Parallelized Louvain-Based Community Detection and AntiBenford Subgraph Mining for Financial Fraud Detection in Transaction Networks

Amit Pimpalkar [1*], Rachna K. Somkunwar [2], Soham P. Chinchalkar [3], Kartik M. Katakdound [2], Atharva S. Bhide [4], Yogeshwar M. Patil [2]
[1] School of Computer Science and Engineering, Ramdeobaba University, Nagpur, Maharashtra, India
[2] Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, India
[3] Information Technology Department, Arizona State University, Tempe, Arizona, Usa
[4] Viterbi School of Engineering, University of Southern California, Los Angeles, Usa
E-mail: amit.pimpalkar@gmail.com, rachnasomkunwar12@gmail.com, sohamchinchalkar@gmail.com, kartik.katak@gmail.com, batharva191@gmail.com, yogi160721@gmail.com
*****Corresponding author

Keywords: parallelism, fraud detection, banking transactions, subgraph mining, louvain modularity, Benford's Law, statistical features

Received: February 16, 2025

Detection of financial fraud remains an issue of concern since there is always a dynamic nature in the illegal patterns of transactions that conceal themselves in massive banking systems. The paper is a hybrid system that integrates the use of the AntiBenford statistical deviation analysis and graph-based community detection to identify the most appropriate methods to identify suspicious behavior. A transaction network graph forms the basis of the approach. Monetary flows, or directed edges, exist between each entity, which are an account, bank or merchant. AntiBenford module identifies digit anomalies of transaction records, whereas the augmented Louvain community detection algorithm, which runs in parallel, identifies well-knit communities, which are indicative of money laundering or collusion. We evaluated the model using the IBM Transactions Anti Money Laundering (AML) dataset where we obtained an accuracy of 96.57 percent more than traditional machine learning, rule-based and statistical anomaly methods. The reliability and interpretability of the method are validated by ROC AUC and precision-recall analysis. Combining statistical anomaly assessing with graph mining, this paper provides a scalable, flexible remedy to network fraud.

Povzetek: Predstavljen je paraleliziran pristop za zaznavanje finančnih goljufij v transakcijskih omrežjih, ki združuje Louvainovo odkrivanje skupnosti z analizo odstopanj od Benfordovega zakona na podgrafih. Predlagana metoda izboljša skalabilnost, zaznavno učinkovitost in interpretabilnost pri analizi velikih omrežij.

1 Introduction

The banking sector has revolutionized because of the digital financial services, which have become customer-convenient. But this advancement has presented new fraud threats. Fraud is very costly to the banks, kills trust, and destroys the financial ecosystem. Financial fraud [1] remains a significant threat to the banking industry, which requires robust and dynamic detection tools. Conventional, rule-based systems of detection compare individual transactions to predetermined rules and are also incapable of tracking changing fraud patterns. Fraudsters constantly work out more complex ways, taking advantage of loopholes and evading controls. The traditional methods also produce false positives, blocking legitimate transactions. Thus, more powerful, more adaptable detectors are required. Some of the developments of data mining and network analysis and particularly graph mining can reveal some of the latent patterns in networks of transactions. This paper discusses a model that combines graph mining with high-order statistics in identifying frauds. The suggested system is based on parallel greedy community detection and Louvain modularity optimization to create a transaction network that reflects the relationships between entities and indicates anomalies. Benford Law [2] is used as statistical prism for analyzing attributes of transactions and spot variance that can be used to identify fraud. A graph refers to a mathematical form of nodes (vertices) that are interconnected by edges. Customers, accounts, merchants, or devices may be the nodes, whereas money transfer, logins, or purchases may be the edges in banking. The fact that a transaction network can be created as a graph reveals the latent structures of financial activity and allows the identification of anomalies that cannot be detected by looking at transactions separately. Figure 1 is based on literature review and represents a graph of different types of frauds, including API fraud, stolen or forged credit cards, website cloning, ATM fraud, payment fraud, delivery fraud, referral and promo fraud, phone fraud, call

forwarding fraud, multiple transfer scams, and money laundering. Customers, accounts, merchants and devices are represented as nodes and their interactions are represented as edges. Bank transaction fraud is better identified with the help of a graph analysis. We can identify suspicious trends and abnormalities of relationships that would not have been discovered otherwise by modeling relationships between customers, accounts, merchants, and devices. Identifying communities of interest, or groups of entities that behave similarly, is one of the key elements of analyzing graphs. If we observe a concentration of accounts with unusually high transaction frequencies or non-geographically common active accounts, it may be an indication of fraudulent activity. One-way fraudsters attempt to conceal their footprint by establishing relationships between accounts that appear to have no commonalities, although graph analysis can help us identify these unusual relationships. Another advantage of using graphs is that one can trace the spread of fraud. In some instances, fraud can be spread within a network when criminals use links between hacked accounts.

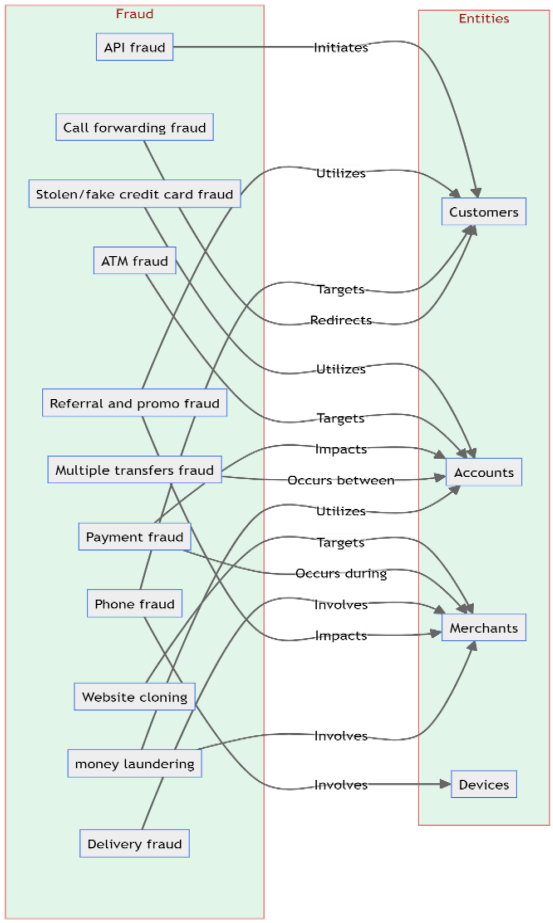


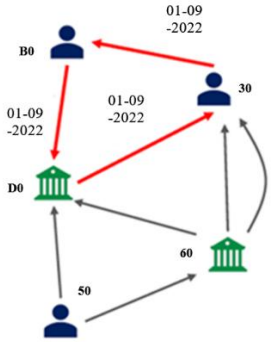
Figure 1: Sample graphical representation of fraud activities in banking transactions

These transactions, as represented in Figure 2, show that there is cyclic transfer, an obligatory feature of laundering activity (a), graphed out as a graph (b). Every exchange creates an arrowed edge between two nodes, which are accounts or banks. A combination of these edges gives a closed circuit with the flow between the banks being 394 $\rightarrow$ 31811 $\rightarrow$ 31813 $\rightarrow$ 394. Such loops in the graph depict funds flowing through various intermediaries and back to their origin, which is likely to mask the source of the money. The regularities of the transactions, the use of the same currencies, and the insignificant temporal distances also reinforce the signal of anomaly. In applying the AntiBenford-Louvain framework to this subgraph, repetition and symmetry increase the modularity contribution of the nodes concerned in the system, allowing it to be isolated as a suspicious attribute of layering or round-tripping in money laundering involving the nodes involved. The analysis of the graph will give us the ability to proactively identify possible victims and prevent them from being impacted by taking precautionary measures.

Although graph analysis is an effective tool, it can be even more effective when integrated with the more sophisticated statistical means. One of them involves the use of Benford Law, which explains why there is an uneven distribution of leading digits in naturally occurring datasets. Examples of such attributes used to identify aberrant behavior and determine fraud include deviations from Benford's Law in the attributes of transactions, such as the value of transfers or the value of an invoice. With the help of the combination of graph-based analysis and effective statistical tools, financial asset managers can outsmart fraudsters and keep the funds of their clients safe. Fraud detection is no longer done using rule-based systems but also incorporates new sophisticated methods such as graph mining and sophisticated statistics. It is a study based on network analysis and statistical rigor to reveal detailed trends in data on transactions based on recent developments. By combining these strategies, we will develop a robust and flexible system of fraud detection that can assist institutions to keep in advance of default risks and ensure the financial well-being of their customers.

