Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4908-4924 Received 21-10-2025; Accepted 21-01-2026; Published online 28-01-2026

Detecting Fraudulent Communities in Financial Networks Using Hybrid Classification and Ranking Approaches

Alia Ayoub a.magdy@fci-cu.edu.eg
Faculty of Computers and Information-
Cairo University – Giza
Egypt.

Ayman El-Kilany a.elkilany@fci-cu.edu.eg
Faculty of Computers and Information-
Cairo University – Giza
Egypt.

Hatem El Kadi hkadi@fci-cu.edu.eg
Faculty of Computers and Information-
Cairo University – Giza
Egypt.

Corresponding Author: Alia Ayoub

Abstract

Suspicious communities refer to networks or organizations that display untypical behaviors in different fields such as in cyber security, social networks, and finance. These groups are also identified by the peculiar patterns of communication, abnormal transactions rates, and links to established criminal aspects. The well-coordinated and deviant characteristics of the members, including inconsistent timing and amount of interaction are often the indicators of possible fraudulent plots or money-laundering. It is important to identify these communities to detect and intervene on the illegal activities at an early stage. The paper presents a model of the detection and identification of the suspicious communities engaging in money-laundering transactions. The proposed framework identifies the highly suspicious nodes through a classification layer first and then identifies the suspicious communities around the suspicious nodes using three different algorithms. The algorithms have various strategies of ranking and classification to identify suspicious communities. The suggested framework was tested on two banking datasets and it was found to be able to find fraudulent communities with a high level of success.

Keywords: Money Laundering (ML), Anti-Money Laundering (AML), Hyperlink Induced Topic Search (HITS), Community detection, Random Walk with Restart (RWR), EXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM).

Citation: Alia Ayoub, et al. Detecting Fraudulent Communities in Financial Networks Using Hybrid Classification and Ranking Approaches. **Advances in Artificial Intelligence and Machine Learning.** 2026;6(1):272. https://dx.doi.org/10.54364/AAIML.2026.61272

1. INTRODUCTION

The act of concealing illicit funds by placing, layering, and integrating them into the financial system is referred to as money laundering. In order to conceal the source of their funds, criminals are employing strategies like shell companies, mules, and smurfing, which is harming the integrity of the financial system and hindering economic development. Communities that endanger the security of the overall economic environment are social networks of individuals and organizations that cooperate to conceal the origin of illegal cash. Therefore, it is crucial to identify money laundering communities in order to protect the integrity of the world financial system. These hidden networks are capable of a variety of crimes, such as corruption, drug trafficking, and funding terrorism. Hidden organizations like these are discovered, and they will assist law enforcement organizations in tracing illicit funds and disrupting criminal enterprises. Furthermore, the early detection of money laundering groups aids in compliance with international norms, protects financial institutions from reputational and legal risks, and fosters public faith in economic systems. Concentrating on related organizations rather than individual transactions makes detection operations more productive and successful at identifying complex criminal-schemes.In response, worldwide anti-money laundering (AML) initiatives have increased, with an emphasis on transaction analysis and identification of suspicious groups. For this reason, we propose a novel approach that analyzes a customer's transaction history in order to identify any dubious activities. The initial identification of suspects is used as the basis for identifying the questionable communities. The identification of the suspicious communities makes it easier to identify unusual patterns and transaction clusters in a timely manner, and financial institutions and regulators may improve their response before the issue worsens. The identification of the high-risk network will allow banks to better monitor and control potential threats, hence lowering overall financial risk. Network analysis is able to reveal underlying relationships between parties, providing a more thorough understanding of how funds are transferred in sophisticated money laundering schemes. Overall, this approach helps to identify money laundering, but it also aids in breaking up and preventing future offenses. In order to identify questionable behavior, a multi-layered framework has been created. The first layer proposes a method for identifying dubious nodes. After that, three hybrid community detection methods were shown. The primary objective of these algorithms, which integrate the Random Walk with Restart (RWR) graph-based link analysis algorithm with the XGBoost machine learning algorithm, is to find potentially dangerous communities surrounding the nodes identified in the first layer. Then, the performance of these algorithms was assessed on two banking datasets of varying sizes and precision, and precision, recall, and F1-measure were calculated to compare their performance.

The rest of the paper is arranged as follows: Section 2 reviews the previously published related work, Section 3 introduces the proposed anti-money laundering framework, Section 4 shows the performance evaluation, Section 5 demonstrates the discussion, and finally, the conclusion and future work are presented in Section 6.

2. LITERATURE REVIEW

A modern body of research on money laundering detection is becoming increasingly focused on the application of community detection methods to identify abnormal behavior in a financial transactions network. Such techniques attempt to determine groups of related entities that can collaborate

to mask criminal financial business. In the literature review, a broad variety of strategies, starting with the traditional clustering algorithms like Louvian and ending with more advanced domain-specific adaptations that include the elements of time and attribute-based community analysis, is presented. In a bid to maximize the scalability and accuracy, researchers combine these techniques with graph theory, supervised learning, and distributed computing. The studies, by modeling the transaction data as graphs and taking advantage of community structures, are able to show the potential of the graph-based analytics to uncover any hidden relationships and more easily detect fraudulent behavior. In the next section, critical contributions made by the various papers will be summarized to show how the various methodologies have been utilized to identify money-laundering communities in the complex transaction environment. We will also demonstrate the state of the art methods in fraud detection using deep learning models.

