Anti-Money Laundering by Group-Aware Deep Graph Learning

Dawei Cheng, Yujia Ye, Sheng Xiang, Zhenwei Ma, Ying Zhang and Changjun Jiang

Abstract—Anti-money laundering (AML) is a classical data mining problem in finance applications. As well known, money laundering (ML) is critical to the effective operation of transnational and organized crime, which affects a country's economy, government, and social wellbeings. Financial services organizations facilitate the movement of money and have been enlisted by governments to assist with the detection and prevention of money laundering, which is a key tool in the fight to reduce crime and create sustainable economic development. In the application of AML, user identity and financial behavior data are widely used to detect laundering transactions. In recent years, an increasing number of money laundering activities have been conducted by organized criminal gangs while most existing works still treat the actions of each account as independent identity behavior without considering the group-level conspired interactions. Therefore, in this paper, we propose a group-aware deep graph learning-based approach for organized money-laundering detection. In particular, we design a community-centric encoder to represent the nodes and attributes in user transaction graphs and derive the adjacent gang behaviors. Then, we devise a scheme of local enhancement to accommodate nodes with similar transaction features, which are aggregated into gangs for downstream detection. Extensive experiments on the real-world dataset from one of the largest bank card alliances worldwide show that our proposed method outperforms state-of-the-art methods in both offline and online modes, showing the effectiveness of money laundering detection with group-aware deep graph learning.

Index Terms—money Laundering; data mining; graph neural network; graph learning

1 Introduction

Money laundering is the process of changing large amounts of money obtained from crimes, such as drug trafficking, into origination from a legitimate source. It is a key path for criminals to disguise their illegal origin, which enables them to enjoy these profits without jeopardizing their source [1], [2]. A meta-analysis by United Nations Office on Drugs and Crime estimates that the total amount of money laundered through the financial system is equivalent to about 2.7 percent of global gross domestic product [3]. A financial crime of such an extent is a serious threat to societies and economies all over the world [4]. Therefore, fighting against money laundering is crucial significance to the long-term health of national financial stability and international business safety.

The financial industry has developed anti-money laundering solutions since the beginning of the twentieth-century [5], [6]. Traditional approaches focus on legislative considerations and compliance requirements, which is time-consuming and labor-intensive as money laundering is a sophisticated and constantly updated activity [7], [8]. Most AML approaches in the market includes: 1). rule-based [7], which makes decisions using general sets of predefined rules and thresholds [9]; 2). learning-based [10], [11], i.e. inferring the risk probability by classical machine learning approaches, such as logistic regression, support vector machine, etc [12]. But classical machine learning methods face significant challenges in detecting human brain-armed money laundering behavioral patterns because most of these models have very limited parameter capacity which may lead to suboptimal performance in detecting conspiracy actions.

In recent years, financial services organizations that facilitate the movement of money have been enlisted by regulators to assist with the AML task. Therefore, various deep learning-based models are possible to be leveraged for money laundering detection with massive transactional behavior data available, including convolutional feature learning [13], sequence learning [14] and graph pattern learning [15]. For example, Mubalaike and Adali [16] developed an ensemble method by combining decision trees with a stacked auto-encoder (SAE) and restricted Boltzmann machine (RBM) for suspicious transaction detection. Weber et. al., [17] utilizes Fast-Graph Convolutional Networks (FGCN) to learn graph representations of transactions for AML, which improves the detection performance.

Despite these approaches achieving remarkable successes, we also observed that existing methods treat each identity as an independent account without considering the group-level (gang-level) interactions. For money laundering criminals, group behaviors are normally conducted. Figure 1 presents a typical process of this gang-level organized criminal. The lockstep criminals frequently manipulate a group of accounts to diffuse a large amount of

- *The work is supported by the National Key R&D Program of China (2022YFB4500205), the National Natural Science Foundation of China (62102287) and the Shanghai Science and Technology Innovation Action Plan Project (22511100700)*
- *Dawei Cheng, Yujia Ye, and Changjun Jiang are with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Shanghai Artificial Intelligence Laboratory, Shanghai, China E-mail: {dcheng, 1853769, cjjiang}@tongji.edu.cn*
- *Sheng Xiang and Ying Zhang are with School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia. Email: {sheng.xiang, ying.zhang}@uts.edu.au*
- *Zhenwei Ma is with the Research and Development Center of Financial Safety, China UnionPay Co., Ltd., Shanghai, China. E-mail: mazhenwei@unionpay.com*

(Corresponding author: Sheng Xiang.)
Manuscript received 8 Jul. 2022; Revised 28 Jan. 2023; Accepted 23 Apr. 2023.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2023.3272396

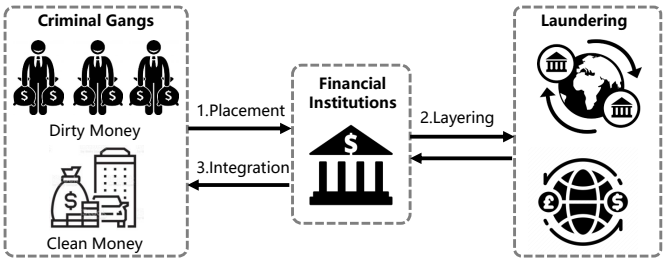

Fig. 1. The procedure of money laundering. (1) Placement: moving a bulk of dirty money into the financial system. (2) Layering: "laundering" on each transaction, such as trading in different user accounts, currencies or stocks across different markets. (3) Integration: retrieving the illicit funds in a legal way.

money in different channels and thus derive these transactions with over-consistent group patterns. Our intuition is also proved by a real-world dataset as shown in Figure 2. We construct the user transaction graph first, in which we color criminal nodes in red. As we can see, the criminal patterns are obvious in the group-level structures. Inspired by these observations, we believe the group interactions among accounts that are built from the graph might be essential to detecting money laundering transactions.

To this end, we devise a group-aware (gang-aware) graph neural network-based approach (GAGNN) for organized money laundering transaction detection. In particular, we design a community-centric encoder to transform the original transactions into graphs and proceed to encode the graph using both topological and attribute-wise information. Then, we devise a scheme of the local enhancement layers to accommodate nodes with similar transaction features, which are aggregated into groups to learn the organized behavior patterns. Finally, a joint optimization strategy is adopted to infer the suspicious probability of money laundering transactions. We conduct extensive experiments to evaluate the effectiveness of our proposed method on the large-scale real-world dataset of UnionPay, one of the largest bank card alliances worldwide. The results demonstrate the superior performance of GAGNN in detecting individual and organized suspicious money laundering transactions. The main contributions of the paper are summarized as follows:

1) We address a crucially important financial criminal problem with a data-driven graph learning approach. Our work paves a new way for group-aware detection in addressing the money laundering threat to the financial industry.
2) We design and implement the group-aware deep graph learning approach, which enables the model to learn from user transaction graphs directly. We also propose a community-centric encoder, local enhancement layer, and prediction network and prove its effectiveness in overcoming organized money laundering threats.
3) We thoroughly evaluate the proposed approach by comparing it with the existing benchmarks on the historical dataset and achieve state-of-the-art performance. In addition, we conduct empirical studies in real-world money-laundering cases, and the result proves our method could prevent potential financial crime for the long-term health of the economic environment and sustainable society.

The rest of the paper is organized as: Section 2 describes the business background and data observation. Section 3 shows the proposed model in detail. We report the experiment results and case studies in Section 4. Section 5 surveys the related work. Conclusion and discussion are described in Section 6.

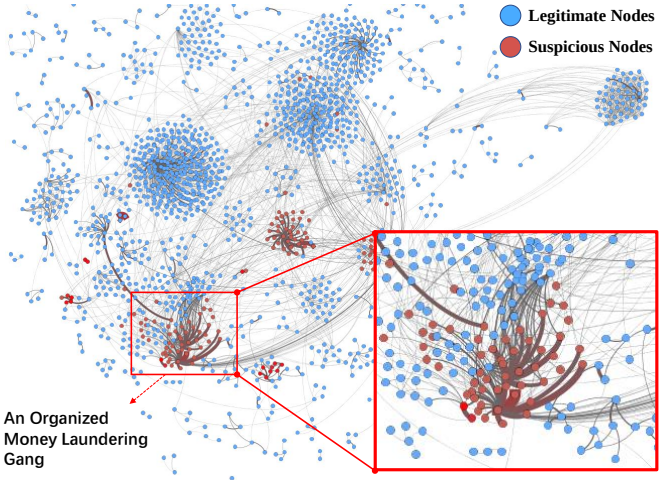

Fig. 2. The layout of a real-world ML user transaction graph. The records being reported as high risk of money laundering are colored red. The legitimate ones are colored blue.

2 Preliminaries

2.1 Background Knowledge

Money laundering is a term used to describe the process of taking funds generated from illegal activities and making them legitimate and clean. Figure 1 illustrates typical procedures of money laundering: 1) Placement stage. In this stage, criminals move a bulk of dirty money into the financial system, which is the most vulnerable stage that may attract the attention of AML agencies. 2) Layering stage. It breaks the funds into various small transactions and employs "laundering" on each transaction in order to make it difficult to be detected, such as trading in different currencies or stocks across different markets. 3) Integration stage. Multiple "laundered" funds are now returned to the criminals legitimately, which means they retrieve their illicit funds in a legal way after fully integrating them into a legitimate source, and are able to use them for any purpose.

As we can see, financial institutions play an important role in ML, which is the main channel for the placement, layering, and integration stages. Meanwhile, these massive illicit transactions are also recorded by IT systems in financial institutions, which makes it possible to detect money laundering suspicious transactions by advanced big data-driven analysis techniques. Therefore, this paper explores moving a step forward and reports our observations and findings on data-driven anti-money laundering combat using a group-aware graph neural network.

