2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)

Money Laundering Activity Detection Method Based on Subgraph Pattern Search

Tong Chen*
College of Computer Science and Technology
Xi'an University of Science and Technology
Xi'an, Shaanxi, China
*Corresponding author's e-mail: 1165853045@qq.com

Shilong Wang
College of Computer Science and Technology
Xi'an University of Science and Technology
Xi'an, Shaanxi, China
e-mail: 2771841046@qq.com

***Abstract*—Given a financial transaction network, how can we detect money laundering activities within it? Money laundering organizations often use decentralized, multi-layered small transactions to hide the source of their illicit funds and employ sophisticated strategies to evade detection. Existing money-laundering detection methods for transaction networks either ignore transaction attributes only by considering topological similarity, or can only identify fixed chain transactions based on K-part graph. Therefore, this paper proposes a money laundering detection method based on pattern subgraph search. By analyzing the flow of funds, this method uncovers money laundering subgraphs hidden within the transaction flows. We model money laundering activities as pattern graphs and design a new scoring function to measure the degree to which the given subgraph conforms to the characteristics of money laundering. Experimental results show that this method outperforms decision trees, random forests, and naive Bayes in terms of precision, recall, and F1 score.**

Keywords-money laundering; graph structure; subgraph search

I. INTRODUCTION

Money laundering refers to the process of "cleaning" illegally obtained funds or assets through a series of complex financial operations, making them appear as if they have legitimate origins [1]. The primary goal of money laundering is to conceal the illicit funds obtained from criminal activities, preventing them from being traced or seized. Money laundering often involves criminal activities such as illegal goods trade and drug trafficking, and if not detected and dealt with promptly, it can pose a serious threat to financial security [2].

In the context of rapid advancements in information technology, financial institutions possess vast amounts of user information and transaction records, within which money laundering activities are often hidden. During the stages of fund transfer and integration, money laundering conceals the flow of funds through decentralized methods, posing a significant challenge to traditional monitoring approaches [3]. Graph-based approach, however, have a unique advantage in revealing relationships between users and can deeply explore complex associations. By applying graph-based structural analysis to large volumes of financial transaction data, money laundering subgraphs can be extracted, revealing patterns of fund flow, potential anomalies in the sources and destinations of funds, and the relationships between transactions, accounts, and entities associated with money laundering. Therefore, utilizing graph structures to analyze massive transaction data and constructing money laundering subgraphs has become a key factor in improving the efficiency of money laundering detection.

The current methods for detecting money laundering activities are as follows: Transaction account-based money laundering detection methods analyze each account independently. However, during the money laundering process, criminals employ various new techniques to evade detection, making it difficult for these methods to address complex laundering strategies, such as multi-account transfers and frequent small-value transfers. On the other hand, transaction network-based money laundering detection methods either focus solely on topological structure similarities while neglecting transaction attributes, or rely on K-partite graphs, which can only identify fixed chain-like transaction behaviors.

This paper proposes a money laundering detection method based on pattern subgraph search. We transform the problem of money laundering activities detecting into a subgraph search problem on a pattern graph. Since subgraph search is an NP-hard problem, traditional algorithms may face significant computational challenges as the size and complexity of the data increase, making it difficult to obtain an optimal solution within a reasonable time frame. Ant Colony Optimization (ACO) has shown excellent performance in solving NP-hard problems and exhibits strong dynamic adaptability, allowing it to efficiently explore large-scale and complex spaces. Therefore, we apply ACO to the subgraph search problem in this study.

The main contributions are as follows:

(1) Proposed a new definition of the money laundering subgraph: it is defined as a pattern graph, no longer limited to the previous fixed patterns, with multiple transactions allowed between the same pair of accounts.

(2) Proposed a money laundering activities detection method based on pattern subgraph search: This method considers both topological structure and transaction attributes, while eliminating the limitations of K-partite graphs, enabling it to detect a wider range of flexible money laundering transaction patterns.

(3) Effectiveness and robustness: Experiments demonstrate that this method can accurately capture money laundering activities, and remains effective even when longer transaction chains are used.

979-8-3315-0976-7/25/$31.00 ©2025 IEEE

II. Related Work

Existing money laundering detection methods can generally be divided into two main categories: one focuses on transaction accounts, while the other focuses on transaction networks.