2.1 Community Detection in the Literature

Community detection is a concept extensively studied across the literature. Authors in [1] used the Louvain algorithm on a large banking dataset of 33,491 non-fraudulent and 241 fraud transactions which resulted in 90 percent accuracy.

The methodology on the discovery of transfer communities that represent high-risk money-laundering in large networks of transactions was introduced in Paper [2]. To begin with, an extensive transaction graph is built by combining the edges. The transfers that are less likely to engage in money-laundering are then filtered using suspicious maximal connected sub graphs (MCSs). A Temporal Directed Louvain algorithm, combined with money-laundering patterns is then applied to the rest of the MCSs. The resultant sub graph is then divided into separate communities, all of which are assigned a score of the risk of money-laundering. Every step is implemented on a distributed Spark platform, in which the TD Louvain algorithm is parallelized and optimized.

The authors in paper [3] introduce a detection system that combines learning with network analysis to investigate group behavior. It starts with the random sampling of communities out of the transaction network and supplementing them with known communities that are suspicious, to form a training dataset. After training a classifier the system works with new activity where the party starting every transaction is used as a seed, the corresponding community is extracted, selected features are computed and either a suspicious or non-suspicious classification is provided. Suspicious communities are then sent to the intelligence analysts to be investigated.

A proposed system [4] of detecting the cases of fraud is based on the community-detection algorithm in an application in a web-based platform, which is an intermodal hub between bankers and customers. It uses Neo4j, which is a graph database, to search and filter the fraud cases. Information which is inputted by the authorized bankers are fed through a training module and stored to be accessed on-demand. Test data is then analyzed by a detection module in which a query builder Cypher query is used together with a response builder to detect fraudulent activity.

The paper [5] proposes a new local community-detection model, which takes advantage of the small clique to isolate community. Instead of starting community expansion with one seed node, the authors start with the largest-linked clique containing the seed (so that the seed is naturally integral

to the community that is identified). They suggest a Triangle-Based Community Expansion (TCE) algorithm which uses triangle-based scores of edges to drive community expansion.

A community search framework is suggested by Paper [6], which combines node embedding and a minimum spanning tree strategy. This method is made up of two phases. To begin with, a node-embedding model, which is named NEBRW, is built on top of a biased random walk and Skip-gram. In the second step, the community-search problem is re-modeled as a form of minimum-spanning-tree problem: given a query node and a desired community size, the challenge is to extract a connected sub graph, which minimizes the total distance between its nodes. A better Prim algorithm is called CSMST, and it builds a spanning tree on the chosen community nodes.

Paper [7] discusses a rapid community search algorithm on attributed graphs by posing the problem of Flexible Attributed Truss Community (Flexible Attributed Truss Community (F-ATC)). The F -ATC model also removes the strong condition of having a constant number of triangles, and can investigate a wide variety of community structures that are more realistic in terms of graph topology. The two heuristic algorithms are beam search-based candidate (k, d)-truss enumeration and maximizing an attribute-score function and a preprocessing stage (pre-computing all k-trusses) is used to further reduce query response time.

The paper [8] provides the forbidden-nodes aware community-search problem, which is a variant of the classic community-search, but the results are not shown including unwanted nodes. Three algorithms (k-core-based FORTE, k-truss-based FORTE and CW-based FORTE) incorporate constraints that guarantee that community members are within closer proximity to query nodes as opposed to forbidden nodes. Using measures like average shortest-path distance and conducted weighted, the method finds cohesive communities, and comprehensive experiments on real world problems have shown it is far more effective than traditional algorithms.

Although there has been a lot of improvement indicated in the above research, there are still a number of limitations that have not been addressed. The current methods mainly use either static graph-clustering methods or a pre-defined community structure, thus overlooking the dynamical and temporal characteristics of money-laundering activity. Other algorithms like the Louvain and its derivatives might be effective in modularity optimization, but will not be able to detect fine or dynamic fraud patterns, especially in sparse or overlapping communities. Furthermore, some of them rely on massive data-processing systems (e.g., Spark, Neo4j), which, though scalable, might not be suitable in certain respects (such as flexibility of fraud detection on a fine-grained scale). Others also assume the presence of well-labeled training data, something which is seldom true in real-world financial systems where fraud is frequently hidden and labels are scarce. Moreover, most of the detection systems examine communities individually, without considering the multi-hop influence and indirect relationship that might be central to laundering schemes. In order to address these limitations, the proposed framework combines graph-based link analysis and machine learning, considers temporal transaction patterns, and proposes recursive exploration strategies, which allow identifying not only tightly but also loosely connected suspicious communities. Its new design of hybrid, adaptive design seeks to increase accuracy in detection, lower the level of false positives, and reveal any hidden laundering designs that otherwise would remain invisible to the traditional techniques.