2.2 Observations from User Transaction Graphs

In this section, we report our observations on user transaction graphs via data-driven analysis. Formally, a user transaction graph (shorted as graph in the rest of this paper) G is composed of users as nodes V and transactions as edges E. Figure 2 shows the layout of a real-world graph with over 2,100 users and 5,000 transactions, in which 287 records are reported as high risk of money laundering and we colored them red. The legitimate ones are colored blue. The structure of this graph is very complex with many nodes, but still, one transaction pattern can be seen. There's a cluster-like transaction pattern: the graph is clustered into various groups, and these clusters may be all normal user nodes, or most of them may be nodes that have been involved in money laundering, and the red nodes have a higher risk of money laundering transactions with each other. If each cluster of nodes on the graph is considered as a group, the groups on

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2023.3272396

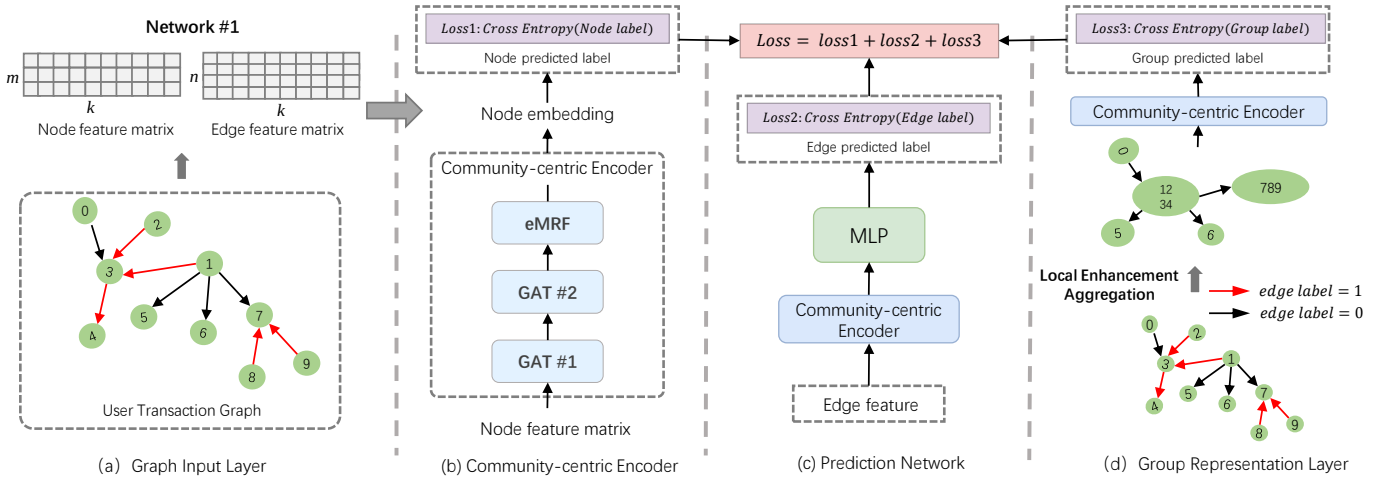

Fig. 3. The architecture of the proposed group-aware money-laundering detection model. (a) illustrates the input layer of the user transaction graph; (b) shows the architecture of the community-centric encoder; (c) displays the prediction network with joint optimization of nodes, edges, and groups representations; d) reports the group representation layer in local enhancement operations which takes the results learned from encoder and aggregate correlated nodes as gangs.

the graph can be roughly divided into two categories: suspicious groups and normal groups. Obviously, criminals and normal users are clearly separated into different groups, even though there are some ambiguous connections between different groups. As each individual transaction meets the compliance of regulation, conventional approaches face challenges in this situation because they fail to represent the group-level transaction behaviors.

Inspired by the above observations, a natural intuition is to employ graph analysis approaches in the ML detection process. However, simply employing graph learning methods, which infer features by its adjacent nodes, may lead to suboptimal results, because there are also many direct connections between criminals and normal users. Thus, if we could decent learning from organized criminal gangs, we may significantly improve the performance of AML approaches.

3 The Proposed Method

In this section, we introduce the architecture of the proposed approach first and then present the procedure of the graph feature learning layer. Finally, we introduce each component of the model and the loss function of money laundering detection.

3.1 Architecture Overview

Figure 3 shows the general architecture of GAGNN for money-laundering transaction detection. Generally, the model includes three parts: 1) Community-centric encoder. We transform the original transaction records into graphs and proceed to encode the node using both topological and attribute-wise information. In particular, we apply graph representation learning to generate node features out of edge features, and then leverage convolution layers to learn the embedding of the nodes, which are then fed into a fully connected layer to produce node classification results. 2) Group representation layer. We generate new edge representations and train them with a shallow neural network to classify transactions. As money-laundering usually appeals to organized behavior, we introduce the group aggregation strategy to merge nodes that are linked with transactions that are inferred as money laundering by the encoder into groups. As such, we derive a new group user transaction graph, which will then be fed into the community-centric encoder to infer the group representations. 3) Prediction network. The prediction network takes a joint optimization strategy, which is defined by combining the node classification loss, transaction classification loss, and group detection loss. In this paper, the user transaction graph is not fully connect and the type of prediction task is inductive. In the rest of this section, we will introduce each module in detail.

3.2 Community-centric Encoder

Given the input transaction behaviors, we construct the user transaction graph $G = (V, E)$ firstly. Users are denoted as nodes $V = \{v_1, v_2, \cdots, v_n\}$ and transactions as edges $E = \{e_1, e_2, \cdots, e_m\}$, where n is the number of nodes and m means the number of edges. The labels for all transactions are denoted as Y. Also, if node v_i is connected with a money-laundering transaction, we label node v_i as a negative sample, and otherwise a positive label. $A = (a_{ij})_{n*n}$ is the adjacent matrix of the graph, in which a_{ij} is 1 when there is an edge between node i and node j, and 0 otherwise. In feature engineering, we construct features for all transactions by concatenating a total of k features including basic transaction attributes, such as amount, times, etc. For better exploitation of the network structure, we consider feeding the network with node-wise information instead of edge-wise semantic information. We introduce deep graph representation learning here to derive the attribute matrix X for nodes and the attribute vector for node v_i is defined as:

$$v_i = \frac{1}{|M_i|} \sum_{j \in M_i} e_j, \tag{1}$$

$$\widehat{y}_i = \frac{1}{|M_i|} \sum_{j \in M_i} y_j, \tag{2}$$

where M_i denotes the indexes of the edges connected with node v_i, e_j denotes the feature vector for the j-th transaction, y_j denotes the ground-truth label the j-th transaction. In our implementation, the ground-truth label is reported by the bank card user and confirmed by financial domain experts of our collaborated financial institutions. $\widehat{y}_i$ denotes the suspicious money laundering label that is used for model training. In this way, the attribute vector for node v_i will be aggregated from its connected edges, and the label for node v_i denotes whether node v_i has been involved in previous ML transactions. Figure 4 shows the process of deriving node-wise representations from the edge-wise features.

In recent years, researchers have also been working on extensions of graph learning. Drawing on ideas from networks such as convolutional networks, recurrent networks, and deep autoencoders, researchers have realized that mechanisms such as convolution, recursion, and attention can also be introduced into graph neural networks. For example, convolutional networks have performed well in the field of computer vision, and inspired by convolutional networks, various graph convolution methods have emerged in recent years to introduce the concept of convolution into graph neural networks. Spectral-based graph convolution networks have been continually refined and improved, and as spectral methods typically process the entire graph at the same time and are difficult to scale to large graphs, spatial-based graph convolution networks have started to develop rapidly. The principle of these methods is basically the same, which is to perform convolution directly on the graph structure by aggregating information from the nearest neighbor nodes with ignoring the grouped (community) characters. Thus, inspired by Jin et al., [18] and Liu et al.,'s work [19], we leverage the graph attention layer (GAT) by extending the Markov random fields (eMRF) as the encoder of the graph network, which is devised for community-aware detection. In particular, we first build two layers of GAT, which can be mathematically represented as:

$$\begin{aligned} X^{(2)} &= A^{(1)}(A^{(0)}X^{(1)}H^{(0)})H^{(1)}, \\ A_{ij}^{(l)} &= \frac{\exp(e_{ij}^{(l)})}{\sum_{k\in\mathcal{N}(i)}\exp(e_{ik}^{(l)})}, \\ e_{ij}^{(l)} &= \text{LeakyReLU}(\alpha^{(l)}[z_i^{(l)}||z_j^{(l)}]) \end{aligned} \tag{3}$$

where $A^{(l)}$ denotes the attention matrix of the l-th GAT layer, $X^{(1)}$ denotes the input node attribute matrix, and $\alpha^{(l)}$ denotes the attention weight of the l-th GAT layer. $z_i^{(l)}$ denotes the output vector of the linear transformation of the l-th GAT layer. $H^{(0)}$ and $H^{(1)}$ are the parameters to be trained in this model. In this case, we transform the matrix A to incorporate the network with topological information as well as include the attribute matrix for the network to learn semantic information.

So far, the model has obtained a result matrix $X^{(2)}$ that is able to classify the nodes. The GAT can only obtain a relatively coarse classification result as it lacks the smoothness constraint to reinforce the group-aware neighboring nodes. We introduce the extended Malkov random fields (eMRF) with a graph attention layer. The essential of the Markov random field is the energy function, which consists of a unary potential function and a pairwise potential function. The Markov random field model simulates and simplifies the "one shot, all shot" nature of graph networks, based on which the unary potential function obtains the node classification probability from the graph convolutional network and measures the node classification result. The pairwise potential function is used to describe the relationship between nodes, encouraging similar nodes to be assigned the same label and nodes that differ more to be assigned different labels.