A. Transaction Account-Based Detection Methods

Transaction account-based money laundering detection methods include rule-based methods and machine learning-based methods. Rule-based detection methods are the most fundamental approach, relying on expert-designed heuristic rules to determine whether money laundering suspicion exists [4, 5]. Rajput et al. proposed an ontology-based expert system that integrates domain knowledge and rules to identify suspicious transactions [6]. However, these methods are typically static and rigid, with a high dependence on human expertise. Once the rules are established, they are difficult to adapt to complex money laundering strategies.

Machine learning-based methods extract statistical features of accounts from different dimensions, analyzing large amounts of financial transaction data and performing pattern recognition to identify potential money laundering activities. Wang et al. proposed a community-oriented algorithm based on structural entropy minimization (SEM) and graph embedding to identify money laundering [7]. Ramadhan et al. combined criminology theory with machine learning to develop a machine learning framework to verify the effectiveness of money laundering [8]. Chen et al. combined graph convolu-tional network (GCN) and two-way Long short-term memory network (BiLSTM) to build a model for detecting money laundering activities in market transactions [9]. Wang et al. proposed GraphALM, an active learning model based on reinforcement learning, aimed at improving the detection performance of money laundering activities in blockchain transactions [10]. These methods allowmachinesto automa-tically learn transaction patterns associated with money laundering, reducing the dependence on human expertise compared to rule-based methods. However, both approaches analyze each account's attribute features and transaction behaviors independently, overlooking the transactional relationships between accounts.

B. Transaction Network-Based Detection Methods

Since data objects are interconnected, graph structures provide a powerful mechanism to capture these relationships. Chen et al. constructed a bipartite graph network, mapping the features of vertices to the source and target sets, and used burstiness and connectivity patterns to detect anomalies [11]. Mu et al. demonstrated the effectiveness of using transaction networks and feature information for financial fraud detection [12]. Wang et al. constructed a transaction network and used the properties of the Egonet model to identify and label anomalous behaviors in electronic banking systems [13]. Liu et al. proposed an improved graph embedding algorithm for money laundering detection [14]. Ren et al. applied one-sided node sampling on a bipartite graph, decomposing the original graph into smaller subgraphs and selecting the top e-commerce fraud subgraphs [15]. Li et al. referred to star-like structures (where a central node is connected to multiple peripheral nodes, potentially representing a central account controlling multiple accounts) as "volcanoes" and "black holes," or other similar structures in anti-money laundering work, which may indicate the beginning or end of suspicious fund flows [16].

Eswaran et al. proposed a SpotLight algorithm that uses random sketching techniques to identify abrupt changes in graph structures, such as the sudden disappearance or emergence of dense subgraphs [17]. Hu et al. identified multiple bitcoin mixing services through graph classification [18]. Cheng et al. detected money laundering activities by finding similar nodes based on deep map learning [19]. Li et al. constructed chain transaction structures based on dense subgraphs to detect anomalous transaction flows between banks [20]. Akartuna et al. visualized money laundering as a network, generating policy-relevant preventive insights through holistic network analysis [21]. Ouyang et al. gathered transaction subgraphs of the same class together and separated subgraphs of different classes to detect bitcoin money laundering through subgraph comparison learning [22]. However, the graph method combined with deep learning has poor interpretability, other methods either only consider the topological similarity of subgraphs without specifically considering transaction attribute features, or they focus solely on detecting anomalies in K-partite graphs, which limits the length of the transaction chains that can be detected and fixes the search patterns.

III. Problem Formulation

A. Basic Terminology

G: Transaction network diagram $G=(V,E,W,T)$, including the transaction amount and transfer time for each node.

V: The node set of graph G, i.e., $S \cup M \cup D$

$\boldsymbol{E_{ij}}$: The transaction edge between two trading accounts in the graph.

a: The difference between the total outbound edge weight of the origin account and the total inbound edge weight of the destination account.

b: The difference between the sum of the inbound and outbound edge weights of the intermediary account node.

S,M,D: Originating account node; Intermediary account node; Destination account node.

Δ***min***: The minimum transfer time of the destination account node minus the minimum transfer time of the originating account node.

Δ***max***: The maximum transfer time of the destination account node minus the maximum transfer time of the originating account node.