2.2 Detecting Fraud Tansactions Using Deep Learning Models

The fraud detection model suggested by [9] involves a Graph Neural Network that learns expressive node representations based on multi-hop neighborhood aggregation with attention and hybrid anomaly detection and reinforcement learning to address the issue of class imbalance and detect outliers on large transactional datasets. They utilized Yelp and Amazon inspired datasets to detect subconscious fraud movements. However, they don't focus on detecting money laundering patterns. They showed good results in comparison to the usual variants of GNN, like GCN, GAT, and GraphSAGE on real-world data, highlighting the importance of the end-to-end learning of graph representations to detect fraud patterns. The authors in [10] introduce a Graph Neural Network-based fraud detection model, which can overcome the imbalance in the classes, and noisy information by similarity-aware up-sampling and reinforced neighborhood aggregation, which give better discrimination between fraudulent and legitimate nodes in complicated transaction graphs. While authors in [11] demonstrate the methods to create aggregated spending profiles through the use of cluster and factor analysis of transactional behavior to summarize consumer spending patterns that may be utilized in the process of risk monitoring and the identification of abnormal variation of a financial activity. It utilizes cluster-derived features that can be used to capture holistic transaction properties that can be a pointer to fraudulent or suspicious activity. Paper [12] introduces a real-time system for finding financial fraud, using Graph Neural Networks (GNNs) with unsupervised anomaly detection techniques to spot tricky patterns in simulated bank transactions dataset in a dynamic graph. Paper [13] solves the problem of class imbalance while using GNN as GNN-based algorithms could poorly behaves when the label distribution of nodes is heavily biased, which is a common problem in financial fraud datasets. Paper [14] solves the problem of the concealing identity of fraudsters that confuse the GNN algorithms. As they usually behave like normal clients. They used Discriminative Feature Guided GNN with reinforcement learning to reduce the degree of confusion occurs to GNN models in detecting the fraud connections.

In contrast to this end-to-end deep learning architecture, our research considers explicit graph structural and ranking characteristics (e.g. HITS scores and community patterns) with gradient-boosted classifiers with more emphasis on interpretability, reduced training complexity and applicability to analyst-driven AML contexts.

3. THE PROPOSED FRAMEWORK

In this section, the proposed framework is presented in FIGURE 1, which consists of two main phases. The first stage entails preprocessing of data whereby raw tabular financial data are transformed into a graph format thus making them more appropriate when subjected to graph-based and machine-learning algorithms that are to identify suspicious nodes. The second stage focuses on the identification of suspicious nodes and clarification of peripheral suspicious groups, which is based on the result of the first stage. Data obtained in the previous phases are, then, tested and evaluated to determine the usefulness and efficiency of the recommended framework.