Particularly, the eMRF layer takes the coarse classification result from GAT as input, and considers nodes as different communities, encouraging the network to assign similar labels for nodes in the same community. Therefore, eMRF layer mines the inherent community relation between nodes and thus generating smooth classifications. In other words, in order to complete the calculation of the pairwise potential function, it is necessary to define the similarity of nodes, so that the higher similarity between

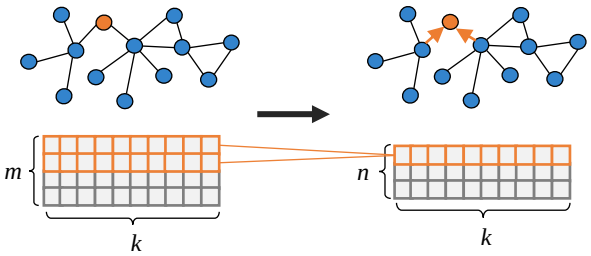

Fig. 4. The process of node-wise representation learning.

2 nodes is, the greater the forces will exert between them, resulting in closer labels being assigned to the 2 nodes. We modify the unary potentials and the pairwise potentials in MRF as GCN to measure both the topological and attribute similarity between nodes. Therefore, we propose the unary potentials and the pairwise potentials in eMRF to measure both the topological and attribute similarity between nodes. $\gamma(v_i, v_j)$ is defined to measure the attributes' similarity between node v_i and v_j as

$$\gamma(v_i, v_j) = \beta * \epsilon(v_i, v_j) + (1-\beta) * R_i(\zeta(v_i, v_j)), \tag{4}$$

where β is a tradeoff parameter that balances topology and attributes. $\epsilon(v_i, v_j) = \frac{d_i d_j}{2e} - a_{ij}$ (d_i is the degree of node v_i and e is the number of edges). $\zeta(v_i, v_j)$ denotes the cosine similarity between the attributes of node v_i and v_j, R_i will then apply the regularization across all pair of $\zeta(v_i, v_j)$. Then the proposed pairwise potentials can be defined as:

$$\Psi(v_i, v_j) = -1^{\sigma(v_i, v_j)}\gamma(v_i, v_j), \tag{5}$$

where $\sigma(v_i, v_j) = 1$ when v_i and v_j are labeled the same category, such as negative or positive simultaneously. $\sigma(v_i, v_j)$ measures the cosine similarity between the labels of nodes (v_i, v_j). The rationale is that when two nodes share the same label, the similarity between their attributes often appeals to be higher and otherwise lower. $-1^{\sigma(v_i, v_j)}$ is introduced to meet our expectation, as it turns $\gamma(v_i, v_j)$ negative and therefore makes the pairwise potential small when two nodes are labeled differently.

The unary potential for node v_i is defined as $\phi(v_i) = -p(v_i)$, where $-p(v_i)$ denotes the possibility of node v_i being labeled as positive given the results coming from GCN layers. Combining the unary potentials and pairwise potentials, the energy function for eMRF can then be represented as:

$$\text{E}(\text{C}|\text{A}, \text{X}) = \sum_{i=1}^{N}\phi(v_i) - \sum_{i\neq j}\Psi(v_i, v_j) \tag{6}$$

In order to transform the energy function and fit it into the GCN network, we employ mean-field approximation and it can be mathematically formulated as:

$$X^{(3)} = (X^{(2)} - \Gamma X^{(2)}H^{(2)}) \tag{7}$$

where $H^{(2)}$ are the parameters to be trained, and $\Gamma = (\gamma(v_i, v_j))_{n*n}$. Z, the output of community-centric, is the node embedding of the Graph G.

As presented above, the GCN encoder serves to incorporate topologically and attribute features. In our implementation, fully connected layers act as the classifier after the community-centric encoder generates the embedding of nodes. In practice, we adopt a shallow multi-layer perception (MLP) network, which can be represented as:

$$X^{(4)} = \text{sigmoid}(\text{NN}_\text{t}(X^{(3)}, W_t)) \tag{8}$$

where $\mathrm{NN_t}$ is a fully connected network with sigmoid activation and parameters W_t. $X^{(4)} = x_{ij\,n*2}$ denotes the binary classification result for all nodes, which represents the probabilities of a node involved in money laundering. Finally, we reach the node classification loss using cross-entropy based on the encoder classifying the node:

$$\mathcal{L}_{node} = \frac{1}{n}\sum_{i=1}^{n} \overline{y}_i \log(\overline{p}_i) + (1-\overline{y}_i)\log(1-\overline{p}_i) \tag{9}$$

where $\overline{y}_i$ denotes the actual label for node v_i and $\overline{p}_i$ denotes the predicted label for node v_i.

3.3 Group Representation Layer

As the main task of our work is to predict the money laundering transaction, we then proceed to construct edge representations. Suppose that node v_i and node v_j are connected by edge e_i, we concatenate the embedding of v_i, v_j and basic features of the edge to update the representation of edge e_i, where $e_i' = [Z_i, Z_j, l]$ and Z_i,Z_j refers to the embedding of node v_i,v_j; l refers to some basic features for an edge. Then, we fed the updated representation into an MLP prediction network for the classification task. The transaction loss is defined as:

$$\mathcal{L}_{trans} = \frac{1}{m}\sum_{i=1}^{m} y_i \log(p_i) + (1-y_i)\log(1-p_i) \tag{10}$$

where y_i denotes the ground-truth label of the i-th transaction and p_i denotes the predicted label of the i-th transaction.

In practice, as described above, money laundry often comes with organized activities, which highlights the importance of aggregating nodes and regards them working as a group. Hence, we introduce a policy to achieve node aggregation: if the MLP prediction network labels a transaction between node v_i and v_j as money laundry, we believe that node v_i and v_j may be conducting money laundry together and then aggregate them into a group.

In real practice, it's noted that money laundry often comes with organized activities, which highlights the importance to aggregate nodes and considering them to work as groups. Our previous observation found that the money laundering transactions were not sparsely distributed throughout the whole user transaction graph, but were concentrated between some individual accounts. In this case, it is possible to make a preliminary presumption about the gang nature of money laundering transactions. The nodes in the graph can be divided into 2 groups, namely ML gang members and normal accounts, although there may be some vague connections between the 2 groups. Traditional detection method faces a major challenge when analyzing this case, as they are unable to detect group-level transaction behavior from a higher dimensional perspective. Hence, we introduce a policy to achieve node aggregation: if the MLP prediction network labels a transaction between node v_i and v_j as money laundry, we believe that node v_i and v_j may be conducting money laundry together and then aggregate them into a group.

By implementing this across the whole graph, we reconstruct the original network to a new Graph $\hat{G} = (\hat{V}, \hat{E})$. We call them as node groups $\hat{V} = \{\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_{n'}\}$ in $\hat{G}$, which can either be an aggregated group node or a single node that doesn't belong to any groups. n' denotes the number of groups. We introduce $M_{n'n}$ to record the relationship between groups and nodes where M_i denotes all the indexes of the nodes belonging to the group $\hat{v}_i$. Afterward, we update the node feature matrix $\hat{X}$ for $\hat{G}$ by employing element-wise summation on nodes that belong to one group. Specifically, for group $\hat{v}_i$:

$$\hat{v}_i = \frac{1}{|M_i|}\sum_{j \in M_i} v^j, \tag{11}$$

where $\hat{X}$ contains the feature vectors for n' groups. Finally, we feed $\hat{X}$ and $\hat{G}$ into community-centric again, which can be mathematically represented as:

$$\begin{aligned}
\hat{X}^{(2)} &= (\hat{A}(\hat{A}\hat{X}^{(0)}H^{(0)})H^{(1)}) \\
\hat{X}^{(3)} &= (\hat{X}^{(2)} - \Gamma\hat{X}^{(2)}H^{(2)}) \\
\hat{X}^{(4)} &= \mathrm{sigmoid}(\mathrm{NN_t}(\hat{X}^{(3)}, W_t))
\end{aligned} \tag{12}$$

3.4 Prediction Network

In the downstream prediction task, we introduce the group-level loss $\mathcal{L}_{group}$ computed by learned embeddings of the group representation layer. In comparison to the node-level loss $\mathcal{L}_{node}$ and the transaction-level loss $\mathcal{L}_{trans}$, $\mathcal{L}_{group}$ here is designed to emphasize the detection of organized money-laundering behaviors in the transaction, which is defined as:

$$\mathcal{L}_{group} = \frac{1}{n'}\sum_{i=1}^{n'} \hat{y}_i \log(\hat{p}_i) + (1-\hat{y}_i)\log(1-\hat{p}_i) \tag{13}$$

where $\hat{y}_i$ denotes the ground-truth label for group $\hat{v}_i$ (if group v_i is an aggregated group, $\hat{y}_i = 1$ and otherwise $\hat{y}_i = 0$) and $\hat{p}_i$ denotes the predicted label for group $\hat{v}_i$. By minimizing the value of $\mathcal{L}_{group}$, the designed model could perform better in terms of detecting organized suspicious money laundering activities.

Finally, we design the loss $\mathcal{L}$ of the detection network by a joint combination of group representation loss $\mathcal{L}_{group}$, node classification loss $\mathcal{L}_{node}$ and transaction classification loss $\mathcal{L}_{trans}$, which is formulated as:

$$\mathcal{L} = \eta\mathcal{L}_{group} + \lambda\mathcal{L}_{node} + \zeta\mathcal{L}_{trans} \tag{14}$$

where η, λ and ζ denote the hyper-parameters to control the importance of the three losses. They are determined by cross-validation and we employ a joint optimization strategy to train the proposed method. By combining the $\mathcal{L}_{node}$, $\mathcal{L}_{trans}$ and $\mathcal{L}_{group}$, the detection network could learn a comprehensive capacity to address the organized money-laundering activities, in which node-level loss $\mathcal{L}_{node}$ represent the risk of whether the user (initiators or receivers of the transactions) involves money-laundering criminal. $\mathcal{L}_{trans}$ could guide the model to better infer transaction-level risk and $\mathcal{L}_{group}$ enables our proposed method the capacity to detect suspicious transactions with an organized-action perspective. Our collaborated domain experts confirm that all components are essential in real-world empirical investigation scenarios, which is also reported by existing studies [20], [21]. Our proposed method can be optimized through the standard stochastic gradient descent-based algorithms. In this paper, we used the Adam optimizer [22] to learn the parameters. We set the learning rate to 0.001 and batch size to 128 by default.