B. Key Features

In the process of money laundering, criminals often use multiple intermediary accounts for transfers between the starting account and the destination account in order to evade detection, thus forming a chain of financial transactions. We assume that money laundering activities exhibit the following characteristics:

- *Local balance of payments:*

In money laundering activities, the illicit funds start from the origin account node v_s, pass through multiple intermediary account nodes v_m, and ultimately flow to the destination account node v_d. The vast majority of funds received by the intermediary account node v_m are transferred to the destination account node v_d, thus the intermediary account node v_m remains in a balanced state throughout the process.

- *Transfer time is close:*

The longer illicit funds remain in intermediary accounts, the greater the risk of being frozen. Therefore, criminals will transfer the funds out of the intermediary account nodes as quickly as possible. As a result, the time span for transfers between the origin account nodes and the destination account nodes is relatively short.

C. Problem Definition

The problem of detecting money laundering transaction patterns in a transaction network is essentially a pattern subgraph search problem, i.e., searching for subgraphs in the entire transaction network graph that match the characteristics of money laundering activities.

Therefore, the problem of detecting money laundering activities is defined as follows: Given a transaction network pattern graph G, the goal is to identify subgraphs $G' = (V,E,W,T)$ of G that satisfies the aforementioned characteristics. In this subgraph, V represents the set of nodes, E represents the set of edges, W represents the inbound and outbound edge weights of each node in the subgraph, and T represents the fund transfer-in and transfer-out times of each node in the subgraph.

1) The formula for the account balance score:

$$a = \left| \sum_{e\in(s,m)\wedge(s,d)} e_{ij} - \sum_{e\in(m,d)\wedge(s,d)} e_{ij} \right| \tag{1}$$

$$b = \left| \sum_{e\in(s,m)} e_{ij} - \sum_{e\in(m,d)} e_{ij} \right| \tag{2}$$

2) Transfer time scoring formula:

$$S_{t1}(\Delta min, \Delta max) = \begin{cases} \frac{1}{1+\Delta min} + \frac{1}{1+\Delta max} & if\ \Delta min>0 and \Delta max>0 \\ \beta*(\Delta min^{\wedge 2} + \Delta max^{\wedge 2}) & if\ \Delta min<0 and \Delta max<0 \end{cases} \tag{3}$$

$$S_{t2}(\Delta min, \Delta max) = \begin{cases} \frac{1}{1+\Delta min} + \beta*\Delta max^{\wedge 2} & if\ \Delta min>0 and \Delta max<0 \\ \frac{1}{1+\Delta max} + \beta*\Delta min^{\wedge 2} & if\ \Delta min<0 and \Delta max>0 \end{cases} \tag{4}$$

3) Overall aggregation formula:

$$Score = \frac{\alpha * m * n}{\sqrt{a^2 + b^2} + 1} + St \tag{5}$$

The smaller the difference between the total outbound edge weight of the origin account and the total inbound edge weight of the destination account, the more suspicious the subgraph is; similarly, the smaller the difference between the total inbound and outbound edge weights of the intermediary accounts, the more suspicious the subgraph is. m and n represent the number of nodes and edges, respectively.

IV. Method

The goal of money laundering activity detection is to identify all subgraphs that satisfy the above characteristic conditions, which is an NP-hard problem. The ant colony algorithm can effectively find subgraphs that match a specific pattern in complex graphs, adapt to dynamic changes, and avoid local optima. Therefore, the ant colony algorithm is used to solve the above problem.

A. Model Diagram Overview

Figure 1 illustrates subgraphs in the transaction network that exhibit characteristics of money laundering. Different colors correspond to multiple subgraphs. "Initial account" represents the starting account layer, "Intermediate account" represents multiple intermediate account layers, and "Destination account" represents the destination account layer. Funds flow out of the initial account layer, pass through the intermediate account layers to the destination account layer, or directly flow into the destination account layer.