Timestamp	From Bank	From Account	To Bank	To Account	Amount Received	Receiving Currency	Amount Paid	Payment Currency	Payment Format
01-09-2022 00:14	394	B0	31811	D0	21126.92	US Dollar	21126.92	US Dollar	Cheque
01-09-2022 00:23	1231	30	21414	D0	217	US Dollar	217	US Dollar	ACH
01-09-2022 00:05	202	40	202	40	14903.78	US Dollar	14903.78	US Dollar	Credit Card
01-09-2022 00:23	31811	D0	31813	30	21126.92	US Dollar	21126.92	US Dollar	Wire
01-09-2022 00:28	1638	50	1638	50	4755.32	US Dollar	4755.32	US Dollar	Reinvestment
01-09-2022 00:35	31813	30	394	B0	21126.92	US Dollar	21126.92	US Dollar	Credit Card
01-09-2022 00:16	31993	60	31993	60	3065.41	US Dollar	3065.41	US Dollar	Reinvestment

(a)



(b)

Figure 2: Financial transactions samples in (a) tabular format and in (b) graph format

In the analysis, a strong system is introduced that uses graph mining and advanced statistics. Significant highlights and the main contributions of the approach are:

- Through banking transactions as a network, graph mining methods, including the Parallelized greedy community detection algorithm, Louvain community detection, and modularity optimization algorithm, are applied to identify suspicious clusters and patterns of abnormal activities.
- We combine Benford's Law with transaction attributes to identify possible manipulation or fraud, utilizing the non-uniform distribution of leading digits in natural sets of data.
- The proposed system was thoroughly tested using the associated research through industry-standard IBM Transactions for AML, which simulates specific fraudulent situations.
- Graph mining, combined with powerful statistical tools, has enabled us to develop a system of fraud detection that is significantly more effective than traditional rule-based and machine-learning systems, which can adapt to the evolving strategies of fraudsters.

The study adds to the current activities of combating financial fraud that poses a high risk of loss of finances and image to the banking industry. In the following part we examine the methodology that is currently in use, discussing the limitations of the existing methods and a new method of detecting fraud. We are speaking about data preparation, graph construction, Louvain modularity, and application of Benford Law with the help of experimental outcomes that demonstrate the efficiency of our algorithm and its prospects of the industry.

2 Literature review

The evolution of fraud detection dates back long ago, but nowadays, it has changed significantly with the emergence of the modern digital financial systems. Fraudsters have enhanced various methods, and they are now taking advantage of the flaws in the virtual world using sophisticated technologies. The primary objective is to detect potential fraud in real-time or through retrospective analysis, thereby minimizing financial losses and preserving the integrity of the financial system. Fraud detection is a process that utilizes data mining methods to identify any abnormal behavior in data that may be indicative of fraudulent activity. The team utilizes specific software to enter the company's data and identify anomalies that may suggest the possibility of fraud and the individuals responsible [3]. System development has come up with various tools used to deter fraud over the decades. The first solutions were rule-based, which means that they used predetermined regulations to determine suspicious transactions. These systems, however, failed to keep up with changing fraud schemes and tended to give a false alarm. The transition to machine-learning and artificial-intelligence schemes, based on graph mining and sophisticated statistics, has enabled large amounts of data to be processed, complex trends discovered, and accuracy improved continuously over time. The emergence of more advanced systems based on machine learning (ML) and artificial intelligence (AI) has been driven by the evolution of fraud detection, incorporating graph mining and more advanced statistical techniques. Such systems can examine large volumes of data, identify more intricate patterns, and learn from new data to improve accuracy. Most of the current fraud-detection systems are now based on neural networks, decision trees, and ensemble models. The distributed denial-of-service (DDoS) family was introduced by Ziming and colleagues [4] to address new attack patterns. Through the characterization of traffic, creation of attack fingerprints, and family segmentation, large tests demonstrated the similarities, which steered classification, identified attacks unknown to them and refined defense measures. In their work, 89 real-world types of DDoS traffic were gathered, and 18 new attacks were tested, which showed that the division of families enhances the analysis and defence of DDoS. A graph neural network (GNN) is used in fraud detection to build neighbor relationships, which detect suspicious nodes and fraudsters, but in some cases, clever fraudsters may evade detection by being legitimate users.

Hence, the authors [5] emphasize relational density theory in this paper and propose a hierarchical attention-based graph neural network (HA-GNN). It uses weighted adjacent matrices across different relations to detect those fraudsters. This theory is based on the relation attention module, which highlights the strength between two nodes,

and the neighborhood attention module, which captures the long-range structural affinity within the graph. They generate node embeddings by collecting information from the node's long-range structures and its original node features. The main drawback of this research is that it cannot detect historical behavior information; it can only detect active fraudsters. Another drawback is that it cannot deal with the increasing volume of real-world data. The authors [6] cite the relationship between suppliers and customers to highlight the company's transaction activities. This relationship also reflects the relationship between various financial data disclosed in the companies' financial statements, which aids in detecting fraudulent financial statements. The conversion of supplier-customer knowledge to graph structure data for the GNN model input using an adjacency matrix is inefficient, which is a drawback.

The research focused on fusing heterogeneous threat intelligence to reconstruct multi-step attack scenarios and discover critical attack paths [7-8]. The researchers developed a threat intelligence fusion methodology based on structured threat information expression (STIX) and causal relationship analysis, a method of reconstruction, otherwise represented as a community discovery problem. A benchmark dataset based on real-traffic evaluation showed that the approach can recover from multi-step attacks and covert channels. Important contributions lie in the modeling of threat intelligence and the application of social network analysis. The inability to use replaceable words, however, meant that it was hard to model the attack scenarios. The authors referenced the telecom industry in [9] for the detection of fraud. To resolve heterophily issues in collaborative networks, they trained a multi-frequency graph neural network that can dynamically combine the attributes of node neighbors of different frequencies. The limitation of the study is that it has not utilized multimodal data, which includes speech, video, and text, to enhance the accurate detection of fraud. Consequently, there is a lack of accuracy in the research. In [10], an original light gradient boosting machine (LGBM) model was proposed to detect fraudsters in Ethereum and Bitcoin cryptocurrency transactions effectively. To classify cryptocurrency fraud, the authors combined different techniques, including random forest (RF) and multilayer perceptron (MLP). One of the flaws in this study was that most of the characteristics were removed from the base dataset, and it is often questioned whether this will affect the accuracy of the results. However, to its disadvantage, the paper lacks discussion on pattern discovery in fraudulent transactions involving cryptocurrencies, and the inconsistency arises from the disappearance of results using data sets with different lengths. Xiaobing et al. [11] proposed a method based on large-scale blockchain data to forecast fraudulent computer addresses and identify abnormalities in crypto transactions. These authors have employed various graph mining methods and utilized Benford Law to obtain distributional information on address transactions with a reasonably acceptable degree of accuracy. To compute the results, decision tree (DT), support vector machines (SVM), RF, LGBM, and logistic regression (LR) were used. However, the research's disadvantage was that the metric is limited, implying that a superior metric better suited to the Benford Law might have enhanced the research.

The area of banking makes use of advanced technologies and algorithms to scan through the patterns of transactions, detect anomalies and trends. The primary objective is to either detect fraud on-the-fly or by hindsight and minimize losses and protect the integrity of the financial system. The discrepancies in the data quantities have been observed to create a discrepant output. The authors suggest implementing a robust measure for comparing small and large samples to Benford law, which plays a crucial role in enhancing the system's applicability [12]. In Shigang et al. [13], the authors attempted to address the issues associated with using traditional data sources to detect fraud, which contain less information on financial statements. The most common classification methods used in the research include support vector machines, random forests, K-nearest neighbors (KNN), and DT; the research could have employed ensemble learning methods to enhance classification performance. The scholars [14] applied a knowledge graph to detect financial fraud in relative party transactions (RPTs). They focused on the total number of RPTs and loan-based RPTs, which received more attention. However, this approach has a drawback: only a portion of the nodes in the knowledge network with both dynamic and static links are visible, leading to an intense final visualization due to the high number of static interactions, which results in inaccurate findings.

The authors Pourhabibi et al. [15] highlight significant challenges that require further work to enhance the effectiveness of graph-based anomaly detection systems. Additionally, they have offered suggestions for resolving these issues. However, [16] has not thoroughly examined the various strategies used. They only used a few materials, primarily academic ones, and analyzing non-academic sources is still essential to examine the implementation issues that new graph solutions confront and the latest changes in the financial crime sector. They have, however, not supplied any answers for unusual use scenarios. They have made suggestions to increase the effectiveness of fraud detection; nonetheless, as with any system development, comprehensive evaluations of every implementation are missing. The scenario of detecting fraud, such as credit card fraud, involves quick decision-making where many transactions are involved. However, as Ganguli et al. [17] emphasized in their study, fraud relevant to credit card fraud must be understood. It has been found that deviations from Benford Law can indicate fraudulent actions, thereby offering a statistical basis for detecting fraud. This Law has been applied to several situations to identify anomalies in datasets that may indicate fraudulent behavior. The Louvain algorithm, developed jointly by Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre in 2008, is a community detection algorithm that utilizes

network analysis to identify clusters or communities within a graph [18]. The study, Wang et al. [19], presented a process by which they applied the fraud detection approach in the supply chain using the Benford distribution, assisted by an Excel sheet. Benford's Law was used to provide insight into different supply chain situations, and the author was able to infer patterns when carrying out calculations. Nonetheless, the weakness of the research was realized through its dependence on information from a single source. The results may be subject to change, which could be achieved by incorporating more diverse data sources. The output of the paper can be improved by the addition of more sophisticated processes to detect red flags of fraudulent practices, including the non-clarity of job specifications, vendor selection without capability evaluation, and incorrect inputs or incomplete information on the negotiation sheet, which points to the vendor's capabilities. Further methodological improvements may lead to more robust outcomes. The authors in Bernaschi et al. [20] have demonstrated an innovative parallel code for the BFS algorithm, enabling the efficient study of large-scale graphs. They claim that their concept of implementation can be applied to a platform that provides many levels of parallelism. It is also worth mentioning that they have been using new-generation Nvidia GPUs, and some of their optimizations are CUDA-specific [21].