3.5 Complexity Analysis and Implementation

The optimization objective function involves all the nodes and edges in the graph, which is computationally inefficient for large-scale graphs. Therefore, we leverage node and neighbor sampling strategies to limit computation costs. Especially for high-degree

TABLE 1
The statistics information of the dataset.

Statistics Items	Week 1	Week 2	Week 3
Suspicious Trans.	101,524	100,293	100,657
Legitimate Trans.	1,753,355	1,691,267	1,636,082
Total Trans.	1.85M	1.79M	1.74M
Suspicious Nodes	9,536	8,985	9,052
Max. Degree	837	628	790
Avg. Degree	2.859	2.907	2.813

nodes, we sample the neighbor nodes according to the Bernoulli distribution with the same parameter. Specifically, we use a part of the graph to train our model in each training step. Sampling a series of nodes in $\mathcal{L}_{node}$ and $\mathcal{L}_{group}$ requires $O(|V|)$ time, where $|V|$ denotes the number of nodes. Afterward, sampling a series of neighbors requires $O(T|V|)$ time, where T is the maximum number in neighbors sampling and $T << |E|$. $|E|$ denotes the number of edges. For each step, eMRF requires N_b^2 times for correlation calculation, where N_b is the batch size. Therefore, the complexity of node-level optimization is $O(N_b^2 T|V|)$, which can be simplified to $O(|V|)$ as N_b and T are constant numbers. For the optimization of $\mathcal{L}_{trans}$, sampling on a sequence of edges requires $O(|E|)$ time. Finally, we reach the overall time complexity of GAGNN as $O(|V| + |E|)$, linear to the number of nodes and edges, demonstrating that our proposed method is computing efficient for large-scale graphs.

In the implementation, the proposed method is trained offline with historical records regularly. For example, the model is trained every night in the industry scenario, which requires 1.5 hours to complete the learning phase on 60 million samples with four pieces of Telsa V100 GPU. The trained model is then leveraged for online prediction on the next day. The corresponding user transaction graph and features are stored in an in-memory database. In the prediction phase, when a new transaction is issued, the model could effectively retrieve and update the adjacent graph component and features from the in-memory database. Then, the retrieved adjacent graph structures and features are employed for online prediction by the offline-trained model. Finally, the new transactions are used for model training each day so that the model can be up-to-date by learning from new data. As we only leverage the transaction and user attributes as features, the retrieve and update in-memory database process could be very efficient.

4 Experiments

In this section, we conduct extensive experiments for evaluating the effectiveness of our proposed methods. We first report statistical information of our dataset and implementation details in experimental settings. Then, we present the experimental results of GAGNN compared with other baselines. Finally, we report the result and observations of case studies in the last subsection.

4.1 Experimental Settings

4.1.1 Datasets

The experiment dataset includes user transactions from 06/09/2021 to 26/09/2021, which is collected from UnionPay. The suspicious money laundering transaction is labeled by financial risk experts with the help of auxiliary process automation tools. We divide the data into three weeks with Week 1 (06/09/2021-12/09/2021), Week 2 (13/09/2021-19/09/2021), and Week 3 (20/09/2021-26/09/2021). There are about 101 thousand labeled suspicious records, accounting for only approximately 5% of the collected 1.8 million transactions each week, which is still a small part of nearly 40 million entire records. In this paper,

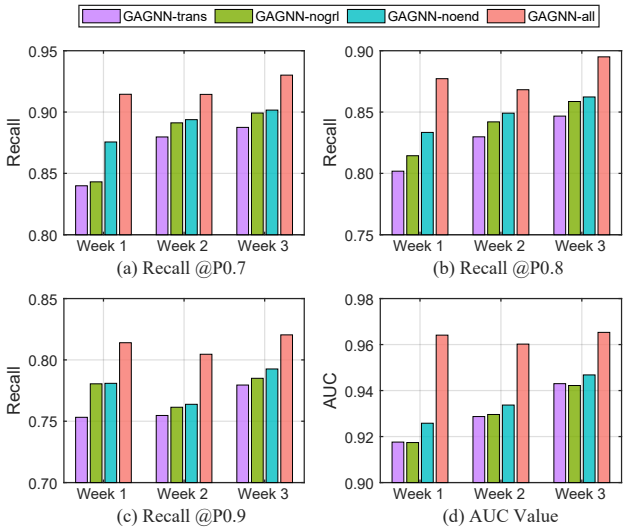

Fig. 5. The ablation study results of our proposed method. We remove each component in turn (GAGNN-trans/nogrl/noend) and compare them with the full version of the model (GAGNN-all).

we use the down-sampling method on legitimate transactions to deal with this highly imbalanced problem. In particular, we select all users who have ever been involved in money laundering and all their corresponding historical transactions. Then, we down-sample from normal users who have not experienced any money laundering activities and then extract all the transaction records of sampled users. We employ the 70% of the data for training and the rest part as a test set week by week. Table 1 reports the detailed statistics of the dataset.To the best of our knowledge, we did not find other real-world bank card transaction datasets with human-annotated money laundering labels. This experiment is conducted on one dataset, and even though it is real-world and large-scale, the results may be limited in the current data distribution and patterns.

4.1.2 Compared Baselines

We employ the following widely used approaches in the banking industry as baselines to highlight the effectiveness of our proposed methods:

- *LR:* Logistic regression (LR) model [23] is widely used method in financial industry. We apply L2 normalization and set $\lambda = 1$, tolerance for stopping criteria to 1e-4 and max iteration to 1000 and Follow the-regularized-leader (FTRL) for optimization.
- *GBDT:* Gradient boosting decision tree [24] is a popular ensemble learning method for classification, which has proved effective in suspicious transaction detection. We set the max depth to 3, The number of boosting stages to 500.
- *SVM:* Support vector machine [24] has become an important classification methods, which has proved effective in fraud transaction detection. We set the learning rate to 0.01.
- *Fuzzy Rule:* [25] A fuzzy rule method that captures human domain knowledge and model non-linear mapping of input-output space. We set the rules according to the recommendation by our collaborated domain experts.
- *GraphSAGE:* [26] A framework for inductive representation learning on large graphs. GraphSAGE is used to generate low-dimensional vector representations for nodes, and is especially useful for graphs that have rich node attribute information.
- *GCN:* [27] A graph convolutional network for supervised learning (node and edge classification) on graph structure.

TABLE 2
The experimental results of our proposed GAGNN in suspicious money laundering transaction detection, compared with the wide-used benchmark methods in the banking industry. We report the results of AUC and recall at different precision levels.

Methods	Week 1					Week 2					Week 3				
	R@P$_{0.6}$	R@P$_{0.7}$	R@P$_{0.8}$	R@P$_{0.9}$	AUC	R@P$_{0.6}$	R@P$_{0.7}$	R@P$_{0.8}$	R@P$_{0.9}$	AUC	R@P$_{0.6}$	R@P$_{0.7}$	R@P$_{0.8}$	R@P$_{0.9}$	AUC
LR	0.7736	0.7542	0.6938	0.6246	0.8356	0.8059	0.7771	0.6759	0.6173	0.8164	0.7851	0.7542	0.7057	0.6340	0.8396
SVM	0.7790	0.7563	0.7293	0.6553	0.8440	0.8333	0.8018	0.7339	0.6408	0.8504	0.8394	0.7981	0.7304	0.6788	0.8613
GBDT	0.7814	0.7625	0.7416	0.6785	0.8878	0.8362	0.8071	0.7483	0.6872	0.8696	0.8427	0.8093	0.7777	0.6913	0.8725
Fuzzy Rule	0.7791	0.7554	0.7279	0.6557	0.8430	0.8152	0.7541	0.6958	0.6443	0.8240	0.7901	0.7591	0.7208	0.6594	0.8483
GraphSAGE	0.7813	0.7621	0.7312	0.6923	0.8942	0.8371	0.8051	0.7478	0.6854	0.8672	0.8432	0.8102	0.7639	0.6909	0.8832
GCN	0.7952	0.7829	0.7453	0.7001	0.8993	0.8663	0.8209	0.7561	0.6902	0.8734	0.8616	0.8329	0.7716	0.7025	0.8902
GAT	0.8472	0.8341	0.7875	0.7148	0.9149	0.8982	0.8441	0.7687	0.6945	0.8802	0.8854	0.8509	0.7752	0.7103	0.9011
Graphormer	0.8441	0.8319	0.7841	0.7113	0.9088	0.8927	0.8432	0.7626	0.6933	0.8791	0.8843	0.8512	0.7749	0.7100	0.9003
Graphconsis	0.8491	0.8381	0.7892	0.7285	0.9181	0.8949	0.8551	0.7742	0.7152	0.8936	0.8919	0.8583	0.7814	0.7284	0.9142
Care-GNN	0.8519	0.8393	0.7914	0.7396	0.9199	0.9001	0.8632	0.7795	0.7193	0.8987	0.9068	0.8658	0.7969	0.7350	0.9185
PC-GNN	0.8587	0.8403	0.7987	0.7439	0.9242	0.9074	0.8712	0.7874	0.7261	0.9103	0.9195	0.8721	0.8159	0.7571	0.9266
GAGNN	**0.9474**	**0.9145**	**0.8772**	**0.8140**	**0.9641**	**0.9381**	**0.9144**	**0.8682**	**0.8046**	**0.9602**	**0.9650**	**0.9301**	**0.8950**	**0.8204**	**0.9653**

- *GAT:* [28] A well-known graph neural network-based model with attention mechanism for graph learning. We set the attention head k to 5, the batch size to 128.
- *Graphormer:* [29] A transformer-based attention mechanism for large-scale graph learning.
- *Graphconsis:* [30] A graph-based financial fraudsters detection model, which learns the relation attention weights associated with the sampled (filter the inconsistent neighbors) nodes. We set the learning rate to 0.001 and batch size to 128.
- *Care-GNN:* [31] A camouflaged and grouped behavior detection model by enhancing the GNN aggregation process with unique modules against camouflages. We set the parameters as the original paper recommended.
- *PC-GNN:* [32] A Pick and Choose Graph Neural Network for node-level supervised learning on graphs. PC-GNN picked nodes and edges with a devised label-balanced sampler to construct sub-graphs for mini-batch training.
- *GAGNN-trans/noend/nogrl:* Our model has several variations: In GAGNN-noend, the community-centric encoder is not employed. We directly fed transaction attributes into the MLP prediction network. In GAGNN-nogrl, the group representation layer is not employed. In GAGNN-trans, we only employ the transactions for feature learning.
- *GAGNN-all* denoted the full model proposed in this paper. We use the GCN and eMRF to encode the nodes and concatenate edge features together to construct edge features, and train them by the joint optimization.