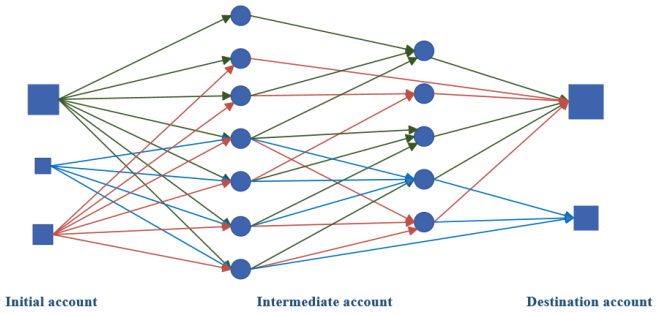

Figure 1 Money laundering activity pattern subgraph

B. Ant Colony Algorithm-based Method

The ant colony algorithm utilizes the mechanism by which ants influence each other through the release of pheromones while searching for food, in order to find the optimal or near-optimal solution. In this paper, pheromone is defined as the degree to which a subgraph conforms to money laundering characteristics. First, multiple initial subgraphs are constructed based on the constraints of the transaction network's layers and the number of nodes in each layer. Each subgraph is then scored, and pheromone is updated. The pheromone update includes two main parts: pheromone enhancement and pheromone evaporation. When ants find subgraphs that match the money laundering characteristics, they leave more pheromone on those subgraphs. The higher the pheromone concentration, the more suspicious the subgraph is. After global pheromone update, it guides the ants to explore new subgraphs. Through continuous iterations of subgraph scoring, pheromone updating, and re-building subgraphs, the pheromone concentration gradually concentrates on high-quality paths, guiding the algorithm toward the global optimal solution.

Based on the above idea, the money laundering activity detection method proposed in this paper using the ant colony algorithm is shown in Algorithm 1. The input of the algorithm is the transaction network *G*, which includes attributes such as the node set, edge set, transfer amounts, and transfer times. The output is all the subgraphs $G'=(V,E,W,T)$ that satisfy the money

laundering characteristics, where each subgraph consists of the origin account nodes, intermediary account nodes, destination account nodes, as well as the transaction edges between them.

Algorithm 1: Ant Colony

Input: Graph G

Output: subgraphs satisfying the characteristics of money laundering: *G'=(V,E,W,T)*

1. InitG ← Init(Transaction network)
2. while IterationNum > 0 do
   a) Calculate_Score_G ← CalculateScore (G)
   b) Update_Phe_G ←Update_Phe(Score, DR,PR)
   c) Subgraphs ← Construct_Subgraph(Update_Phe_G)
3. return *G'*

V. Experiments And Results

A. Experimental Environment

The experiments in this paper were conducted on a PC configured with an Intel Core i7-8570H processor, 16GB RAM, and a 64-bit Windows 10 operating system. The experiments for detecting money laundering activities based on pattern subgraphs were implemented using Python 3.9.

B. Dataset

Due to the involvement of account privacy in financial datasets, obtaining real data is challenging. To validate the effectiveness of the method, this paper uses a simulator to generate synthetic transaction data, which includes five attributes: origin node, target node, amount, transaction time, and label. The dataset labels suspicious accounts as 1 and normal accounts as 0, with a time range from February 21, 2023, to August 7, 2023. A total of five sets of data were generated, and the specific parameters are shown in Table I.

TABLE I. THE DESCRIPTION OF DATASET

Node	Edge	Timestamp
0.5K	966	168days
1.0K	1654	168days
1.5K	2155	168days
2.0K	2977	168days
2.5K	3486	168days

C. Evaluation Index

We treat the money laundering activity detection problem as a binary classification problem, with the goal of identifying normal subgraphs and suspicious subgraphs. To evaluate the effectiveness of the method, this paper uses the following three metrics to assess the performance of the proposed method:

1) Precision: The proportion of actual positive instances (such as suspicious accounts) among those predicted as positive by the model.

2) Recall: The proportion of actual positive instances correctly identified as positive by the model out of all the actual positive samples.

3) F1_score: Accuracy and recall are neutralized for comprehensive evaluation of model performance.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1\_score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

The confusion matrix is shown in Table II:

TABLE II. CONFUSION MATRIX

Actual/Predicted	Positive	Negative
Positive	**TP**	**FN**
Negative	**FP**	**TN**

D. Experimental Result

The subgraph of a typical money laundering pattern found is shown in Figure 2. Because the subgraph is generated step by step under the constraints of the number of layers and the number of nodes under the layer, the outgoing connection of each account node is not fixed. For example, the start account node can point directly to the destination account node, and the intermediate account node v_2 does not have to point to the next intermediate account node, but can also point directly to the destination account node layer, or other account node layers, and the number of intermediate account layers is not fixed. In addition, the transfer time between the account nodes is relatively close, and the outgoing and incoming weights of the intermediate account nodes are roughly the same.