Table 1: Comparative study of state-of-the-art fraud detection techniques in banking transactions

Technique	Dataset Used	Metric Value	Key Strengths	Limitations	Ref.
Intuitionistic Fuzzy Logic	Synthetic Bank Transactional Data	A: 98.00 P: 81.00 R: 93.00 F1: 86.00	Offers a unique approach based on fuzzy logic for fraud detection	Specificity and adaptability to different fraud patterns need to be carefully considered	[22-23]
Rule-based Systems	IBM, Financial Transactions	A: 91.48 P: 91.21 R: 91.20 F1: 91.35	More adaptable than rule-based systems, it handles complex data Reliable AI model for detecting fraudulent bank transactions Easy to implement, interpretable results	The complexity of fraud detection and imbalanced data may pose challenges Black-box nature requires significant training data Limited adaptability to evolving fraud patterns, prone to false positives/negatives Can be time-consuming and inaccurate	[1], [3], [24], [27]
Decision Tree	China Stock Market Trading Research	A: 72.21 P: 52.24 R: 56.49 F1: 47.91	Effective when combined with a bagging ensemble for credit card fraud detection	It may not be the optimal algorithm in all cases	[13, 14]
Logistic Model Tree	CBank, Czech Financials	F1: 91.00	Shows promising results in fraud classification and detection	Performance may vary depending on the dataset and specific fraud scenarios	[11]
Random Forest	China Stock Market Trading Research	A: 84.11 P: 54.35 R: 59.41 F1: 54.04	Outperforms SVMs, logistic regression, and KNN	It may not be the best choice in all scenarios	[13]
XGBoost	China Stock Market Trading Research	A: 67.36 A-I: 69.07 A-II: 67.27 AUC: 75.36	Effective in many circumstances	Not advantageous in all situations	[14]
Support Vector Machine	China Stock Market Trading Research	A: 93.18 P: 68.45 R: 56.11 F1: 58.52	Effective for high-dimensional data, good generalization ability	Computational cost, sensitive to parameter tuning	[13, 25]
Deep Learning SqueezeNet	PhishStorm - phishing / legitimate	A: 93.05 P: 94.26 R: 93.75	Highly effective for complex patterns, learns from large datasets	High computational cost, prone to overfitting if the data is limited	[26]
Hybrid CNN–LSTM	Custom Bank Transaction Dataset	A: 94.8	Utilizes a hybrid CNN–LSTM for sequential transaction modeling and improved feature learning	High computational cost and limited interpretability for large-scale deployment	[29]
Ensemble Learning	Kaggle Credit Card Dataset	P: 93.5 R: 91.8	Aggregates Random Forest, XGBoost, and SVM for balanced classification	Requires hyperparameter tuning; performance drops on unseen transaction distributions	[30]
Adaptive ML Pipeline	(European Credit Card Data	F1: 92.3	Introduced adaptive resampling and model retraining for dynamic fraud trends	Limited graph-level understanding; temporal correlations not modeled	[31]
AI-Driven Deep Neural Ranking	European AML Case Data	AUC: 0.95	Integrates rule-based filtering with deep neural ranking for risk prioritization	Rule-based dependency restricts generalization to new transaction types	[32]
Hybrid ML and Deep Graph Model	Real-World Transaction Network Dataset	A: 95.6	Combines deep graph representation learning with supervised fraud classification	Requires extensive labeled data and a large memory footprint	[33]
Temporal Graph-Based Networks	Temporal Banking Dataset	AUC: 0.93	Captures evolving transaction dependencies using temporal graph embeddings	Computationally demanding; limited in real-time adaptability	[34]

A detailed comparison of various fraud detection methods in banking transactions is presented in Table 1. This comparison highlights the virtues and shortcomings of both methods, providing valuable insights for any financial institution seeking to enhance its fraud detection framework. Rule systems make them interpretable, albeit with a high rate of false positives, and they must be updated regularly to accommodate changes in fraud schemes. Algorithms in machine learning, such as support vector machines and deep learning, are flexible but may be challenging to apply in practice. They are non-explanatory; thus, it would be hard to know the rationale

behind their choices. Detecting an anomaly in the statistics offers certain flexibility, although it is limited in identifying complicated fraudulent schemes, and the explanation is not as straightforward as it can be with regression logic. Decision trees and random forests, which are algorithms in data mining, provide valuable information, but they may not be effective in capturing the dynamics of financial fraud. Through a comparison of these techniques, the study suggests the possibility of exploring the proposed approach to enhance the ability to detect fraud and protect the banking sector from the constantly evolving threats, with financial institutions facing the challenge of dealing with sophisticated fraudsters, applications that integrate sophisticated analytics and flexibility become increasingly important in maintaining the robustness of the financial system and safeguarding consumer integrity.

According to Pham et al. [25], a system for anomaly detection in cryptocurrency transaction networks has been developed. They applied three unsupervised machine-learning techniques on two distinct graphs: unsupervised support vector machines, Mahalanobis distance, and k-means clustering. This research's drawback was that, although they provided parallelized computation for quicker detection, doing so necessitates significant work, such as adding thread safety to learning algorithms, which is not covered in this study. They have not thoroughly reviewed the 38 million transaction dataset. This paper [27] reviews the classification scheme for applying data mining techniques to detect financial fraud, demonstrating the relationship between data mining techniques and different types of financial fraud. It provides an overview of techniques for financial fraud detection. However, the drawback of this paper is that it has not developed a specific system for this purpose and has not provided any relevant research on detecting outliers for those frauds, resulting in a lack of research data.

The detection is based on data-mining techniques through which data volumes are sifted to identify any abnormal activity that may indicate fraud. Specialized software is used by teams to load company data, mark anomalies, and trace the people behind suspicious activity. The area of activity gained thrust in the late 20th century, coinciding with the emergence of large-scale network datasets. Researchers have begun searching for algorithms and methods to efficiently mine these graphs, with their applications being practical in various fields, including social network analysis, bioinformatics, and finance. Graph models and mining techniques have played a crucial role in identifying evolving behavior, particularly in the corporate fraud area [2]. Within the framework of fraud detection, the Louvain algorithm is tested to identify patterns and associations related to transaction data in the banking sphere. More easily, anomalies or outliers can be identified by establishing communities or a wealth of transactions that exhibit similar behavior. This model enhances the capacity of fraud detection systems to identify irregularities that cannot be detected using conventional analysis methods. The Louvain algorithm is quite adept at identifying dense substructures or clusters in graphs, finding frequent substructures, and providing an interactive visualization of mined knowledge [28]. Algorithms such as Louvain can help researchers and professionals easily identify trends indicative of fraudulent behavior within large datasets. The use of algorithms, such as the one created by Louvain, and principles, such as the Law of Benford, has enhanced the effectiveness and accuracy of fraud detection. Recent literature has resorted to graph mining to uncover fraud through analysis of relationships between entities of information. Graph mining has emerged as an effective technique to detect complex relationships particularly in large-scale network data. These graphs have become a target of research by researchers in such areas as social network analysis, bioinformatics and finance. Graph models and mining methods have been found to be critical in the identification of changing behavior, especially in corporate fraud. The laws of Benford have been applied, and algorithms such as the Louvain algorithm have been implemented to aid in uncovering patterns. All these methodologies contribute to enhancing fraud detection capabilities across a broad spectrum of areas, including corporate fraud and financial transactions. With these systems, as technology continues to advance, the continuous development of the systems is essential to keep threats of ever-changing fraud in the financial arena at bay.

3 Material and methods

3.1. Preliminary definitions and dataset analysis

In this section, we describe the main concepts, and we have a thorough analysis of our dataset. We establish fundamental aspects, such as transactions, graphs, sub-graphs and the Law of Benford which form the basis of our study. We also measure the parallelism of the dataset, conduct statistical analysis of the chosen communities, and give the visual summaries to gain further insight. This detailed introduction forms a good background of the methodology and further sections.

Transaction: The concept of a transaction is the most vital in the world of banking. All transactions of money are registered as transactions having a distinct ID that can be verified in subsequent payment. A set of tasks represents transactions, which start with one task and cease only after all the tasks in the set are done. Moreover, there are legitimate transactions and illegitimate transactions depending on some factors. This is why one must be careful and attentive when undertaking transactions to make them genuine [21].

Graph: This is a type of data structure which employs the use of vertices and edges. The basic building blocks of graphs are nodes also called edges. They are entities, data points, or objects and are intended to hold more information about these respective entities or attributes.

The links or the connections between the nodes/vertices in the graph can be referred to as the edges. There are directed and undirected edges in case they possess an orientation or not [29]. There are further subdivisions in graphs; directed graphs, undirected graphs, weighted graphs, cyclic graphs, acyclic graphs, connected graphs, disconnected graphs and trees.