4.1.3 Evaluation Metrics and Parameter Settings

We evaluate the performance of the proposed approach by AUC and R@P$_N$. The first metric AUC is defined as the area under the ROC curve. Compared with the trade-off of precision and recall, we can more directly distinguish which method performs better with AUC. The second metric R@P$_N$ indicates the recall rate when the precision rate equals N. As the results of suspicious money laundering transaction detection are critical for financial institutions, a high precision rate is generally required. In the experiment, we set N according to industry demands in order to measure the ability of detected top-ranked suspicious transactions. The higher score of both the AUC and R@P$_N$ indicate the higher performance of the methods.

In this experiment, we prefer to use the originally proposed parameters for each baseline method. As to the implementation of *GAGNN-all*, the number of hidden units of GAT#1 is set at 64 so that the model will be scalable facing large-scale datasets. In eMRF layer, β is introduced and set at 0.44 to balance topological similarity and attribute-wise similarity. The unit of $\mathrm{NN_t}$ is set at 128 to extend the original node feature into higher dimensional features. The experiments are conducted on a server with four 3.6GHz Intel Cores and 24GB RTX 3090 GPU.

TABLE 3
The AUC value of employing different graph learning methods as the community-centric encoder.

Methods	Week 1	Week 2	Week 3
GAT	0.9306	0.9349	0.9483
GraphSAGE+eMRF	0.9448	0.9413	0.9439
GCN+eMRF	0.9603	0.9571	0.9634
Graphormer+eMRF	0.9615	0.9592	0.9639
GAT+eMRF (GAGNN)	**0.9641**	**0.9602**	**0.9653**

4.2 Suspicious Money Laundering Detection

In this section, we evaluate the accuracy of suspicious transaction detection of money laundering, which is one of the main tasks of this paper. The money laundering label is annotated at the transaction level instead of the user level. Consequently, this is an edge classification problem here with group-aware demands. We compare our method with the widely-used approaches in the banking industry (LR, SVM, GGDT, Fuzzy Rule), state-of-the-art graph learning baselines (GraphSAGE, GCN, GAT, Graph Transformer, Graphconsis, Care-GNN and PC-GNN). We report the AUC score and the $R@P_N$ (recall at different precision level) value with different numbers of N, from 0.6 to 0.9.

Table 2 shows the accuracy report of eleven baseline methods. As we can see, the shallow methods (LR, SVM, and Fuzzy Rule) cannot achieve satisfactory results, as reported in lines 1, 2, and 4. GBDT performs better than shallow approaches by ensemble learning on basic classifiers, with an average of 2-4% improvements. With the help of deep graph representation and attention mechanism, graph-based methods boost the prediction accuracy by over 6% improvements, which strongly demonstrates the effectiveness of leveraging the graph feature in money laundering detection. The last three baselines are better that GAT and PC-GNN perform better than Care-GNN and Graphcosis, which shows that sub-graph structure is essential for suspicious money laundering detection. According to the precision increase from 0.6 to 0.9, the recall value gradually decreased. The last line reports the result of our proposed method. GANN is considerably superior to all baselines and the improvements are more significant compared to well-known industry benchmarks, from averaged AUC from 85% to 96%. The R@P$_N$ is also greatly improved with a recall score of 0.8, boosting from 72% to 86%, and a recall score of 0.9, boosting from 65% to 80%. Higher accuracy

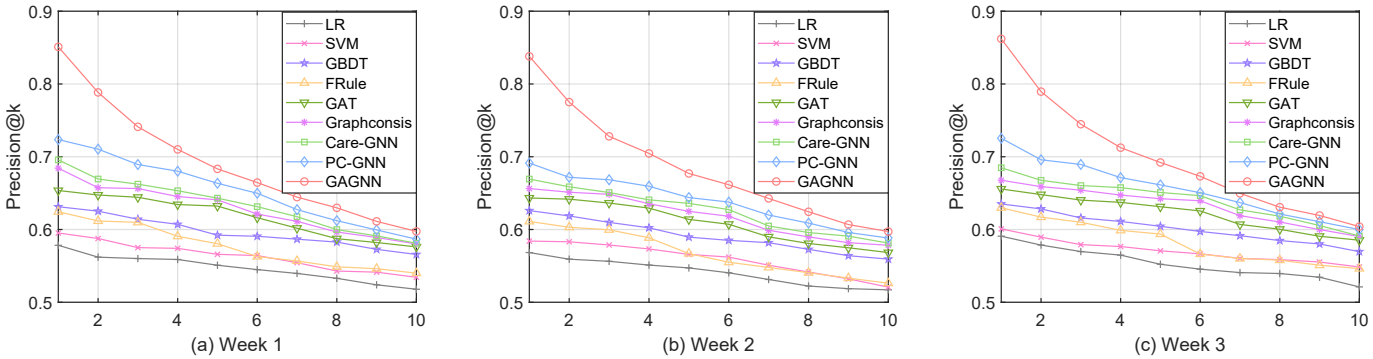

Fig. 6. The precision@k of each test window in the organized money laundering detection experiment. The x-axis denotes the top k% of predicted ML suspicious transactions and y-axis means the precision of the ML groups.

of R@P$_N$ shows that our proposed GAGNN could detect more suspicious money laundering transactions and keep a high-level precision simultaneously. This capability is crucial for real-world applications, proving the effectiveness of our proposed method in money laundering detection.

4.3 Ablation Study

Figure 5 reports the result of three variations of our proposed model, GANN-trans, GANN-nogrl and GANN-noend, compared with the full version of GANN. It is clear that these variations are not performed well compared with GAGNN-all, proving the effectiveness of our proposed community-centric encoder and group representation layer. Please note that GAGNN-noend performs better than GAGNN-nogrl, which means the group representation layer contributes more importantly to suspicious transaction detection, which demonstrates the effectiveness of the main contribution of our proposed method.

In addition, we also evaluate the effectiveness of this paper's proposed community-centric encoder by replacing the graph feature learning layer with GCN, GraphSAGE and Graph transformer. Table 3 reports the AUC value of each method in three weeks test window. As we can see, employing GAT alone cannot achieve satisfactory performance, demonstrating the essence of the proposed eMRF module. GAT+eMRF achieves the best performance compared with the rest baselines. The reason might be that a higher-capacity model, like the graph transformer, also leads to a higher risk of overfitting. The GAT is the best trade-off between the model's capacity and generality in this task.

4.4 Organized Money Laundering Detection

As described above, another major task of this work is the ability of the model in detecting organized money laundering activities. In this experiment, we evaluate the performance of group money laundering behaviors; in other words, unlike the previous transaction-level detection tasks, organized behaviors are aggregated as groups in this test. We utilize the precision of predicted top k percentage of confident suspicious transactions for evaluating the organized money laundering detection experiment.

In particular, we select the top 1% to 10% of most confident money laundering transactions by our proposed method and compared baselines. Then, we aggregate the connected money laundering suspicious transactions into groups until there are no more ML nodes or edges connected to the group. Afterward, we compute the predicted output with the actual label in groups and marked them as true if more than 50% of nodes are involved in the money laundering transaction. The threshold value of 50% was determined by domain experts with the help of cross-validation. Finally, we could compute the precision of grouped money laundering detection by each method in top k% of confidence.

TABLE 4
The accuracy of money laundering detection methods in different groups, where Bi denotes the bridging nodes and Bd means bridged nodes.

Methods	Bi4	Bd4	Bi5	Bd5
LR	0.7090	0.5000	0.6185	0.7500
SVM	0.7636	1.0000	0.6597	0.8125
GBDT	0.8000	1.0000	0.7422	0.8750
Fuzzy Rule	0.7818	1.0000	0.7113	0.8125
GAT	0.8363	1.0000	0.8041	0.8823
Graphconsis	0.8545	1.0000	0.8267	0.9375
Care-GNN	0.8545	1.0000	0.8333	0.9375
PC-GNN	0.8727	1.0000	0.8400	0.9375
GAGNN	0.9090	1.0000	0.8659	0.9375

Figure 6 shows the results of the organized money laundering experiments on various baselines. The x-axis denotes the number of top k-th confident results and the y-axis denotes the precision. As can be seen from the chart, the top results keep sufficient precision. The top 1% confident results receive a precision of over 55% for all the methods. In general, the precision of baseline methods gradually decreases against the increase of k, that is because the more samples predicted, the less confidence in the model. From the three subgraphs of Figure 6, we observe that the precision of logistic regression, SVM, and Fuzzy Rules are not comparable to other methods. It is probably because the conventional models do not include the graph features and in this experiment, group representations are vitally important. Graph attention networks (GAT) and GBDT perform better than the rest four classical compared benchmark methods. The deep graph learning and ensemble learning approaches are proved to be effective, especially for deep graph learning, such as PC-GNN, Care-GNN, and Graphcosis, which achieve the best performance in these baselines. GAGNN performs significantly better than GAT across all time periods and all the top k % confidence. The improvements vary from 3% to 20%, which are more remarkable in the top 1 and 2%. This phenomenon proves the superior performance of our methods in organized money laundering detection, with only the top 1% of predictions, achieved averaging 85% precision, with is significantly better than averaged 65% of precision by compared benchmarks. The reason might be that with a group representation layer and deep graph encoders, GAGNN could effectively learn from the structures in organized money laundering graphs. The essential and effectiveness of group representation layer and deep graph encoders are demonstrated in addressing the organized money laundering detection problem.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2023.3272396