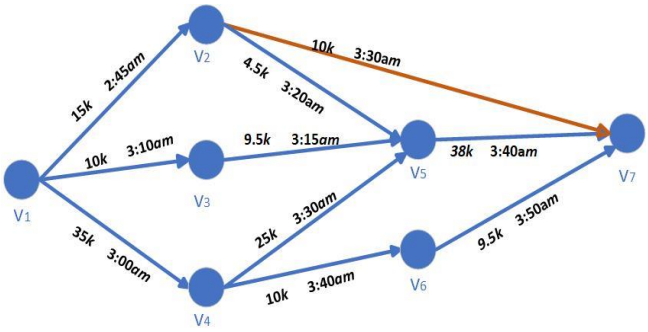

Figure 2 Money laundering features simplified subgraph

Next, the validity and stability of the proposed method on the simulated data set are analyzed. Decision tree, Naive Bayes, and random forest algorithms are used for comparison. The proposed method and the comparison method were run independently for 10 times. The results of the ant colony algorithm are obtained through multiple iterations to provide

the final outcome, while the best result from the 10 runs of decision tree, Naive Bayes, and random forest is selected for comparison. The relevant parameters of the ant colony algorithm are shown in Table III.

In the experiments with the decision tree model, Naive Bayes model, and random forest model, the test set is designated as 25% of the total dataset, with the remaining 75% used as the training set, as shown in Table IV. The following two features are used to construct the decision tree, Naive Bayes, and random forest models:

(1) Calculate the average of the time difference for transfers in and transfers out in the subgraph.

(2) The difference in inbound and outbound edge weights of intermediary account nodes compared to the inbound and outbound edge weights of the origin and destination accounts.

TABLE III. RELATED PARAMETERS OF ANT COLONY ALGORITHM

Node	Subs	Iterations	Decay	Increase	α	β
0.5K	10	10	0.05	10	0.7	0.6
1.0K	10	10	0.05	10	0.7	0.6
1.5K	15	10	0.05	10	0.7	0.6
2.0K	15	20	0.05	10	0.7	0.6
2.5K	15	20	0.05	10	0.7	0.6

TABLE IV. MACHINE LEARNING METHOD PARAMETERS

Node	Label	Testset
0.5K	20	0.25
1.0K	25	0.25
1.5K	32	0.25
2.0K	35	0.25
2.5K	40	0.25

The generated datasets for each group are used as the x-axis, with Precision, Recall, and F1_score as the y-axes, to evaluate the detection performance of the proposed algorithm on different datasets. The experimental results are shown in Figures 3, 4, and 5.

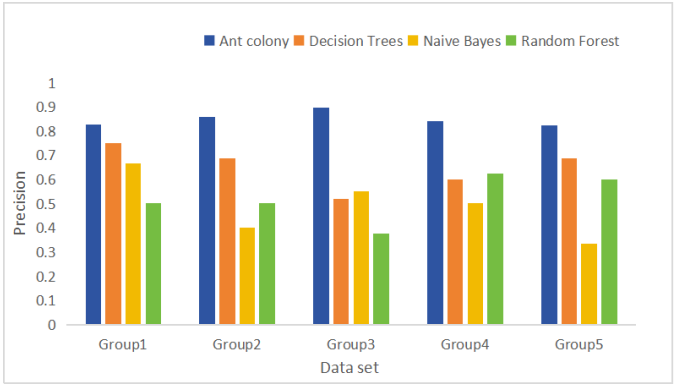

Figure 3 The precision of ant colony algorithms, decision trees, naive Bayes and random forests