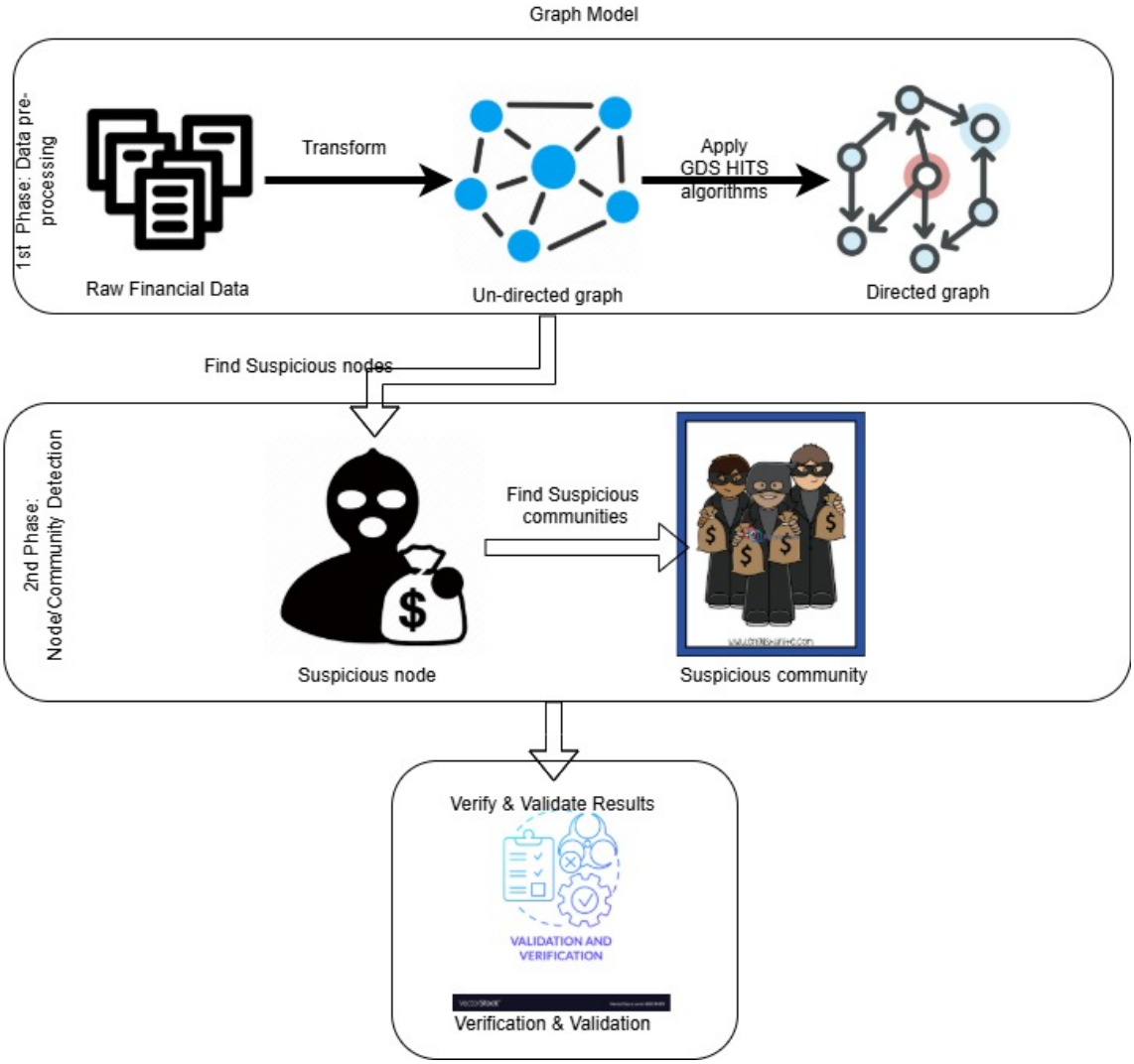

Figure 1: Proposed framework

3.1 First Phase: Data Pre-Processing

During this stage, the dataset gets to be ready in preparation to be used in the following stages. Based on the dataset presented in TABLE 1, the banking transactions that took place at a given time, i.e. on a daily basis, were converted into a graph format. In this graph, the vertex indicates customers or accounts involved in the transactions and the edge indicates the transaction direction (i.e. paid or received) and the amount of money transferred. After that, the HITS (Hyperlink-Induced Topic Search) algorithm is changed and used to process the financial transaction graph. The nodes (customers) are provided with two scores, which are the authority score (Depending on the amount of incoming money) and the hub score (Depending on the amount of outgoing money). In order to further describe transactional behavior, edge weights were added depending on the intensity

and frequency of money transfers. The weighted HITS algorithm prioritizes the ranking of nodes based on their authority score and hub score to detect the most questionable ones. In the long term, the behavior of the nodes will be tracked through the authority and hub scores that will be used to efficiently identify suspicious customers. Another form of the graph representation, in which the graphs of all the days/timestamps are combined into one graph representing the entire transaction history, is also created. Both variations of the graphs will be utilized in the algorithms of the second stage.

Table 1: Dataset Snap Shot of the complete transaction log

TX ID	SENDER ACCOUNT ID	RECEIVER ACCOUNT ID	TX_TYPE	TX AMOUNT	TIMESTAMP	IS_FRAUD	ALERT ID
2288	29	63	TRANSFER	65.55	27	FALSE	-1
2289	55	66	TRANSFER	13.73	27	TRUE	2
2290	94	66	TRANSFER	24.95	28	FALSE	-1
2291	94	34	TRANSFER	24.95	28	FALSE	-1
2292	94	49	TRANSFER	24.95	28	FALSE	-1
2293	94	18	TRANSFER	24.95	29	FALSE	-1
2294	94	8	TRANSFER	24.95	29	FALSE	-1
2295	94	82	TRANSFER	24.95	29	FALSE	-1
2296	94	93	TRANSFER	24.95	29	FALSE	-1
2297	94	94	TRANSFER	24.95	29	FALSE	-1
2298	94	73	TRANSFER	24.95	29	FALSE	-1

3.2 Second Phase: Node/Community Detection

This component is structured into two main parts: the detection of suspicious nodes, followed by the identification of suspicious communities.

3.3 Detecting Suspicious Nodes

This subsection includes the input of the pre-processed data of the directed graph of each day, with the added authority and hub scores, to three machine learning models: Isolation Forest, XGBoost, and LightGBM, to discover suspicious nodes on a given day/timestamp. Isolation Forest is used as an unsupervised baseline, which detects anomalies in case of limited labels, and XGBoost and LightGBM are supervised learning methods that can learn sophisticated fraud patterns. XGBoost focuses on the robustness and interpretability whereas LightGBM is more focused on scales and efficiency. TABLE 2 shows a sample of the training data. The names of the nodes are given in the leftmost column, and the names of the timestamps of each day are defined in the upper row. The authority and hub scores calculated in Phase 1 are contained in the cells of the table. These models were applied in Scikit-learn machine learning Python library. Judging by the outcomes that will be revealed in the performance evaluation section, it can be seen that XGBoost algorithm is superior to the other two models. XGBoost model deployed by us is able to detect a fraudulent node or

a normal node with an accuracy of about 91%. In this regard, we put more focus on precision, which is indicative of the capability of the model to reduce the false-positive predictions, which is a critical consideration in the money laundering detection, since the wrong classification of a legitimate node as a fraudulent node can be disastrous. Where recall can be used to gauge how well a given model can recognize all real positive examples, precision is more essential in such a situation. The evaluation measures of the three models are clearly shown for comparison in the performance evaluation at Section 4, in TABLE 4.

Table 2: A snap shot of the training data

Node_name	0	1	2	3	4	5
0	0	0	0.037433	0.035314	0	0
1	0.002399	0.266622	0.006698	0	0.002399	0
2	0.013749	0	0	0	0.013749	0.02278
3	0	0	0.006698	0	0	0
4	0.021509	0.048016	0.006698	0	0.021509	0.011497
5	0.034049	0	0.041626	0	0	0

3.4 Detecting Suspicious Communities

The next step following the identification of the suspicious nodes as explained in the previous subsection involves the next stage that focuses on identification of communities of the suspicious nodes. This is done by implementing three different algorithmic strategies. The initial algorithm will utilize the Random Walk with Restart (RWR) algorithm in the detection of suspect communities being generated by a specific node of fraudulent behavior; it will hereafter be referred to as the Time-Aware RWR Community Detector. The second algorithm adds a customized feature to the XGBoost model that identifies and analyses the first and second-degree neighbors of a suspicious node; the optimized model is henceforth referred to as XGBoost-NHD (Node Hub Detector). The third algorithm will combine both RWR and XGBoost into a hybrid framework which is aimed at promoting the detection of dangerous communities. This hybrid approach is known as the Walk-and-Classify algorithm, which focuses on the sequential exploitation of RWR to explore graphs and then use XGBoost to conduct classification of the obtained results. All of these algorithms will be discussed in the following subsections.