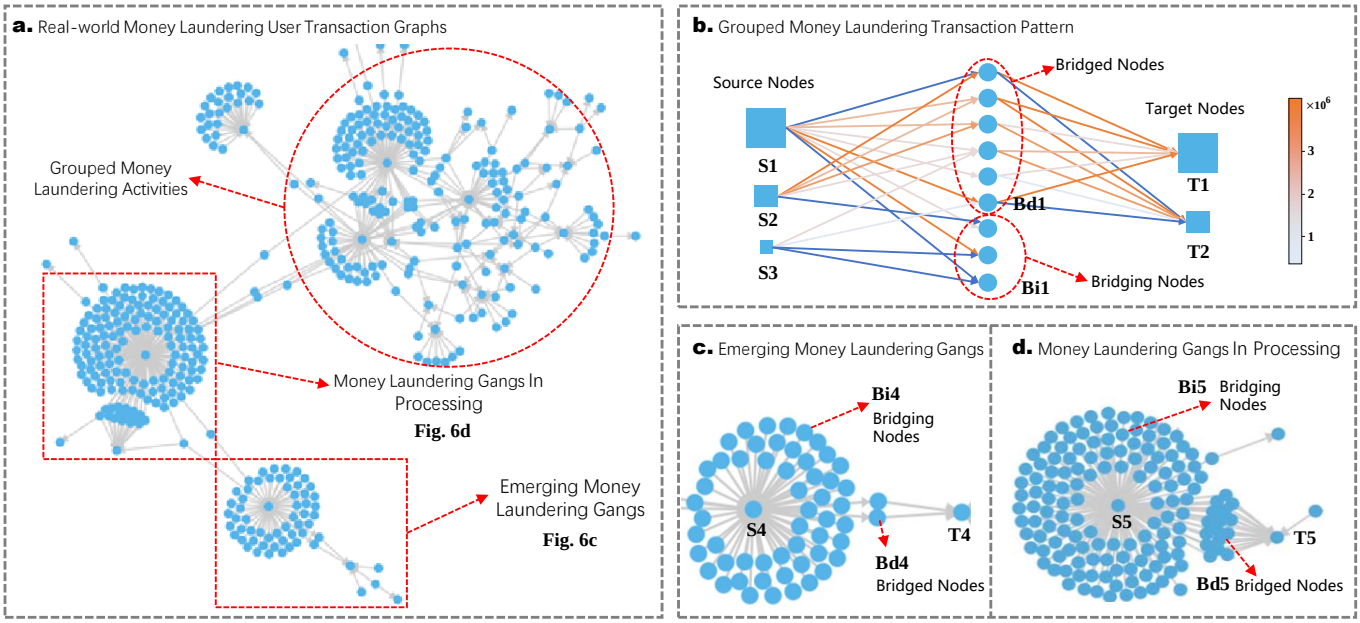

Fig. 7. Visualization of user transaction graphs in real-world anti-money laundering system. a) shows the graph marked as grouped ML activities, in processing ML gangs and emerging ML groups. b) display the grouped money laundering transaction pattern where the node size and edge color denotes the amount of money transferred. c) reports the emerging money laundering groups and d) displays the money laundering gangs in processing.

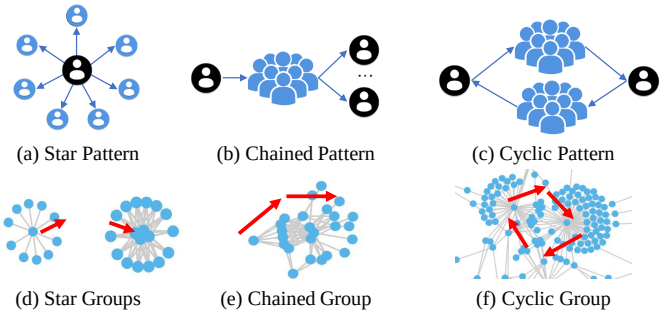

Fig. 8. The visualization of different money laundering patterns and the corresponding organized groups detected in real-world situation.

4.5 Case Studies

In this section, we report the case studies applying our methods to an industry-level anti-money laundering system in Unionpay. We select a typical transaction network and report the statistical information of the predicted high-risk suspicious transactions and visualize them in the empirical study. Figure 7a illustrates the real-world money laundering user transaction graph, in which there are three types of ML groups. 1) the grouped activities are located in the top right of the figure, which contains complex graph structures and preserves group-level activities obviously. We mark them in the red circle. 2) denotes the money laundering gangs in processing and are detail reported in Figure 7d. 3) emerging ML gangs are marked at the bottom the Figure 7a and amplified in Figure 7c. These types of gangs are three typical stages of money laundering: emerging, processing, and grouped ML activities. The main purpose of the anti-ML model is to detect these gangs in the early stage. Meanwhile, we discover the most common ML pattern in case studies and report them in Figure 7b. Normally, the ML activities include source nodes and target nodes, who aim to transfer money from source to target without being detected by the anti-ML system. Therefore, there are a lot of bridge nodes between sources and targets, as shown in Figure 7d, there are over 100 bridge nodes involved in the ML activities. We define bridged nodes, denoted as Bd, who have completed to transfer of the money from sources to targets, and the bridging nodes, Bi, denotes that received money from sources but have not transferred it to the target nodes. As we can see, the gangs illustrated in Figure 7c and Figure 7d both follow this pattern, proving the effectiveness of our proposed method in organized money laundering activities detection. The pattern is also observed by pattern mining research in the financial literature [9].

Then, we investigate the detection accuracy of bridging nodes, which is critically important for anti-money laundering combat, as well as the bridged node. Because we could take proper regulation activities in advance once detect the bridging nodes in the emerging ML groups so that to cut the laundering procedure and prevent criminal behavior consequently. As we can see, Figure 7c shows a typical emerging ML group with only two bridged nodes (Bd4) and over 50 bridging nodes (Bi4). Table 4 reports the accuracy of detected ML suspicious activities of Bi4 and Bd4. Most methods successfully detect 2 Bd4 nodes with an accuracy of 100%. However, the accuracy bridging nodes Bi4 is much less than Bd4, with only averaged 78% accuracy of baselines. Our methods show significant advantages in this situation with the precision of 90.9% in bridging nodes. According to the ML stage, this advantage is more prominent with the precision of Bi5 improving near 25% and Bd5 improving about 18% compared with logistic regression. Our methods achieve the best performance in all groups and the improvements is more significant in the bridging groups, denoted as Bi4 and Bi5. At the same time, our proposed GAGNN achieves at least as well as the most competitive benchmarks in bridged groups, with the same accuracy of 93.75% in the Bd5 group with graph attention network. The superior performance and the ability of pattern discovery of our proposed method in the case study demonstrate its effectiveness in organized money laundering activities. This means that the proposed approach will be invaluable in offering constructive evidence and strong hint information to financial regulators.

To further investigate organized money laundering patterns, we visualize three typical types of suspicious structures in Figure 8. The first pattern, named “Star Pattern”, is shown in Figure 8a, which normally occurred during placement (shown in the left

part of Figure 8d, diffusing the money from the source node) and integration phase (the right part of Figure 8d, aggregating money to target nodes). Figure 8b and 8e present the "Chained Pattern" and its groups that transmit money from the source node to the targets by various bridge nodes, forming numerous chains. The chain pattern is more likely observed in the laying and laundering phase. Figure 8c and 8f show the "Cyclic Pattern", which is more complicated than "Star" and "Chain" patterns. In a cyclic group, the source node transmits money to the target while the target also transfers back to the source through various bridge nodes. In this process, the dirty money could be laundered by multiple steps among masses of cyclic transactions. After empirical analysis with the domain experts in the collaborated financial institution, we also observed that "Chained Pattern" could consist of multiple "Star Pattern". At the same time, the "Cyclic Pattern" is the combination of two or more "Chained Pattern". Please note that the arbitrary combinations of these patterns, which are also reported in previous studies [33], [34], may lead to very intricate groups as shown in Figure 8d, 8e and 8f. As a result, it is more urgent than ever to develop a more powerful and flexible approach to fighting against these organized criminals.

5 Related Works

This section presents a review of recent literature on money laundering detection and graph learning in financial networks.

5.1 Money Laundering Detection

Money laundering is the act of disguising, concealing, and transforming the illegal income obtained from a crime or other illegal and illicit act to make it formally legal through various means [35]. Therefore, Anti-Money Laundering (AML) detection has always been of great significance to stabilize the financial market [11]. Rule-based approaches are the most frequent and classical ones in the financial industry. For example, Panigrahi et. al, [36] introduced an intrusion to the database detection method using different components, including a rule-based component and Bayesian component. Rajput et. al, [37] proposed an ontology-based expert-system that contain domain knowledge and a specific set of rules to detect suspicious transaction. However, these rule-based approaches rely heavily on expert knowledge, and hence are easily circumvented and do not work well to detect new types of money laundering crimes.

Machine learning algorithms were recently applied in AML, which push the boundaries of traditional rule-based classifications [11]. These methods can be classified into supervised learning such as Decision Tree, SVM, and Random Forest; and unsupervised learning such as clustering [38]. Clustering is often used to group transactions into different clusters to detect patterns of suspicious transaction sequences. Wang et. al, [39], for instance, implemented Clustering With Slope (CLOPE) to group financial data into transaction groups. However, clustering approaches neglect the flows of transactions and relationships between accounts. Other supervised machine learning algorithms have also been used in AML tasks. Guevara et. al [40] compared several ML algorithms and proposed a probabilistic graphical modeling technique (PGM) as a Bayesian network for unsupervised data to detect anomalies in Non-banking transactions. Savage [41] implemented supervised learning using Random Forest and Supper vector machine for AML tasks.