Subgraph: Subgraph In graph theory, a subgraph is a smaller graph obtained by an encompassing parent graph that has edges and vertices. The importance of subgraphs is hard to underestimate, and they are crucial to the activity and analysis of graphs by giving us the opportunity to operate within a narrower, more extensive section of the bigger graph [36]. In addition, the methodology will be used to maintain the original structural and connectivity properties with the benefit of studying it better and more effectively. They have many subgraphs such as Induced Subgraphs, Spanning Subgraphs, Connected Subgraphs, and Maximal Subgraphs and each has its peculiarity and importance.

Benford Law: It is a very strong instrument which assists us to consider the validity of statistics and financial records more accurately. Thus, it is necessary to remember Benford Law when examining random collections of numbers and take relevant and correct decisions [37]. It is a rule we must know provided we are to act with large masses of haphazard natural numbers, like the sales records of a corporation, or the look-up tables of logarithms. According to it, about 30 percent commences with digits 1, 18 percent with 2 etc., the small percent with 9. The frequencies among the first non-zero digits in the datasets are usually not distributed evenly across populations. Rather, it follows a log-uniform distribution, except that digit 1 is the most common, then 2, then 3, etc., through 9. This is what is referred to as Benford Law. The Benford Law shows that the likelihood of a sample having a significant digit, d can be determined as illustrated in Eq. (1).

$$P(d) = \log_{10}(1 + \frac{1}{d}), \mathrm{d} = 1, 2, 3, \ldots, 9 \quad (1)$$

Parallelism divides the information so that it can be easier to be processed by the GPU and leave sufficient memory to the other data. Parallelization may be done in two forms, one is by parallelizing the GPUs, the other is parallelizing the CPU. The main factor to be considered when choosing between GPUs or CPUs to use in an application is the fact that GPUs are much more suitable in case of parallel processing. The known fact is that CPUs have a small extent of parallelism, which implies that not all parallelization in workloads is designed equally. Thus, it is essential to consider the needs of the application cautiously and decide whether it is the GPUs or the CPUs, which should be given priority to achieve the best performance [38].

Dataset: A financial dataset is usually structured information that pertains to financial transactions, financial institutions, financial markets and other pertinent entities. These data sets are important in quantitative analysis, risk analysis, some financial modeling exercises, and management of the portfolio. The dataset used in our research is the information and data that are of different types and varying sizes, which have small, medium, and large volumes. The basic dataset employed in this study as shown in Figure 2 is IBM Transactions for AML dataset. This extensive body of data, generated by the simulator proposed by Suzumura and Kanezashi [39], includes approximately 7 million transactions presented in a tabular format. Specifically, the implementation utilizes two main files from this set: the LI-Small_Trans.csv file, which is a substantial file containing the primary transaction records, and the accompanying LI-Small_Patterns.txt file, which is crucial for our work as it contains predefined, labeled transactions that exhibit known money laundering patterns, Erik A., [40]. Further structures can be time series or hierarchical data, or others may be even more complex [41]. The following five communities and their temporal analysis represent the duration throughout which transactions are carried out in specific communities. Here are the durations of the top five communities.

1. Community 66 spans from 2022-09-01 00:00:00 to 2022-09-14 13:48:00
2. Community 123 spans from 2022-09-01 00:00:00 to 2022-09-14 14:10:00
3. Community 31 spans from 2022-09-01 00:00:00 to 2022-09-11 13:28:00
4. Community 9 spans from 2022-09-01 00:00:00 to 2022-09-10 23:59:00
5. Community 50 spans from 2022-09-01 00:00:00 to 2022-09-10 23:59:00

Table 2: Dataset statistics analysis of the selected communities for research

Community sequence number	Total nodes	High-degree nodes with %	Low-degree nodes with %	The average degree of high-degree nodes	The average degree of low-degree nodes
66	31351	4040 (12.89%)	27311 (87.11%)	9.31	2.32
123	22744	3068 (13.49%)	19676 (86.51%)	9.38	2.33
31	19725	3219 (16.32%)	16506 (83.68%)	5.29	1.76
9	19352	2415 (12.48%)	16937 (87.52%)	9.06	2.43
50	18541	2442 (13.17%)	16099 (86.83%)	9.39	2.28

The statistical analysis of the selected communities is presented in Table 2 above. After calculating the total number of nodes, we sorted the nodes into two categories: high-degree thresholds and low-degree thresholds. Then, the average degree threshold of both categories is enumerated. The degree threshold represents the number of connections a particular node has. This helps plot the

community subgraph. The high-degree threshold denotes comparatively higher connections between nodes, while the Low-degree threshold denotes comparatively lower connections between nodes. The average degree of high-degree nodes represents the average degree threshold for high-degree nodes, and the average degree of low-degree nodes represents the average degree threshold. Figure 3 shows clearly the top five communities as determined by the framework, giving a visual representation on the number of nodes in each community. What is interesting about such communities is that this is indeed true since they indeed possess graphical networks of possibly fraudulent transactions, which is marked by the anomaly detection facilities in the framework. The bar plot is an easy and informative visual representation of the relative frequency of these anomalous communities, which will help the analyst to evaluate the extent and the importance of suspicious activities, which have been detected within a short time.

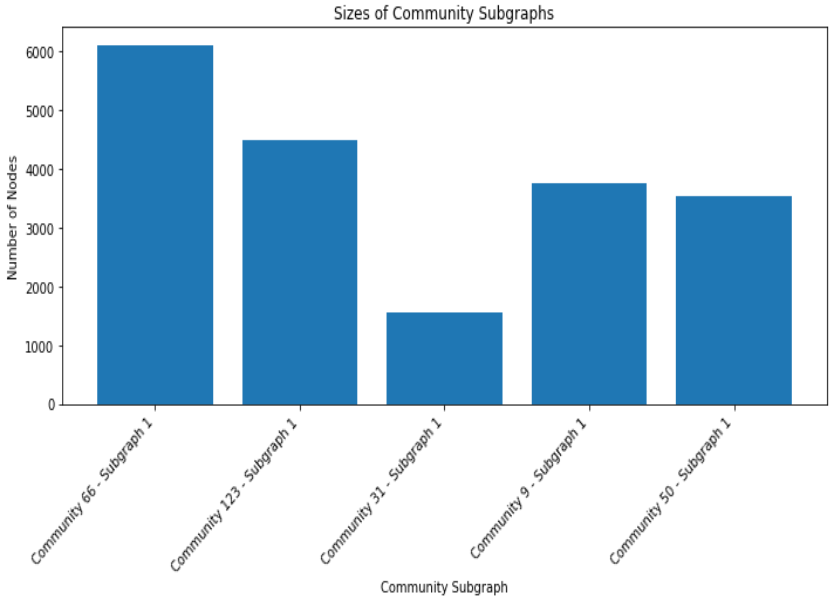


Figure 3. The top five communities were retrieved with the help of the threshold

The given tool offers an intuitive and natural way to interpret the results and makes it even more significant to study the issue outside of the given framework. As the financial network develops, it can be interesting to watch how it progresses and how the community's shift. A partial visualization could be achieved by tracing the number of new edges (connections) into the network against the expansion of the closest detected communities. They are reflected in the graphical representation in Figure 4, where the x-axis represents the duration of analysis, and the y-axis depicts the number of new edges and the number of communities. The number of new edges is gradually increasing over time, indicating that the network is expanding at some point. At the same time, the number of communities might also increase, which will be a manifestation of the formation of new groups of related organizations. With this graph, we can gain a deeper understanding of the network's evolving form, including periods of rapid growth and consolidation, and identify the first signs of unusual activity that may warrant further investigation.