3.5 Algorithm 1: Time-Aware RWR Community Detector

Suspicious communities are the groups that represent abnormal or harmful behavior, which is a common occurrence in the financial, social, and cyber-security realms. The discovery of such communities helps to expose any fraudulent activity including money-laundering schemes. Random Walk with Restart (RWR) is a graph-based analytics model which derives suspicious communities through a single-source RWR score vector and produces personalized rankings on nodes. Being a link-analysis method, RWR uses node-to-node proximities within a network, usually Python based with Scipy and Numpy modules, and returns personalized rankings based on the source score vector. Community discovery techniques group nodes together based on the network distance and RWR

is frequently used to evaluate a node in context e.g. ranking web pages or evaluating users in a social network. RWR aggregates the standard random walk procedures by periodically revisiting the seed node, which increases the traversal efficiency and ranking accuracy. This process helps in determining powerful nodes in social groups through the repeated propagation of suspicious nodes in order to detect fraud formations as indicated in [15], [16]. The RWR algorithm is used in an iterative manner in order to find the most suspicious nodes that directly relate to a given seed node on a particular day. RWR process is initiated by the seed node and gets the nearest related node in the graph of the same day. The resulting node is then considered the new seed and the process of recursive application goes on till all the suspicious nodes are identified. The combination of these nodes will create a suspicious community which can then be analyzed in terms of behavior to identify fraudulent behavior. FIGURE 2 demonstrates an example of a fraud community that was obtained on the basis of the dataset. An example is given that on introducing node 55 into the RWR algorithm, the algorithm generates a rank ordered list of suspected nodes in decreasing order of relevance scores. The highest-ranked node, 66, is then re-entered into the RWR algorithm, and produces node 67, and so on in a recursive way. This step is applied on the dataset that is related to a particular time point (in this instance that is a transaction day) at which the fraud was committed.

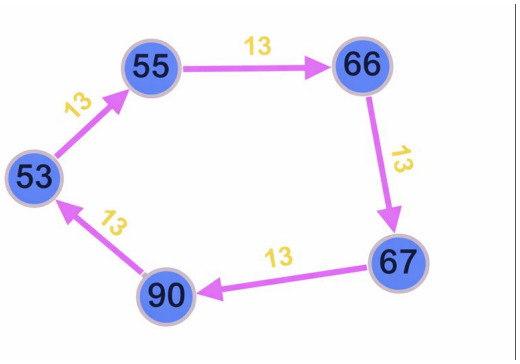

Figure 2: Example of a fraud community (Cycle pattern)

The algorithm exits when any of the following conditions are met: the relevance score falls below 0.5, the newly found node has already been discovered and added to the community, or the number of detected nodes reaches a predefined limit ***n***, which can be adjusted as needed.

3.6 Algorithm 2: XGBoost-NHD (Node Hub Detector)

The XGBoost machine learning model has been used in this algorithm to identify suspicious communities. Before its implementation in our framework, a custom module had been created to obtain the first and second neighbor (i.e. 1-hop and 2-hop hub) of a given fraud node. This neighborhood extraction module will be used alongside the XGBoost to improve the identification of communities that are identified with frauds.

It starts with the detection of the 1st and 2nd -degree neighbors of a particular fraud node, with the full dataset of transactions history graph. These nodes are then fed into an XGBoost classifier to determine their chances of being fraudulent. XGBoost classifier is already trained on the prediction of the behavior of fraudulent nodes since it has already been trained on prediction of the fraud node.

TABLE 2 provides a snap shot of the training data. Depending on the output of the classifier, nodes are either referred to as a fraud or normal. These nodes that have been detected to be fraudulent help to create a suspicious community whereby the transaction history is analyzed to determine common links amongst them. In every iteration, there are connected suspicious nodes and scattered suspicious nodes that can be discovered by the classifier. Besides, when suspicious nodes have no immediate connections, historical transactions are utilized to determine their indirect connections and narrow down the fraud community. Their transaction history is studied in situations where the fraud nodes are already linked but have not yet developed a complete community where the missing nodes are found and included to complete the structure of the community. In case fraud nodes are scattered and have no obvious relationships, these nodes are not included into the existing community, as they might be a part of different fraud networks. Ideally, in case the fraud nodes are properly classified and organize as a separate community, the result is seen as true and no additional adjustments are needed. To help illustrate, take a fraud node that is named node 55. When node 55 is transmitted to the XGBoost-NHD (Node Hub Detector), the algorithm recalls the first and second-degree neighbors of node 55. These adjacent nodes are then fed to the XGBoost classifier that determines whether each of the nodes is a fraudulent or normal node. The results are shown in TABLE 3, which shows the neighborhood level of each node as well as its prediction of the prediction of the fraud (1 is fraud and 0 is normal). Due to the description of the transaction history log, it is possible to build connections between the detected fraudulent nodes and form the corresponding suspicious community, as in the example provided above in FIGURE 2.