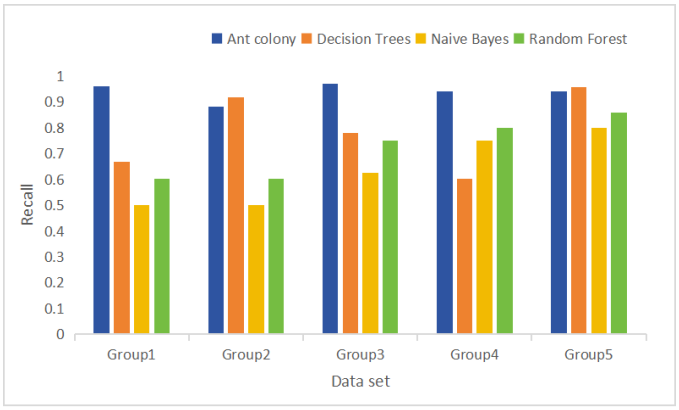

Figure 4 The recall of ant colony algorithms, decision trees, naive Bayes and random forests

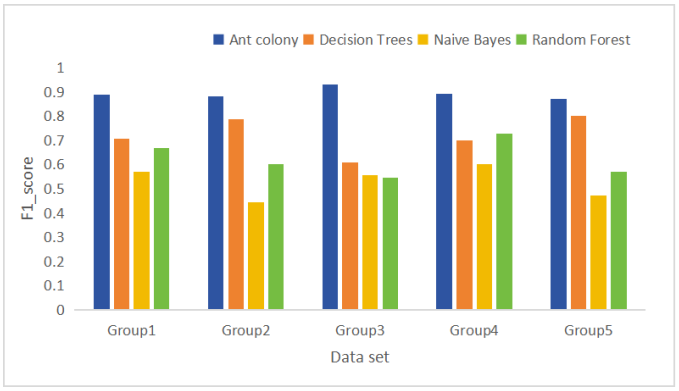

Figure 5 The F1_Score of ant colony algorithms, decision trees, naive Bayes and random forests

The experimental results indicate that the method proposed in this paper can accurately identify subgraphs of money laundering activities, outperforming decision trees, Naive Bayes, and random forests across all metrics. The analysis shows that, under the same parameters, decision trees and random forests perform better than Naive Bayes in terms of recall and precision. This is mainly due to the strong independence assumption of features in Naive Bayes, which significantly affects its performance.

Stability: Since the subgraph score is closely related to the path selection, it reflects the quality of the solution. Therefore, the following will analyze the dynamic change of the neutron graph fraction during the algorithm recognition process. The horizontal coordinate represents the number of iterations, and the vertical coordinate represents the maximum fraction of the subgraph, as shown in Figure 6. The results show that in the first few iterations of the algorithm, the score may fluctuate due to the instability of the initial pheromone update. With the progress of iteration, the score gradually increases and tends to the optimal solution.

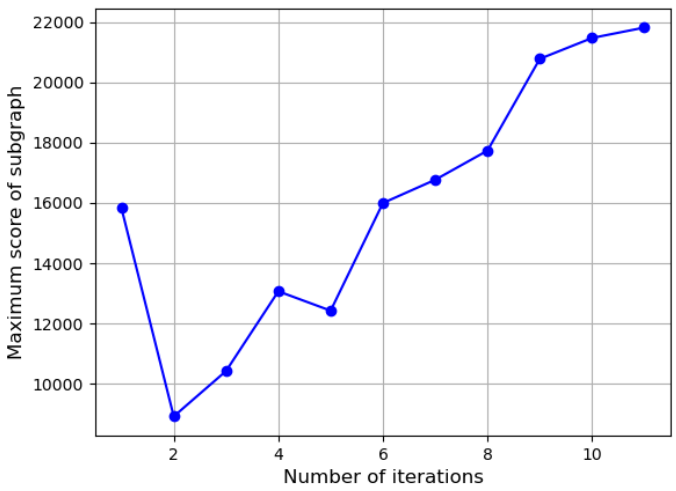

Figure 6 The dynamic change of the maximum score of the subgraph

VI. Conclusion

This paper proposes a money laundering detection method based on pattern subgraph search, transforming the money laundering detection problem into a subgraph search problem. The ant colony algorithm is used to filter subgraphs on the modeled transaction network, iterating multiple rounds to identify subgraphs that match the characteristics of money laundering activities. Experimental results show that, compared to Naive Bayes, decision tree, and random forest algorithms, the proposed method performs well in terms of precision, recall, and F1-score, while also providing interpretable results. Additionally, the method has been validated for its effectiveness on longer transaction chains, offering a new perspective for future detection of money laundering activities.

