*).

Team, C., (2025). Crypto scam revenue 2024: Pig.butchering grows nearly 40 % YoY as fraud industry leverages AI and increases in sophistication. Technical Report Chainalysis, Inc. https://www.chainalysis.com/blog/2024-pig-butchering-scam-revenue-grows-yoy/.

Wang, D., Lin, J., Cui, P., Jia, Q., Wang, Z., Fang, Y., Yu, Q., Zhou, J., Yang, S., & Qi, Y. (2019). A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International conference on data mining (ICDM)* (pp. 598–607). IEEE.

Wang, R., Zhang, Y., & Peng, L. (2024a). Anomaly detection service for blockchain transactions using minimal substitution-based label propagation. *IEEE Transactions on Services Computing*, *17*(5), 2054–2066. https://doi.org/10.1109/TSC.2024.3407601

Wang, Z., Ni, A., Tian, Z., Wang, Z., & Gong, Y. (2024b). Research on blockchain abnormal transaction detection technology combining CNN and transformer structure. *Computers and Electrical Engineering*, *116*, 109194. https://doi.org/10.1016/j.compeleceng.2024.109194

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. arXiv:1908.02591 [cs.SI], https://arxiv.org/abs/1908.02591.

Xiang, S., Zhu, M., Cheng, D., Li, E., Zhao, R., Ouyang, Y., Chen, L., & Zheng, Y. (2023). Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 14557–14565). (*vol. 37*).

Xiao, L., Han, D., Li, D., Liang, W., Yang, C., Li, K.-C., & Castiglione, A. (2023). CTDM: Cryptocurrency abnormal transaction detection method with spatio-temporal and global representation. *Soft Comput.*, *27*(16), 11647–11660.

Xue, R., Zhu, N., He, J., & He, L. (2023). Bitcoin transaction pattern recognition based on semi-supervised learning. *Journal of Computational Science*, *71*, 102055. https://doi.org/10.1016/j.jocs.2023.102055

Yu, L., Zhang, N., & Wen, W. (2021). Abnormal transaction detection based on graph networks. In *2021 IEEE 45th annual computers, software, and applications conference (COMPSAC)* (pp. 312–317). IEEE.

Zheng, Y. (2022). GRU-GAT model for blockchain bitcoin abnormal transaction detection. In *2022 IEEE conference on telecommunications, optics and computer science (TOCS)* (pp. 666–674). IEEE.