Table 3: Output snapshot of the XGBoost-NHD (Node Hub Detector)

Node_NameLevel	Fraud?
550	1
971	0
661	1
671	1
991	0
491	0
531	1
901	1
642	0
962	0
342	0
702	0

This paradigm allows the effective identification of suspicious communities and possible patterns of money laundering. The algorithm is created to find community structures in a weighted network, which is optimized on precision, recall, and F1-score. The findings of the evaluation, as per two benchmark datasets, are shown in the next section.

3.7 Algorithm 3: Walk-and-Classify Algorithm

In this strategy, we use an ensemble technique, which combines the advantages of the previous two algorithms, which is to use Random Walk with Restart (RWR) and the XGBoost classifier. It is an approach that uses the graph of the entire transaction history in the analysis of a suspicious node. When RWR is given a suspicious node, it can give a set of adjacent nodes and a probability distribution of the relevance or proximity of the adjacent node to the given node. Out of this sample, the most relevant n nodes are chosen and handed over to the XGBoost classifier that will be used to label each of the nodes as a fraud or a non-fraud. The fraudulent nodes that have been received are then clustered into the suspicious community of the original node. Various values of n were experimented with where n of 20 was identified to be adequate in community extraction. As an example, given a suspicious node (say, Node 55), Walk-and-Classify algorithm will apply RWR to initially view the top 20 closest neighboring nodes based on whole transaction history. These nodes are further categorized with XGBoost to establish whether they are frauds or not. The nodes that are found to be fraudulent are clustered together with the original node to create a suspicious community. As an illustration, Nodes 66, 67, 53, and 90 are considered to be frauds, and they, together with Node 55, comprise the found community that can possibly denote a pattern of money laundering in a specific cycle as in FIGURE 2.

The ensemble method improves the detection of fraud through the integration of network based proximity analysis (RWR) with machine learning based classification (XGBoost). It will be found especially useful in revealing pattern based on cyclic money laundering, but it is also less sensitive to the identification of fan-in patterns as in FIGURE 3.

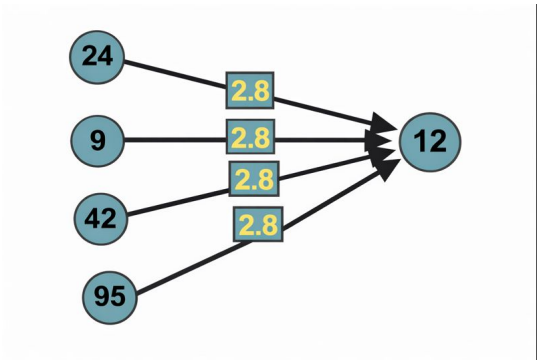

Figure 3: Example of a fraud community (Fan-in pattern)

3.8 Performance Evaluation

The key aim of this section is to determine and analyze the usefulness and the stability of the proposed algorithms in the correct identification of suspicious nodes and communities of the fraud activity and money laundering in particular. The review will determine the capacity of every algorithm to detect fraudulent acts accurately but with minimum false positives and false negatives. Moreover, the results of the three algorithms used in our framework are compared to determine the performance of the algorithm and its strengths in identifying complex fraud patterns.

Each of the algorithms is tested with both of our two synthetic banking[1] graphs of size100vertices-10Kedges, and 1Kvertices-100Kedges.The datasets form 17,145 transactions, and 117,534 transactions respectively. The latter corresponds to extremely high density of transactions, that is, a high number of financial transactions per account, and is meant to load the proposed approaches with a heavy load of transactions. In addition, synthetic data must be employed due to the scarcity of real-world financial transactional data that sufficiently satisfies the demands of our research problem, namely privacy, confidentiality, and regulations. Both data are organized set of financial transactions, and each record corresponds to one financial transaction between two accounts. It contains attributes that are important like transaction ID, sender account ID, receiver account ID, type of transaction, amount of transaction and the time of transaction. It also has an IS_FRAUD field that shows whether the transaction is fraudulent (1) or legitimate (0) and an ALERT_ID that identifies flagged transactions with a particular fraud alert as demonstrated earlier in TABLE 1. This information forms the basis of suspicious identification and money laundering trends by both graph and machine learning.

TABLE 4 presents the results of the evaluation of the fraud node detection algorithms: XGBoost, LightGBM and Isolation Forest that are used in the second step of our framework to obtain the suspicious/fraud nodes. This is what 5-cross fold validations of both datasets give.

Table 4: Evaluation results for the fraud node detection algorithms: XGBoost, LightGBM, and Isolation Forest

Data set	Algorithm utilized	Precision	Recall	F1-measure
100vertices-10Kedges	XGBoost	**0.85**	**0.84**	**0.84**
	LightGBM	0.50	0.55	0.52
	Isolation Forest	0.40	0.45	0.42
1Kvertices-100Kedges	XGBoost	**0.917**	**0.902**	**0.9096**
	LightGBM	0.511	0.562	0.5325
	Isolation Forest	0.250	0.495	0.332

Conversely, the results of the evaluation of the three different algorithms employed to form the suspicious communities as defined in section 3, are presented in TABLE 5, TABLE 6, and TABLE 7 of Time-Aware RWR Community Detector, XGBoost-NHD (Node Hub Detector), and Walk-and-Classify algorithm, respectively.