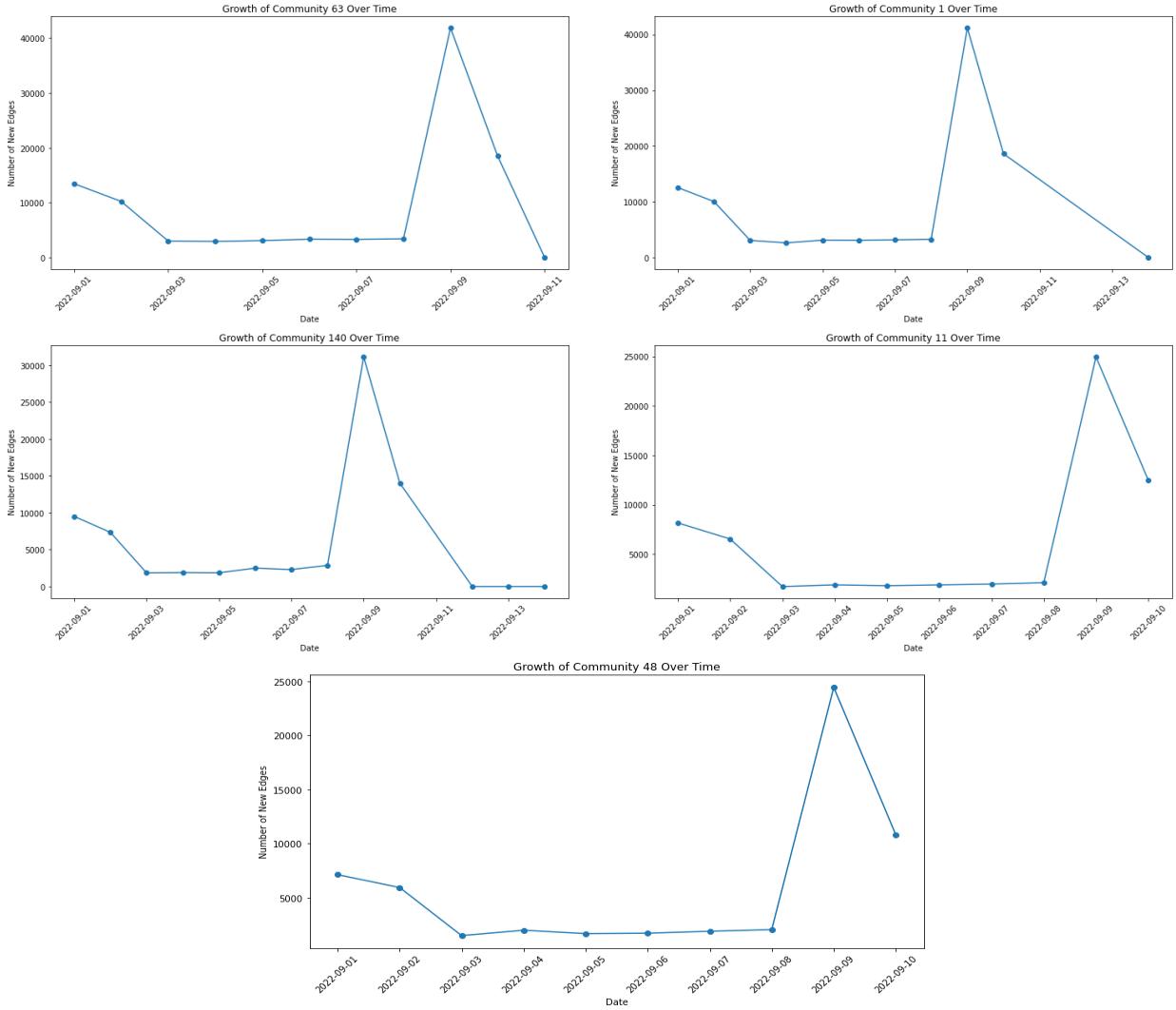


Figure 4: Visualizing the growth of the nearest identified communities in the financial network

3.2. Proposed architecture

Figure 5 refers to the suggested architecture of an anomalous transaction detection system, found on transactional database and graph analysis with community classification. The system attempts to identify suspicious transactions based on anomalies in the frequency and quantity of transactions. The proposed scheme is generally broad-based in detecting suspicious activities in financial networks. The heart of this system is a strong database which contains specific details about the account holders and their financial transactions including some vital details like the number of transactions. It is detailed data that is used to analyze further. Everyday transactions are made within the banking systems. One should first graph the system, whereby each account is represented as a node, and each transaction is represented as an edge before the financial data can be used. Other studies have explained chi-square values, edge weights, and modularity [1, 24].

The system is also optimized and parallelized to obtain efficient results. The model takes advantage of the AntiBenford subgraph, which is an element that is new and takes the form of a graph of the transactional data. The resulting transaction network is a representation of all the nodes as individual objects involved in the transaction (a merchant, an intermediary or an account), and all the edges reflect the transactional relationship of two objects. Value of the transaction is the edge value, and the network monitors the direction and magnitude of the financial flow. More to the point, the AntiBenford subgraph model employs Benford Law which is a statistical model of the natural distribution of leading digits to detect aberrant subgraphs in the larger financial network. These subgraphs will be potential hotspots where frauds can take place, or suspicious activities may take place. The abnormalities with reference to the Benford Law are identified under the methodology where the transactions with a distribution that is significantly other than the expected distribution are considered and this means there are abnormalities that need further study. The flagged transactions arrangement is of subgraphs as a form of interrelationship and context of the transactions.

When the statistical AntiBenford analysis is merged with the Louvain community detection algorithm, the abnormalities can be identified at both the micro and macro scales. Statistical deviation initially illustrates abnormal transaction patterns, and community detection groups the anomalies into solid clusters, exposing the possible legalized networks that would not be evident when analyzing them separately. The model uses the following methodology to determine the top five communities as potentially fraudulent transaction node, based on the frequency and amount of transaction.

1. The transactions dataset is provided as input to the AntiBenford Subgraph framework, through which an edge list is generated that potentially identifies fraudulent transactional nodes.
2. The calculation of the AntiBenford Subgraph framework works as follows:
   - A chi-square test (a test of independence) can be used to determine whether the observed frequencies differ significantly from those expected if handedness is unrelated to nationality. A chi-square test is notably used in hypothesis testing.
   - The x_s, d is the number of edges in the subgraph, which is introduced by S, whose weight first digit is d; further, by applying the formula, we calculate the edge weight by a formula $E = \sqrt{(x_a^2 \times x_b^2)}$
   - where a and b are the two nodes multiplied by the first digit, the overall edge list is generated as input to Louvain's algorithm.
3. In Louvain's algorithms, the modularity of the edge list is computed, resulting in the generation of a threshold.
4. With a threshold, communities consisting of fraudulent transactional nodes are being created.
5. The user can set a fixed threshold according to their needs or have it analyzed dynamically through the edge list.
6. Modularity Calculation works as follows:
   - The modularity of a network partition can be computed using Equation (2).

$$Q = \frac{1}{2m} \sum_{ij} \left(W_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j) \qquad (2)$$

   - Where,

W_{ij} is the weight of the edge between nodes *i* and *j*.
d_i *and* d_j are the degrees of the nodes *i* and *j*, respectively, the sum of the weights of edges incident to nodes i and j.
m is the total weight of all edges in the network.
c_i *and* c_j are the communities to which the nodes *i* and *j* belong.
$\delta(c_i, c_j)$ is the Kronecker delta function, which equals 1 if nodes i and j belong to the same community, and 0 otherwise.

   - Modularity is a measure of the quality of partitioning a network into communities. It quantifies the extent to which the division of nodes into communities is better than what would be expected by random chance. A higher modularity value indicates a better community structure, where nodes within communities are densely connected, and there are fewer connections between communities.
7. The framework generates the top five communities as output, which consist of potentially fraudulent transaction nodes based on their frequency and amount. This targeted approach helps identify high-risk activities for further investigation.

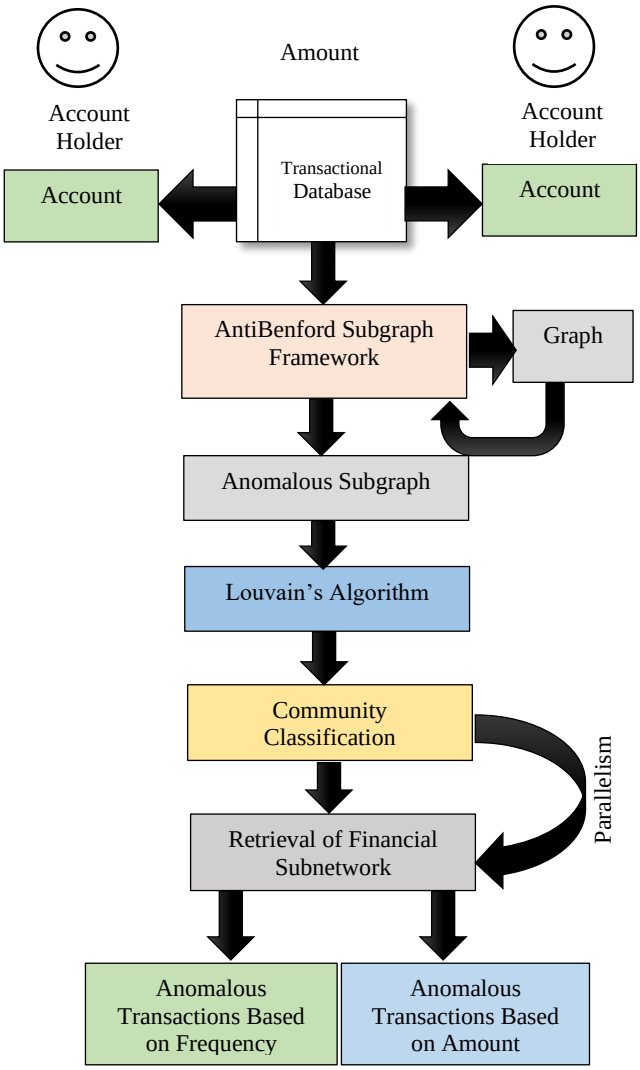


Figure 5: Proposed architecture for an anomalous transaction detection system

In the AntiBenford subgraph, the weight of the edge between two transaction nodes is further drawn by the first digit of the transaction amount. The heuristics are inserted to enhance the strength of transactions that do not fit within the anticipated statistics, especially those stipulated

by the Benford Law. The Law of Benford defines a logarithmic distribution where the smaller numbers (i.e., the ones and twos) are observed more frequently, and the higher the number, the more likely it is to be the first in naturally occurring financial data. Fraudulent activity, however, will generally indicate abnormal frequencies of digits, often with larger leading digits, which violate this law since they are artificial. When we sort the edges by the leading digit, we give an edge to the anomalous transactions in the graph. More odd central digits (such as 8 or 9) are assigned relatively more weight and are thus brought into the center of focus in the Louvain modularity optimization.