Different learning approaches have their own advantages and disadvantages. The selection of models requires a trade-off between accuracy and interpretation [38], [42]. Models such as neural networks and gradient boosting models are known as black-box models. These black-box models often provide highly accurate results but lack relatively in interpret ability, so such deep learning-based data mining models are difficult to explain the rationale behind their model design to financial institutions, while the cost of training is often higher than rule learning [11], [43]. On the other hand, white-box models such as decision trees and linear regression [44], [45] have strong interpretability, and cost less to explain the model, but the accuracy of their results is generally poor and their performance cannot meet financial standards. However, Given that money launderers are developing newer methods and often commit crimes as organized activities, The real performance of AML algorithms depends on their adaptability and generalization capabilities. Also, current supervised ML data mining techniques pay attention to identifying individual anomalous transactions and thus will be less effective in the context of detecting new patterns of money laundering activities.

5.2 Graph Learning on Financial Networks

Graph learning algorithms have been studied broadly in the financial literature [46], [47], [48], [49], [50], [51], [52], such as fraud detection [19], [53], [54], loan default prediction [55], [56], anomaly detection [57], and blockchain analysis [58]. For example Cheng et al., [59] proposed a spatial-temporal attention-based graph network (STAGN) for credit card fraud detection which learns the temporal and location-based graph features by a graph neural network. Wang et al., [60] proposed a semi-supervised attentive graph neural network, which utilized the labeled and unlabeled data at the same time for fraud detection. With the development of graph neural network, graph-based anomaly detection methods [61], [62], [63], [64] achieved remarkable progress, which could also be leveraged to suspicious transaction detection. For example, Wu et al., [65] introduced a graph learning-based method for anomaly detection in the Industrial Internet of Things (IIoT) applications. Goodge et al., [64] addressed the local outlier detection by unifying a GNN-based message-passing framework. But these works mainly focus on the individual abnormal node or edge detection instead of grouped outliers. The existing models on group-aware anomaly detection have been studied on tabular and/or sequential data [66], [67], [68] using various techniques, such as hierarchical Bayes model [69], [70], kernel-based model [71], deep generative models [67] and autoencoders [72]. The most similar of our work in the anomaly detection field is the abnormal pattern [73], [74], community [75], [76] and subgraph detection [77], [78], [79] on graphs. But to our best knowledge, these works cannot perform well in the grouped money laundering detection task because: 1) the annotated labels and label-supervised process are essential for model training. Thus, we formulate the money laundering detection problem as a supervised task. The anomaly detection method could not adequately use the labeled information and may consequently lead to sub-optimal performance; 2) suspicious behaviors are increasingly hidden due to their adversary nature. Using generic anomaly detection approaches can hardly handle this issue as these patterns are hidden behind labeled data.

The issue of money laundering exists and continues to threaten the stability of the financial market for the past decades, facing the challenges of identifying suspicious money laundering

transactions - huge and organized behaviors, changing money laundering patterns to complex graph structures [80]. Graph neural network allows for a straightforward way to represent financial behaviors and describe relationships between fraudsters and illicit transactions [81], [82], [83]. Researchers introduced the high-performance graph analysis method recently to retrieve suspicious transactions [9], [84]. But the method cannot infer the graph feature in model training. Recently, graph learning-based suspicious transaction detection methods [85], [86], [87] have been leveraged for money laundering detection. For example, Weber et al., [17] employs graph convolutional neural networks for forensic analysis of financial data and infers that graph deep learning for AML bears great promise in the fight against criminal financial activities. Alarab et al., [88] proposed a graph-based long and short memory (LSTM) model for anti-money laundering in blockchain network. In the cryptocurrency literature, graph techniques are widely utilized for money laundering detection [89], [90], [91] and show the superiority in learning from transaction-level graphs [92]. However, little research has been done to incorporate organized money laundering detection with group-aware graph neural networks in a supervised learning paradigm.

6 Conclusion

In this paper, we propose the group-aware graph neural network-based approach (named GAGNN) to detect organized money laundering activities, a capital felony in the financial industry. By extensive study in Unionpay, hosting national-wide card transaction records, we observe the grouped (gang) behavior in money laundering criminals and we devise GAGNN which could learn from user transaction graph directly by a community-centric encoder and deep group representation layer. We thoroughly evaluate the proposed method in a real-world dataset compared with widely-used benchmarks and achieve the best performance. We also conduct empirical studies in the industry-level systems and the superior performance of our method is strongly demonstrated in detecting organized money laundering activities and emerging ML groups. The ability of the proposed method to address the challenges of organized ML criminals is proved in the experiment. The idea of modeling suspicious transactions based on group-aware deep graph learning can be applied widely in organized behavior detection. In future work, we plan to explore semi-supervised graph learning methods to learn from large-scale unlabeled data, which we believe could enrich our capability in addressing group-aware money laundering detection tasks.

References

[1] D. Hopton, *Money laundering: A concise guide for all business*. Routledge, 2020.

[2] M. Levi, "Money laundering and its regulation," *The Annals of the American Academy of Political and Social Science*, vol. 582, no. 1, pp. 181–194, 2002.

[3] U. N. O. on Drugs, Crime, V. I. Ctr, and Austria, "Estimating illicit financial flows resulting from drug trafficking and other transnational organized crimes," 2011.

[4] M. Jullum, A. Løland, R. B. Huseby, G. Ånonsen, and J. Lorentzen, "Detecting money laundering transactions with machine learning," *Journal of Money Laundering Control*, 2020.

[5] M. S. D. Council, "Money laundering policy," *Policy*, vol. 3, p. 8, 1910.

[6] L. Wu, J. Tang, Y. Xia, J. Pei, and X. Guo, "The sixth international workshop on deep learning on graphs-methods and applications (dlg-kdd'21)," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 4167–4168.

[7] D. Masciandaro, "Money laundering: the economics of regulation," *European Journal of Law and Economics*, vol. 7, no. 3, pp. 225–240, 1999.

[8] A. Salehi, M. Ghazanfari, and M. Fathian, "Data mining techniques for anti money laundering," *International Journal of Applied Engineering Research*, vol. 12, no. 20, pp. 10 084–10 094, 2017.

[9] X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng, "Flowscope: Spotting money laundering based on graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4731–4738.

[10] S. Gao and D. Xu, "Conceptual modeling and development of an intelligent agent-assisted decision support system for anti-money laundering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1493–1504, 2009.

[11] Z. Chen, L. D. Van Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam, "Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review," *Knowledge and Information Systems*, vol. 57, no. 2, pp. 245–285, 2018.

[12] A. I. Canhoto, "Leveraging machine learning in the global fight against money laundering and terrorism financing: An affordances perspective," *Journal of business research*, vol. 131, pp. 441–452, 2021.

[13] P. Xia, Z. Ni, H. Xiao, X. Zhu, and P. Peng, "A novel spatiotemporal prediction approach based on graph convolution neural networks and long short-term memory for money laundering fraud," *Arabian Journal for Science and Engineering*, pp. 1–17, 2021.

[14] N. Aggarwal, S. Wareham, and R. Lehmann, "Applications of machine learning in the identification, measurement and mitigation of money laundering," *Journal of Financial Compliance*, vol. 4, no. 2, pp. 140–166, 2020.

[15] G.-Y. Sheu and C.-Y. Li, "On the potential of a graph attention network in money laundering detection," *Journal of Money Laundering Control*, 2021.

[16] A. M. Mubalaike and E. Adali, "Deep learning approach for intelligent financial fraud detection system," in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2018, pp. 598–603.

[17] M. Weber, J. Chen, T. Suzumura, A. Pareja, T. Ma, H. Kanezashi, T. Kaler, C. E. Leiserson, and T. B. Schardl, "Scalable graph learning for anti-money laundering: A first look," *arXiv preprint arXiv:1812.00076*, 2018.

[18] D. He, Y. Song, D. Jin, Z. Feng, B. Zhang, Z. Yu, and W. Zhang, "Community-centric graph convolutional network for unsupervised community detection," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 3515–3521.

[19] C. Liu, L. Sun, X. Ao, J. Feng, Q. He, and H. Yang, "Intention-aware heterogeneous graph attention networks for fraud transactions detection," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 3280–3288.

[20] M. E. Beare, *Critical reflections on transnational organized crime, money laundering and corruption*. University of Toronto Press, 2003.

[21] M. Levi and M. Soudijn, "Understanding the laundering of organized crime money," *Crime and Justice*, vol. 49, no. 1, pp. 579–631, 2020.

[22] D. P. Kingma and J. L. Ba, "Adam: Amethod for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2014.

[23] H. B. McMahan, "Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization," 2011.

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *NeurIPS*, 2017, pp. 3146–3154.

[25] Y.-T. Chen and J. Mathe, "Fuzzy computing applications for anti-money laundering and distributed storage system load monitoring," 2011.

[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[27] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks."

[29] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 877–28 888, 2021.

[30] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1569–1572.

[31] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 315–324.

[32] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: a gnn-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021*, 2021, pp. 3168–3177.

[33] P. Weibing, "Research on money laundering crime under electronic payment background," *Journal of Computers*, vol. 6, no. 1, pp. 147–154, 2011.

[34] K. Singh and P. Best, "Anti-money laundering: Using data visualization to identify suspicious activity," *International Journal of Accounting Information Systems*, vol. 34, p. 100418, 2019.

[35] M. Levi and P. Reuter, "Money laundering," *Crime and justice*, vol. 34, no. 1, pp. 289–375, 2006.

[36] S. Panigrahi, S. Sural, and A. K. Majumdar, "Detection of intrusive activity in databases by combining multiple evidences and belief update," in *2009 IEEE Symposium on Computational Intelligence in Cyber Security*. IEEE, 2009, pp. 83–90.