References

[1] Ferwerda J, Reuter P. National assessments of money laundering risks: Stumbling at the start[J]. Risk Analysis: An International Journal, 2024, 44(9).

[2] Catherine, Denny, Shihab M. Bank account classification for gambling transactions[C]// 2021 3rd East Indonesia Conference on Computer and Information Technology. IEEE, 2021: 302-308.

[3] Mahootiha M, Golpayegani A H, Sadeghian B. Designing a new method for detecting money laundering based on social network analysis[C]//2021 26th International Computer Conference, Computer Society of Iran (CSICC). IEEE, 2021: 1-7.

[4] Flowers E, Dua P, Balota E, et al. Heuristic money laundering detection engine: U.S. Patent 10, 832, 249[P]. 2020-11-10.

[5] Alkhayer, J.Y. and Gupta, C.M. (2024), "Arbitrators' suspicion of money laundering: choices against principles", Journal of Money Laundering Control, Vol. 27 No. 3, pp. 459-471.

[6] Rajput Q, Khan N S, Larik A, et al. Ontology based expert-system for suspicious transactions detection[J]. Computer and Information Science, 2014, 7(1): 103.

[7] Wang, S., Wang, P., Wu, B. et al. Structural entropy minimization combining graph representation for money laundering identification. Int. J. Mach. Learn. & Cyber. 15, 3951–3968 (2024).

[8] Ramadhan, S. (2024), "Harnessing machine learning for money laundering detection: a criminological theory-centric approach", Journal of Money Laundering Control, Vol. ahead-of-print No. ahead-of-print.

[9] Chen, Y., Du, M. Financial Fraud Transaction Prediction Approach Based on Global Enhanced GCN and Bidirectional LSTM. Comput Econ (2024).

[10] Wang Q, Tsai W T, Shi T. GraphALM: Active Learning for Detecting Money Laundering Transactions on Blockchain Networks[J]. IEEE Network, PP[2024-12-31].

[11] Chen, Z. and Sun, A. (2020) Anomaly Detection on Dynamic Bipartite Graph with Burstiness. In: IEEE International Conference on Data Mining (ICDM). Italy. pp. 966-971.

[12] Huang, D., Mu, D., Yang, L., et al. (2018) CoDetect: Financial Fraud Detection With Anomaly Trait Detection. In: IEEE Access. pp. 19161-19174.

[13] Wang, Yuan, L. Wang, and J. Yang. "Egonet based anomaly detection in E-bank transaction networks." 2020:012038.

[14] Liu J, Yin C, Wang H, et al. Graph embedding-based money laundering detection for Ethereum[J]. Electronics, 2023, 12(14): 3180.

[15] Ren, Yuxiang, et al. "EnsemFDet: An Ensemble Approach to Fraud Detection based on Bipartite Graph." (2019).

[16] Z. Li, H. Xiong, Y. Liu, and A. Zhou, ''Detecting blackhole and volcano patterns in directed networks, '' in Proc. IEEE Int. Conf. Data Mining, Dec. 2010, pp. 294–303.

[17] Eswaran, Dhivya, et al. "SpotLight: Detecting Anomalies in Streaming Graphs." ACM (2018).

[18] Xiaoyan Hu, Meiqun Gui, Guang Cheng, Ruidong Li, Hua Wu, Multi-class Bitcoin mixing service identification based on graph classification, Digital Communications and Networks, Volume 10, Issue 6, 2024, Pages 1881-1893, ISSN 2352-8648.

[19] D. Cheng, Y. Ye, S. Xiang, Z. Ma, Y. Zhang and C. Jiang, "Anti-Money Laundering by Group-Aware Deep Graph Learning, " in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12444-12457, 1 Dec. 2023.

[20] Li, Xiangfeng, et al. "FlowScope: Spotting Money Laundering Based on Graphs." 2020:4731-4738.

[21] Akartuna E A, Johnson S D, Thornton A. A Holistic Network Analysis of the Money Laundering Threat Landscape: Assessing Criminal Typologies, Resilience and Implications for Disruption[J]. Journal of Quantitative Criminology:1-42[2024-12-31].

[22] Ouyang S, Bai Q, Feng H, et al. Bitcoin Money Laundering Detection via Subgraph Contrastive Learning[J]. Entropy, 2024, 26(3).