This will encourage such deals to be concentrated around the detected suspicious communities. Finally, this approach will render the framework more accommodative of the fraud conduct through coding the statistical anomalies in the graph format, which will be more precisely identified at the community level. The Louvain algorithm can be used with the framework to perform community classification, which shows the structure of a financial network. This operation brings together the closely related ones and brings out the different communities or groups in the network. Louvain Algorithm gives an upper hand to the analysts who can extract useful information on the financial network. The algorithm is helpful because it helps analysts to find the most interesting subgraphs, the abnormal subgraphs, and the anomalies of the node behavior. Algorithms 1 and 2 have been used as a preprocessing step to enhance the speed of convergence and accuracy of modularity of algorithms respectively. Such a narrow-down method is beneficial to perform an efficient and effective analysis, and, thus, potential fraud cases may be identified with the help of such a complex system of bank transactions. The algorithm determines the types of nodes into communities based on their interconnections and gives data regarding the form of the aberrant networks and offers insights into the communal behaviors in transactions. This analysis helps to perceive modularity of the dataset and background patterns of ties that define noted abnormalities. The framework makes use of parallel processing to access the financial subnetworks of the identified unusual communities in the most efficient way imaginable thus maximizing the financial analysis of the network. The given approach allows analyzing the most topical data, as opposed to the search in the whole network.

Algorithm 1: Parallelized Greedy Community Detection Algorithm	
Input	A network G(V, E) with a set of nodes V and a set of edges E.
Output	A community assignment for each node in V.
Step-1	Parallel Node Partitioning Divide V into k disjoint subgraphs. $V_1, V_2, V_3, \ldots \ldots V_k$, where k is the number of available GPU cores. Round-robin partitioning: Assign nodes sequentially to subgraphs in a circular fashion. Graph partitioning algorithms: METIS or Scotch for more sophisticated partitioning based on graph properties (e.g., edge weight distribution).
Step-2	Parallel Modularity Calculation (Within Subgraphs) For each subgraph V_i (in parallel across GPU cores): Initialize an empty community assignment C_i for nodes in V_i. For each node v in V_i: Calculate the modularity change (ΔQ) resulting from moving v to each possible community. Efficiently store these ΔQ values (e.g., using a data structure like a hash table).
Step-3	Synchronizations and Communication Aggregate modularity changes: Gather the ΔQ values computed across all subgraphs. This involves MPI communication primitive mechanisms depending on the parallel programming environment.
Step-4	Parallel Node Movement (Within Subgraphs) For each subgraph V_i (in parallel across GPU cores): Identify the node v_i in V_i with the maximum positive ΔQ (or minimum negative ΔQ if maximizing modularity is not the goal). If such a node exists, move v_i to the community that yields the maximum ΔQ. Update the community assignment C_i accordingly.
Step-5	Iteration and Termination Repeat steps 2-4 until a stopping criterion is met (e.g., no further node movements improve modularity, a maximum number of iterations is reached, or a convergence threshold is satisfied).

Beginning with the step of attentive collection and processing of different transaction information that encompasses a substantial volume of data concerning the accounts in which transaction takes place, and league of transaction metadata that embodies the comprehensive examination of financial activities. It is the stage in which a deep analysis is performed, and the richness and integrity of data are guaranteed. The proposed structure will be a holistic solution to anomaly transactions and possible financial fraud detection in the banking network through the harmonious composition of these key components. An increase in the modularity of identifiable

communities that is greatest would reveal homogenous communities of aberrant transactions to financial institutions, enabling them to uncover and remove risks prior to their occurrence. This is a multifaceted solution that gives financial institutions an upper hand against evolving fraudulent methods and ensures their financial systems remain secure. The computations of the greedy phase can be split across multiple cores of a GPU to parallelize the Louvain algorithm on a GPU. The cores are capable of computations of a specific proportion of the nodes and communities adjacent to them, enabling them to run multiple tasks simultaneously and complete them much faster.

Algorithm 2: Louvain Community Detection and Modularity Optimization Algorithm	
Input	A network G(V, E) with a set of nodes V and a set of edges E.
Output	A community assignment for each node in V.
Step-1	Initialization (Community Seeding): For each node v in V, create a unique community C_v containing only v. Set the current modularity (Q) to 0.
Step-2	Modularity Optimization (Local Improvement): While there exists a node v in V: For each neighboring community C' of v: Calculate the change in modularity (ΔQ) that would occur if v is moved to C'. If ΔQ > 0: Move v to C'. Update the modularity Q with ΔQ. If no node relocation leads to a positive ΔQ, exit the loop.
Step-3	Community Aggregation (Hierarchical Construction): Create a new network G'(C, E') where: Nodes in C' represent the communities identified in step 2. An edge exists between communities C_i and C_j in G' if there is at least one edge connecting a node in C_i to a node in C_j in the original network G. The weight of the edge in G' reflects the number of connections between the corresponding communities in G.
Step-4	Iterative Refinement: Apply steps 2 and 3 to the newly created network G' (C, E'). Repeat this process until there is no further improvement in modularity (ΔQ ≤ threshold) for a predefined number of iterations.
Step-5	Termination: The algorithm stops when the modularity change (ΔQ) falls below a predefined threshold for a set number of iterations, indicating a stable community structure.

The framework divides the graph into smaller subgraphs, with each subgraph assigned to a separate core of the GPUs to speed up the modularity optimization scheme. Such that this splitting can be based on a round-robin approach, or force graph partitioning algorithms such as MeTiS or Scotch to provide an effective workload placement. Individual GPU cores, possibly in parallel, then compute the change in modularity caused by the movement of nodes between communities within the subgraph assigned to it. To further parallelize this process, the framework scales the calculation of modularity contributions for various nodes by utilizing more than one core of a GPU simultaneously. A framework that coordinates the node movements of all cores takes, at the end of each iteration, the modularity changes made by each core and synchronize them with the movements. With this parallelization scheme, the framework can optimize modularity at a significantly higher rate (by far), which drastically lowers the amount of time one needs to calculate to detect the presence of anomalous communities in the financial transaction network. The cores can communicate with each other to share information on surrounding communities and changes in modularity. Table 3 depicts the algorithms employed in this research, with the pseudocode outlining the AntiBenford subgraph extraction process, the Louvain community detection algorithm, and the follow-up modularity optimization.

Table 3: Pseudocode for AntiBenford subgraph extraction, Louvain community detection, and modularity optimization

AntiBenford Subgraph Extraction	Louvain Community Detection and Modularity Optimization

# AntiBenford Subgraph Extraction 	
# Input: Transaction dataset T 	
# Output: Set of AntiBenford subgraphs G_AB 	
 	
function AntiBenford_Subgraph_Extraction(T):
G = initialize_graph()
for t in T:
u = t.From_Account
v = t.To_Account
amount = t.Amount
digit = first_digit(amount)
 	
G.add_edge(u, v, weight = normalize(amount))
increment_frequency(digit)
 	
expected = {d: log10 (1 + 1/d) for d in range (1, 10)}
chi2_scores = {}
 	
for subgraph in partition_graph(G):
obs = count_first_digits(subgraph)
chi2 = sum(((obs[d] - expected[d])**2) / expected[d] for d in range(1, 10))
chi2_scores[subgraph] = chi2
 	
G_AB = {s for s, chi2 in chi2_scores.items() if chi2 > THRESHOLD_AB}
return G_AB
	# Louvain Community Detection and Modularity Optimization
	# Input: Graph G_AB
	# Output: Detected communities C
	
	function Louvain_Community_Detection(G_AB):
	communities = {v: v for v in G_AB.nodes}
	modularity_prev = -inf
	modularity_gain = inf
	while modularity_gain > EPSILON:
	for v in G_AB.nodes:
	current_comm = communities[v]
	best_comm = current_comm
	best_deltaQ = 0
	neighbor_comms = get_neighbor_communities(v, communities)
	for c in neighbor_comms:
	deltaQ = compute_modularity_gain(v, c, G_AB, communities)
	if deltaQ > best_deltaQ:
	best_deltaQ = deltaQ
	best_comm = c
	if best_comm != current_comm:
	move_node(v, best_comm, communities)
	G_coarse = aggregate_communities(G_AB, communities)
	modularity_new = compute_modularity(G_coarse)
	modularity_gain = abs(modularity_new - modularity_prev)
	modularity_prev = modularity_new
	G_AB = G_coarse
	C = extract_final_communities(communities)
	return C