Table 5: Evaluation results of the 1st algorithm (Time-Aware RWR Community Detector)

Data set	Precision	Recall	F1-measure
100vertices-10Kedges	1.0000	1.0000	1.0000
1Kvertices-100Kedges	0.98	0.9218	0.95

In order to compare the performance of the three algorithms, TABLE 8 shows the average precision, recall, and F1-score of the three algorithms having been tested across the two datasets.

[1] IBM: AML-Anti-Money-Laundering-Data, https://ibm.ent.box.com/v/AML-Anti-Money-Laundering-Data, last accessed 2023/11/22.

Table 6: Evaluation results of the 2nd algorithm (XGBoost-NHD (Node Hub Detector))

Data set	Precision	Recall	F1-measure
100vertices-10Kedges	1.0000	0.8947	0.9415
1Kvertices-100Kedges	1.0000	0.8605	0.9082

Table 7: Evaluation results of the 3rd algorithm (Walk-and-Classify algorithm)

Data set	Precision	Recall	F1-measure
100vertices-10Kedges	1.0000	0.8140	0.8870
1Kvertices-100Kedges	0.9342	0.7368	0.8129

Table 8: Average scores of the three community detection algorithms

Algorithm	Precision	Recall	F1-measure
1^{st}:(Time-Aware RWR Community Detector)	**0.99**	**0.96**	**0.975**
2^{nd}: (XGBoost-NHD)	1.0	0.8776	0.92485
3^{rd}:(Walk-and-Classify algorithm)	0.9671	0.7754	0.84995

4. DISCUSSION

The analysis of the evaluation results, as outlined in TABLE 4 – TABLE 8, indicate the efficiency of the suggested framework in detecting fraud nodes as well as suspicious communities in two distinct datasets of different sizes. This was found to be the case with the XGBoost among the models of detecting fraud nodes (TABLE 4) with the model having a higher precision, recall and F1-score than the LightGBM and Isolation Forest models. This is due to the fact that XGBoost is capable of dealing with imbalanced data and derivation of complex non-linear patterns in financial transactions, which are typical in the case of fraud detection.

Concentrating on the community detection algorithms (TABLE 5 through TABLE 7), the Time-Aware RWR Community Detector showed the best average performance with almost perfect precision and recall to both datasets. This is because it explores a graph based at certain timestamps of transaction so that it can determine closely related suspicious nodes in a time context. Its power is in its ability to identify systematic fraud patterns, particularly cycle-based laundering schemes that take place within limited periods of time. Conversely, it can be considered a weakness in cases where the fraudulent transactions are divided in various timestamps that are handled by the XGBoost-NHD and Walk-and-Classify algorithms.

The XGBoost-NHD method also did a great work, especially when the accuracy is concerned and scored a perfect result in both sets of data. It is effective because it uses the neighborhood information specifically, first and second-degree node relationship and learned classification boundary using the XGBoost model. This allows it to identify suspicious communities with high precision

but its slightly lower recall than the Time-Aware RWR implies that certain fraud nodes that are not closely connected with the seed may be overlooked.

Walk-and-Classify algorithm that combined both RWR and XGBoost showed high precision but significantly low recall particularly on the bigger dataset. While the top **n** RWR-ranked nodes offer both; structural relevance and predictive utility for machine learning, this approach may limit coverage when fraudulent nodes are less centrally connected or absent from the highest-ranked group. However, the algorithm is still useful to detect dense communities that are cycle-based but is not as sensitive to dispersed and fan-in structures.

All in all, the Time-Aware RWR Community Detector is the most robust solution that can be used with time-localized laundering activities which can be followed, and the XGBoost-NHD is the most high-precision solution that can be applied to tightly connected fraudulent nodes. The Walk-and-Classify algorithm is a hybrid method that balances structural analysis with classification, trading off recall. This confirms the importance of integrating graph analysis with learning-based methods and suggests that different community structures may require variations for optimal detection.

5. CONCLUSION AND FUTURE WORK

In this paper, a comprehensive architecture is proposed for detecting suspicious nodes and their associated communities potentially involved in money laundering activities. The raw financial banking dataset is transformed into a graph structure, where nodes represent entities and edges represent transactions. To enhance node features beyond basic transactional attributes, such as the amount of money sent or received, the HITS (Hyperlink-Induced Topic Search) algorithm is applied to compute authority and hub scores for all nodes.

Three different algorithms are applied and tested to detect money laundering communities:

The Time-Aware RWR Community Detector uses the Random Walk with Restart (RWR) algorithm to a transaction sub graph that is filtered by a given fraud timestamp, e.g. a known day of activity. Starting with a seed node of suspicion, the algorithm determines the relatively similar nodes in the transactions of that day and creates a suspicious community on the basis of interaction relevance. It can be done using more timestamps related to the fraud to create the result that can be combined with the previous one to create a larger suspicious community or analyzed independently to record the changing pattern of laundering over time.

The XGBoost-NHD (Node Hub Detector) uses XGBoost machine learning algorithm with a custom Python module that obtains the 1 st and 2 nd -degree neighbors of a suspicious seed node. Such neighbors are sent to the XGBoost classifier to determine whether a particular node is fraud or not. The suspicious community is then created as the identified fraudulent nodes are grouped together.