[37] Q. Rajput, N. S. Khan, A. Larik, and S. Haider, "Ontology based expert-system for suspicious transactions detection," *Computer and Information Science*, vol. 7, no. 1, p. 103, 2014.

[38] S. Kumar, L. Akoglu, N. Chawla, J. A. Rodriguez-Serrano, T. Faruquie, and S. Nagrecha, "Machine learning in finance," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 4139–4140.

[39] X. Wang and G. Dong, "Research on money laundering detection based on improved minimum spanning tree clustering and its application," in *2009 Second international symposium on knowledge acquisition and modeling*, vol. 2. IEEE, 2009, pp. 62–64.

[40] J. Guevara, O. Garcia-Bedoya, and O. Granados, "Machine learning methodologies against money laundering in non-banking correspondents," in *International Conference on Applied Informatics*. Springer, 2020, pp. 72–88.

[41] D. Savage, Q. Wang, X. Zhang, P. Chou, and X. Yu, "Detection of money laundering groups: Supervised learning on small networks," in *Workshops at the Thirty-First AAAI Conference on artificial intelligence*, 2017.

[42] G. Pang, J. Li, A. van den Hengel, L. Cao, and T. G. Dietterich, "Anomaly and novelty detection, explanation, and accommodation (andea)," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 4145–4146.

[43] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8.

[44] M.-J. Segovia-Vargas *et al.*, "Money laundering and terrorism financing detection using neural networks and an abnormality indicator," *Expert Systems with Applications*, vol. 169, p. 114470, 2021.

[45] B. A. S. Hasan and K. Kelly, "Bayesian stress testing of models in a classification hierarchy," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[46] X. Zhu, X. Ao, Z. Qin, Y. Chang, Y. Liu, Q. He, and J. Li, "Intelligent financial fraud detection practices in post-pandemic era," *The Innovation*, vol. 2, no. 4, p. 100176, 2021.

[47] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.

[48] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, and Y. Qi, "Geniepath: Graph neural networks with adaptive receptive paths," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4424–4431.

[49] J. Wang, R. Wen, C. Wu, Y. Huang, and J. Xion, "Fdgars: Fraudster detection via graph convolutional networks in online app review system," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 310–316.

[50] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2703–2711.

[51] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 549–558.

[52] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi, "Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[53] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 2077–2085.

[54] Q. Zhong, Y. Liu, X. Ao, B. Hu, J. Feng, J. Tang, and Q. He, "Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network," *Proceedings of The Web Conference 2020*, 2020.

[55] D. Cheng, Z. Niu, and L. Zhang, "Delinquent events prediction in temporal networked-guarantee loans," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[56] T. Liang, G. Zeng, Q. Zhong, J. Chi, J. Feng, X. Ao, and J. Tang, "Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets," *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021.

[57] T. Zhao, C. Deng, K. Yu, T. Jiang, D. Wang, and M. Jiang, "Error-bounded graph anomaly loss for gnns," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[58] A. Singh, A. Gupta, H. Wadhwa, S. Asthana, and A. Arora, "Temporal debiasing using adversarial loss based gnn architecture for crypto fraud detection," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 391–396.

[59] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial-temporal attention," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[60] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, pp. 598–607.

[61] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, pp. 626–688, 2015.

[62] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, vol. 133, p. 113303, 2020.

[63] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.

[64] A. Goodge, B. Hooi, S.-K. Ng, and W. S. Ng, "Lunar: Unifying local outlier detection methods via graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6737–6745.

[65] Y. Wu, H. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, pp. 9214–9231, 2021.

[66] L. Xiong, B. Póczos, and J. Schneider, "Group anomaly detection using flexible genre models," *Advances in neural information processing systems*, vol. 24, 2011.

[67] R. Chalapathy, E. Toth, and S. Chawla, "Group anomaly detection using deep generative models," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, 2019, pp. 173–189.

[68] A. Belhadi, Y. Djenouri, G. Srivastava, A. Cano, and J. C.-W. Lin, "Hybrid group anomaly detection for sequence data: application to trajectory data analytics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9346–9357, 2021.

[69] R. Yu, X. He, and Y. Liu, "Glad: group anomaly detection in social media analysis," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[70] W. Song, W. Dong, and L. Kang, "Group anomaly detection based on bayesian framework with genetic algorithm," *Information Sciences*, vol. 533, pp. 138–149, 2020.

[71] K. M. Ting, B.-C. Xu, T. Washio, and Z.-H. Zhou, "Isolation distributional kernel: A new tool for point & group anomaly detection," *ArXiv*, vol. abs/2009.12196, 2020.

[72] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 665–674.

[73] H. Sun, J. Huang, J. Han, H. Deng, P. Zhao, and B. Feng, "gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 481–490.

[74] F. Jie, C. Wang, F. Chen, L. Li, and X. Wu, "Block-structured optimization for anomalous pattern detection in interdependent networks," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 1138–1143.

[75] M. Mongiovi, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "Netspot: Spotting significant anomalous regions on dynamic networks," in *Proceedings of the 2013 Siam international conference on data mining*. SIAM, 2013, pp. 28–36.

[76] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[77] M. Shao, J. Li, F. Chen, and X. Chen, "An efficient framework for detecting evolving anomalous subgraphs in dynamic networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2258–2266.

[78] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He, "Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proceedings of the Web Conference 2021*, 2021, pp. 2081–2091.

[79] L. Zhao and L. Akoglu, "On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights," *Big Data*, 2021.

[80] A. F. Colladon and E. Remondi, "Using social network analysis to prevent money laundering," *Expert Systems with Applications*, vol. 67, pp. 49–58, 2017.

[81] U. Desai, S. Bandyopadhyay, and S. Tamilselvam, "Graph neural network to dilute outliers for refactoring monolith application," in *Proceedings of 35th AAAI Conference on Artificial Intelligence (AAAI'21)*, 2021.

[82] E. Kurshan and H. Shen, "Graph computing for financial crime and fraud detection: Trends, challenges and outlook," *International Journal of Semantic Computing*, vol. 14, no. 04, pp. 565–589, 2020.

[83] B. Dumitrescu, A. Băltoiu, and Ş. Budulan, "Anomaly detection in graphs of bank transactions for anti money laundering applications," *IEEE Access*, vol. 10, pp. 47 699–47 714, 2022.

[84] X. Sun, W. Feng, S. Liu, Y. Xie, S. Bhatia, B. Hooi, W. Wang, and X. Cheng, "Monlad: Money laundering agents detection in transaction streams," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 976–986.

[85] M. Starnini, C. E. Tsourakakis, M. Zamanipour, A. Panisson, W. Allasia, M. Fornasiero, L. L. Puma, V. Ricci, S. Ronchiadin, A. Ugrinoska *et al.*, "Smurf-based anti-money laundering in time-evolving transaction networks," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21*. Springer, 2021, pp. 171–186.

[86] A. Mohan, P. Karthika, P. Sankar, A. Peter *et al.*, "Improving anti-money laundering in bitcoin using evolving graph convolutions and deep neural decision forest," *Data Technologies and Applications*, no. ahead-of-print, pp. 1–17, 2022.

[87] N. Rajput and K. Singh, "Temporal graph learning for financial world: Algorithms, scalability, explainability & fairness," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4818–4819.

[88] I. Alarab and S. Prakoonwit, "Graph-based lstm for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data," *Neural Processing Letters*, pp. 1–19, 2022.

[89] W. W. Lo, S. Layeghy, and M. Portmann, "Inspection-l: Practical gnn-based money laundering detection system for bitcoin," *arXiv preprint arXiv:2203.10465*, 2022.

[90] P. Ni, Q. Yuan, R. Khraishi, R. Okhrati, A. Lipani, and F. Medda, "Eigenvector-based graph neural network embeddings and trust rating prediction in bitcoin networks," in *Proceedings of the Third ACM International Conference on AI in Finance*, 2022, pp. 27–35.

[91] G.-Y. Sheu and C.-Y. Li, "On the potential of a graph attention network in money laundering detection," *Journal of Money Laundering Control*, vol. 25, no. 3, pp. 594–608, 2022.

[92] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

Dawei Cheng is an associate professor with the department of computer science and technology, Tongji University, Shanghai, China. Before that, Dawei was a postdoctoral associate at MoE key lab of artificial intelligence, department of computer science, Shanghai Jiao Tong University. He received the Ph.D. Degree in computer science from Shanghai Jiao Tong University, Shanghai, China. His research fields include data mining, graph learning and big data in finance.

Yujia Ye received her BSc degree in computer science and technology from Tongji University, Shanghai, China. She will pursue her master degree of computer science in National University of Singapore, Singapore. Her research interests include data mining, graph neural network and big data applications.

Sheng Xiang is a PhD candidate in the Center for Artificial Intelligence, major in Computer Science, University of Technology, Sydney (UTS). He received his BSc degree in Bioinformatics Engineering from Shanghai Jiao Tong University. His research interests include graph machine learning in finance, graph generative algorithm, bipartite graph processing, and dynamic graphs.

Zhenwei Ma is senior data scientist with the Research and Development Center of Financial Safety, China UnionPay Co., Ltd. He received his master degrees in computer science from Shanghai Jiao Tong University and the BSc degree in mathematics from Huazhong University of Science and Technology. His research interests include credit card fraud detection and money laundering detection.

Ying Zhang is a Professor and ARC Future Fellow (2017- 2021) at Australia Artificial Intelligence Institute (AAII), the University of Technology, Sydney (UTS). He received his BSc and MSc degrees in Computer Science from Peking University, and PhD in Computer Science from the University of New South Wales. His research interests include query processing and analytics on large-scale data with focus on graphs and high dimensional data.

Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995. He is currently a Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests include concurrency theory, formal verification of software, service-oriented computing, big data in finance, intelligent systems, financial risk management and big data computing.