When applied to large-scale networks, the efficiency of the Louvain algorithm can be significantly improved by adding parallelism. The Louvain Algorithm would be an excellent option for analyzing large networks consisting of millions of nodes and edges. Its usefulness is particularly evident in dynamic network analysis, which smoothly adjusts itself to the changing environment of communities as time progresses. Furthermore, it is highly successful in identifying overlapping communities, indicating its ability to identify nodes that belong simultaneously to more than one community cluster in this case. It applies the concept of modularity optimization, a measure used to determine the quality of a community structure. The Louvain community detection and modularity optimization algorithm begins by initializing the assignment of each node to its own community. The Louvain algorithm takes each node as an independent network entity. Then, it proceeds to a modularity optimization stage, during which the nodes are repeatedly moved to adjacent communities. A node is, however, relocated only in the event of increasing the modularity of the network. It repeats itself until local changes in the allocation of nodes achieve no further increase in modularity. After the modularity optimization process is completed, the algorithm combines the discovered communities into a new network. Each node represents a community from the previous step, and the edges are adjusted in this new network to capture the relationships between communities. The algorithm proceeds to the iterative refinement step, and in this process, steps 2 and 3 are repeated on the newly constructed network. This process continues until further refinement of modularity can be achieved through the progressive optimization and combination of communities. Lastly, the algorithm terminates when successive iterations of the modularity fail to yield a substantial change in the modularity. Our iterative algorithm is a successful mechanism that divides the network into communities at a time, maximizing modularity. It means that the ideal community structure for the network has been discovered, and there is no point in refining it further. Ultimately, we obtain financial sub-networks with the most significant number of transactions, as well as those with high and low transaction frequencies. The specific method for calculating modularity (Q) and ΔQ depends on the chosen function (e.g., Newman-Girvan modularity). Different threshold values for ΔQ and the number of iterations were used to control the trade-off between computational efficiency and community quality. This algorithmic framework provides a modular and adaptable approach for identifying communities within a network.

4 Experimental evaluation and discussion

To realize the proposed model of fraud detection, the team installed powerful hardware architecture, which incorporated the performance of the best technology. The key element of the system is the Nvidia T1000 graphics card, which has a high-level Turing architecture. This graphic card has 896 cuDA cores, 8GB of high-speed GDDR6 memory, and a bandwidth of 160 GB/s. It has a single-precision performance of an astounding 2.5 TFLOPS, and it can process intricate data at a high rate. The PC will come with 32GB RAM and an Intel i7 processor to improve the functionality of the model, and this will offer adequate memory and computational

abilities to facilitate the operations of the model. Software-wise, the team developed under the Visual Studio Code with Python as the language. It is also suggested that the model used in identifying abnormal transactions in financial networks should be built on the ability of parallel computing to streamline its operations and efficiency. The parameters of Table 4 were chosen as a result of empirical testing and as per the established conventional wisdom in graph-based anomaly detection. The level of AntiBenford deviation, it was discovered, of 0.25 is a perfect measure of how to differentiate normal numeric and abnormal numeric distributions, without going too far to mark legitimate accounts. A convergence tolerance of 1×10^{-5} and a limit of 500 iterations for Louvain optimization ensured modularity stabilization without unnecessary computation. The coherent community sizes were obtained when the resolution parameter r = 1.0 was used and the transaction density was varied, which demonstrates its applicability to large financial networks. Eight worker threads, corresponding to the hardware capacity, provided almost linear speedup in the greedy community module. Combined with other values of these parameters, the obtainable combinations constitute a set of reliable, stable, reproducible, and computationally efficient parameter combinations for identifying suspicious communities in large-volume transaction graphs. During sensitivity analysis, we changed the modularity threshold, Q_{th} between 0.35 and 0.55 to determine the optimal means of detecting suspicious communities. We discovered that if we set the threshold less than 0.35, we got too many false positives due to the combination of loosely connected and non-suspicious nodes, and when we set the threshold more than 0.55, a genuine suspicious community would be divided into smaller parts, significantly reducing the recall. However, the threshold value of Q_{th} = 0.45 turned out to be the most efficient point; this one offers the optimal balance between detecting accuracy and avoiding the disintegration of the suspected suspicious communities detected.

This enables this framework to compute the hypothetical gain of modularity of any given node in parallel and significantly shortens the computation. This is reasonable because it will save much time particularly when the creation of the communities is postponed without the reassignment of the node. On the same note, the allocation of nodes to new communities has been parallelized and the framework takes care of dependencies between moves that may affect the same communities. Parallel processing has also been utilized to compute the weights between the newly created communities of the edges. The total weight of the weight of the edges of each pair of communities is independent and hence they can be computed concurrently and thus reduce more time in building the new, aggregated network. Architecture also demands parallelism in the creation of the new network per se. As the new nodes and their edges are not dependent on the others, the framework can generate the nodes (symbolizing the initial communities) and their relations in parallel facilitating the process of creating the network.

Table 4: Parameter settings used for AntiBenford analysis and community detection algorithms

Parameter	Description	Standard Value Used	Rationale in Implementation
AntiBenford Threshold	χ^2 deviation threshold used to flag statistically irregular subgraphs	0.25	Empirically tuned from validation to balance FP and missed anomalies; higher values suppress minor deviations while preserving actual laundering patterns.
Minimum Group Size	Minimum transactions required per node group for reliable digit frequency estimation	10	Ensures adequate statistical significance when computing Benford deviation for smaller account sets.
Edge Weight Normalization	Normalization factor applied to transaction amount for uniform graph scaling	[0, 1]	Keeps edge weights comparable across transactions, avoiding dominance of large-value transfers.
Resolution Parameter	Controls the granularity of community detection in the Louvain algorithm	1	Default resolution yielding balanced community sizes; empirically observed to maximize modularity without fragmentation.
Maximum Phase Iterations	Upper limit of refinement loops per Louvain phase	500	Provides sufficient convergence while preventing excessive re-computation in large graphs.
Convergence Threshold	Minimum change in modularity (ΔQ) required to continue optimization	1×10^{-5}	Standard tolerance ensures modularity stabilization and reproducible partitions.
Parallel Workers	Number of CPU/GPU threads used in the Parallelized Greedy module	8	Matches available logical cores on experimental hardware; maximizes throughput with minimal synchronization overhead.
Graph Neighborhood Radius	Defines hop distance for extracting AntiBenford subgraphs around flagged nodes	1	Restricts local subgraph expansion to immediate transactional neighbourhoods, emphasizing recent fund circulation loops.
Random Seed	Initialization value for reproducible random partitioning and sampling	42	Maintains experiment repeatability and identical modularity outcomes across runs.
Modularity Threshold	Defines the minimum modularity value Q_{th} for a community to be flagged as suspicious	0.45	Determined through sensitivity analysis, this ensures a balance between suppressing FP and maintaining community cohesiveness in fraud pattern detection.

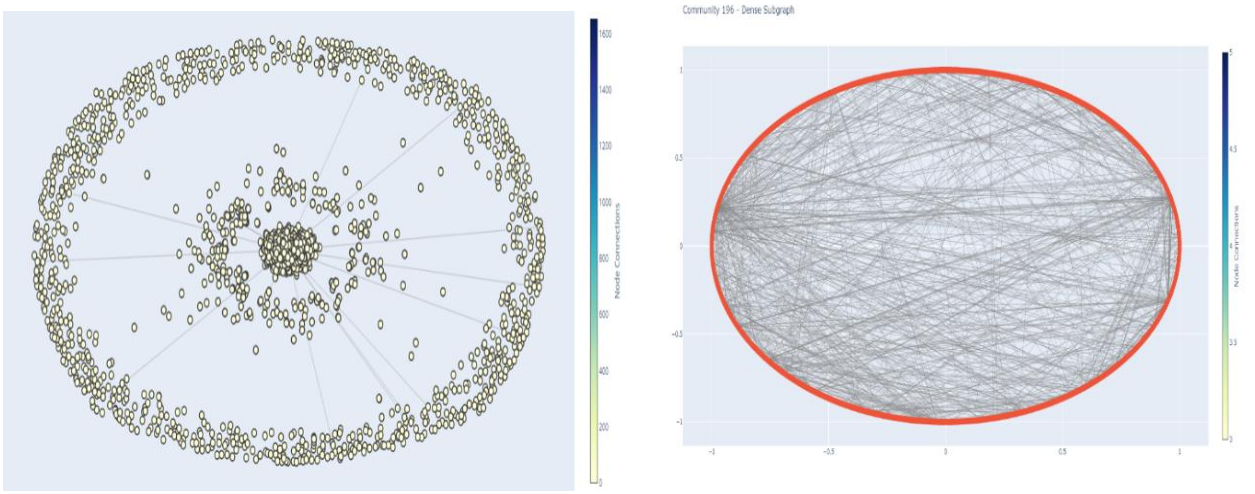


Figure 6: A subgraph of a few of the communities that satisfy the maximum threshold

Lastly, parallelism was employed in the iterative aspect of the modularity optimization and community aggregation process with the help of the framework. The framework may also be improved in efficiency by performing such steps in parallel on two or more network segments, especially when the network can be partitioned into segments such that the calculations of the modularity of interacting network segments are not immediately affected by one another. The proposed structure has boosted considerably the rate at which unusual dealings can be detected in an intricate monetary web when parallel processing is done at various stages of the examination. Such optimization allows financial institutions to identify any suspicious resources in time and effectively and can prevent the spread of fraudulent activities in the system, thus guaranteeing the health of the organization and its relations with buyers. This is a successful and scalable model used in identifying anomalous transactions in financial institutions. The system can identify suspicious activities which otherwise would go unnoticed because it uses graph analysis, community classification, as well as parallel processing. Figure 6 shows the results of the analysis in the form of subgraphs of some of the abnormal communities observed. NetworkX was used to produce the force-directed layout of the subgraph of suspicious communities of high-modularity. Nodes were selected based on consistent statistical ($\chi^2 > 0.25$) and modularity ($Q_{th} = 0.45$) thresholds, and community boundaries were annotated using anomaly scores to highlight key transactional clusters.