The Walk-and-Classify algorithm presents a hybrid algorithm that combines the RWR with the XG-Boost. Utilizing a suspicious seed node and a complete history of the transactions, RWR identifies the most closely related n nodes to the seed node. The XGBoost is then used to classify these nodes based on their likelihood of being fraudulent. This leaves a suspicious community comprising of the nodes identified to be fraudulent.

All of the algorithms are tested and evaluated using two synthetic banking datasets with varying sizes: 100 vertices and 10K edges, and 1K vertices and 100K edges. According to experimental results, Time-Aware RWR Community Detector has the best performance in terms of precision, recall, and F1-score.

In the future, we intend to expand the size of the dataset in order to further confirm the scalability and strength of the proposed approach. Moreover, we will investigate the usage of Graph Neural Network (GNN)-based models; including GCN (Graph Convolutional Network) and GraphSAGE (Graph Sample and Aggregation) if we have the required computational resources for running these graph deep learning models. Utilizing the results of this paper as a great milestone, we can compare its results with the results of the GNN and its counterparts to extend our framework with a robust comparison.

6. STATEMENTS & DECLARATIONS

Data Availability

"This study uses a publicly available dataset, cited in the footnotes of Section 4 (Performance Evaluation)."

Competing Interests

"The authors declare that they have no conflicts of interest."

Funding

"The authors did not receive any financial or non-financial support for the submitted work."

Author Contributions

"All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [AA], and [AE]. Project Supervision and Validation were done by [AE], and [HE]. The first draft of the manuscript was written by [AA] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript."

Acknowledgements

"The authors would like to acknowledge the valuable discussions and support provided during the course of this research."

References

[1] Mardiansyah H, Suwilo S, Nababan EB, Efendi S. The Role of Louvain-Coloring Clustering in the Detection of Fraud Transactions. International Journal of Electrical and Computer Engineering (IJECE). 2024;14:608-616.

[2] Li X, Cao X, Qiu X, Zhao J, Zheng J. Intelligent Anti-Money Laundering Solution Based upon Novel Community Detection in Massive Transaction Networks on Spark. 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD). IEEE. 2017:176-181.

[3] Savage D, Wang Q, Chou P, Zhang X, Yu X. Detection of Money Laundering Groups Using Supervised Learning in Networks. 2016. ArXiv preprint: https://arxiv.org/pdf/1608.00708

[4] Sarma D, Alam W, Saha I, Alam MN, Alam MJ, Hossain S. Bank Fraud Detection Using Community Detection Algorithm. In 2020 second international conference on inventive research in computing applications (ICIRCA). IEEE. 2020:642-646.

[5] Hamann M, Röhrs E, Wagner D. Local Community Detection Based on Small Cliques. Algorithms. 2017;10:90.

[6] Liu J, Wang D, Feng S, Zhang Y. An Approach of Community Search with Minimum Spanning Tree Based on Node Embedding. Complexity. 2021;2021:6673444.

[7] Matsugu S, Shiokawa H, Kitagawa H. Fast Algorithm for Attributed Community Search. J Inf Process. 2021;29:188-96.

[8] Wang C, Zhu J. Forbidden Nodes Aware Community Search. In Proceedings of the AAAI Conference on Artificial Intelligence 2019;33:758-765.

[9] Polu OR, Chamarthi B, Chowdhury T, Ushmani A, Kasralikar P, et al. Graph Neural Networks for Fraud Detection: Modeling Financial Transaction Networks at Scale. In 2nd International Conference on Sustainable Business Practices and Innovative Models (ICSBPIM-2025). Atlantis Press. 2025:712-729.

[10] Chen J, Chen Q, Jiang F, Guo X, Sha K, et al. SCN_GNN: A GNN-based Fraud Detection Algorithm Combining Strong Node and Graph Topology Information. Expert Syst. Appl. 2024;237:121643.

[11] Hu X, Chen H, Chen H, Liu S, Li X, et al. Cost-Sensitive GNN-Based Imbalanced Learning for Mobile Social Network Fraud Detection. IEEE Trans Comput Soc Syst. 2024;11:2675-2690.

[12] Rasul I, Shaboj SI, Rafi MA, Miah MK, Islam MR, Ahmed A. Detecting Financial Fraud in Real-Time Transactions Using Graph Neural Networks and Anomaly Detection. J Econ Financ Account Stud. 2024;6:131-142.

[13] Liu Y, Ao X, Qin Z, Chi J, Feng J, Yang H, He Q. Pick and Choose: A GNN-Based Imbalanced Learning Approach for Fraud Detection. Proceedings of the Web Conference 2021. 2021:3168-3177.

[14] Zhang J, Xu Z, Lv D, Shi Z, Shen D, et al. DiG-In-GNN: Discriminative Feature Guided GNN-Based Fraud Detector Against Inconsistencies in Multi-Relation Fraud Graph. In Proceedings of the AAAI conference on artificial intelligence. 2024;38:9323-9331.

[15] Tong H, Faloutsos C, Pan JY. Fast Random Walk With Restart and Its Applications. In Sixth international conference on data mining (ICDM'06). IEEE. 2006;613-622.

[16] Jin W, Jung J, Kang U. Supervised and Extended Restart in Random Walks for Ranking and Link Prediction in Networks. PloS one. 2019;14:e0213857.