This visualization gives a concise summary of the unusual points in the financial network that should be investigated since it targets the communities with the greatest amount of it. Circular nodes are used to represent individual accounts or entities and connecting lines are used to represent financial transactions between them. The correlation of the accounts within these subgraphs' points to the possibility of fraudulent transactions or other questionable trends that would not have been revealed during a more standard analysis. This pictorial presentation of the financial institutions can determine and react to areas of concern which seem most pressing at that time enabling them to take proactive steps to avert all the risks and safeguard the integrity of their financial systems. The analysis of the dataset of transaction has revealed certain interesting facts about Benford Law. Figure 7 shows that the frequency of the first-digits of the transaction data is not close to the expected Benford distribution, and there is significant deviation. Such deviation implies that the data contains some transactions that are not obeying the natural and expected frequency of first digit. Although this deviation does not necessarily mean a form of fraud, it is certainly a red flag that needs to be investigated further. Digits that are front runners or are unrelated to Benford distribution can be indicators of manipulation, statistical abnormalities, or other measures of covering up fraud. Intimate scrutiny of these deviations can help financial institutions to identify suspicious transactions that might need further scrutiny by the framework. This analysis will serve as a good point of departure in detecting the anomalies and possible fraud in the complex web of financial transactions.

Financial institutions can then conduct a special inspection of the transactions involved by channeling their efforts to the points at which the transaction data does not comply with the Laws of Benford. After all, the lessons that can be learned through the analysis of the Law of Benford, as well as the other elements of the given framework, will enable financial institutions to identify and address fraudulent actions before they occur, thereby protecting the integrity of their systems and maintaining the trust of their clients.

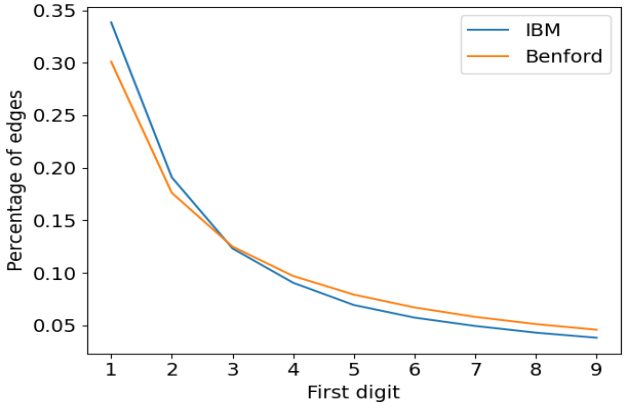


Figure 7: Transactions Deviation of IBM and Benford's dataset transactions with normal distribution

The suggested framework for identifying abnormal transactions in the banking sector has proven to provide better performance than those that are currently state-of-the-art and have also been tested and deployed on the same dataset and parameters. As Figure 8 demonstrates, the precision of the proposed method, which incorporates the use of graph mining and superior statistical algorithms, surpasses that of conventional rule-based systems, statistical anomaly detection, and machine learning applications. Although traditional rule-based systems can achieve accuracy rates of about 80%, statistical anomaly detection techniques can reach approximately 85%, and machine and deep learning techniques can achieve around 95%, the proposed framework's accuracy is impressive, with 96.57% on the IBM datasets. Moreover, the combination of Benford Law will enhance the framework's functionality in identifying irregularities that may indicate fraudulent activities. Through the analysis of the nature of transactions, including the frequency of leading digits, the framework will be used to identify anomalies that may reflect manipulations, irregularities, or deliberate efforts to conceal fraudulent activities.

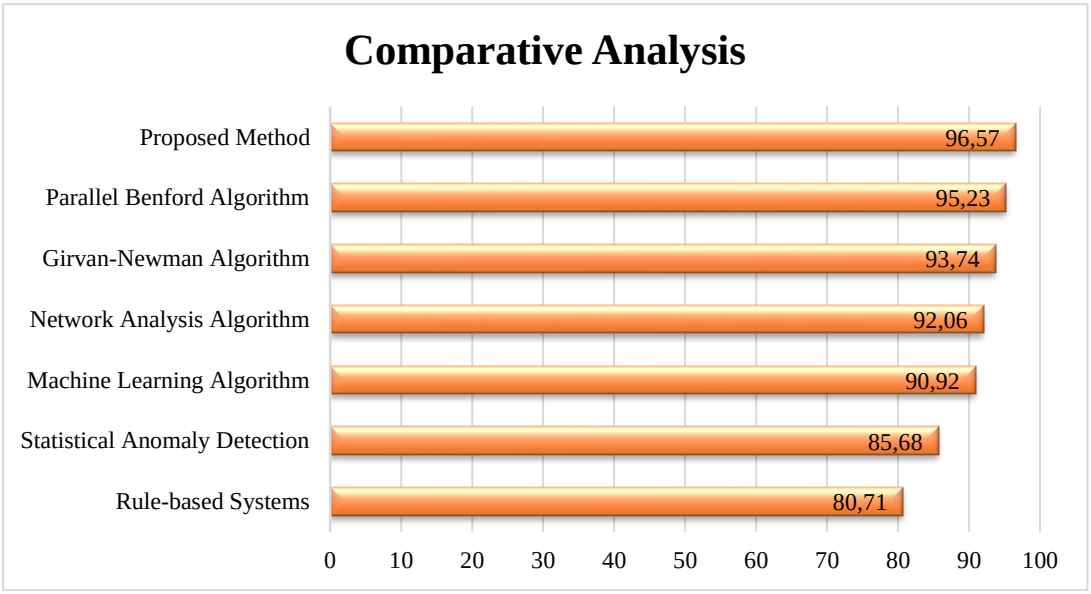
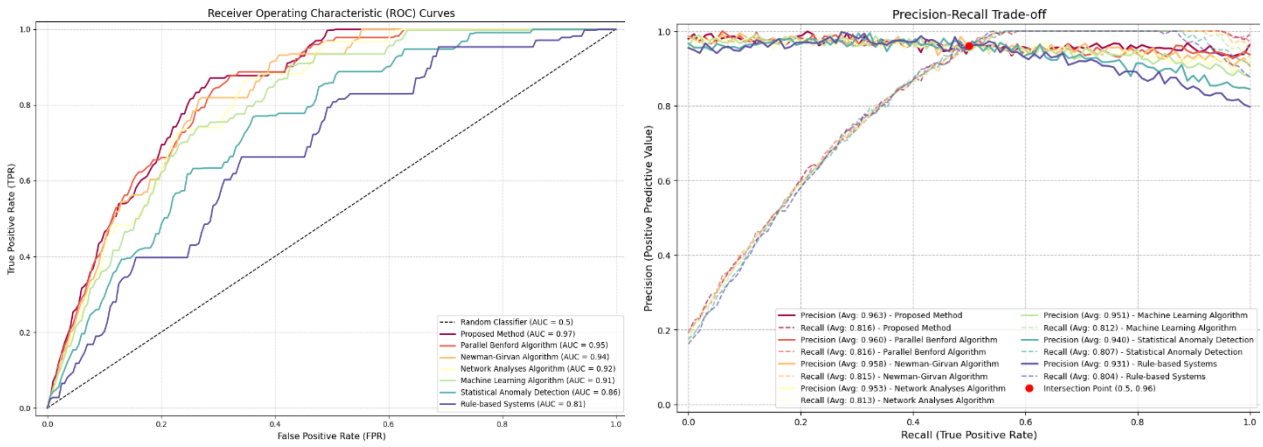


Figure 8: Comparative analysis of the proposed method with state-of-the-art methods



(a) ROC Curve (b) Precision-Recall Trade-Off

Figure 9: Evaluation metrics using ROC Curve and Precision-Recall Trade-Off

The respective receiver operating characteristic (ROC) curves and precision recall (PR) trade-offs of the proposed and the baseline algorithms are shown in Figure. 9. The ROC curve shows the compromise between the true positive rate and the false positive rate whereas the PR curve displays the connection between the optimal precision and the optimal recall at various thresholds. The steeper ROC curve and higher area under the curve, respectively, is an indication of the higher rejections of the model to distinguish among fraud and legitimate transactions, which confirms the possibility of the proposed framework to effectively distinguish between fraud and legitimate transactions. Equally, the results of the high PR performance show high degree of consistency, even amidst instances of disparity in classes, which is a typical feature of financial information. These graphical results prove the statistical power and capability to identify the suggested approach in comparison with traditional machine learning tools and rule-based methods. Table 5 offers a comparison analysis between the proposed approach of detecting fraud and the state-of-the-art methods that are used to detect fraud in banking transactions on various parameters.

Table 5: Comparison of state-of-the-art fraud detection techniques in banking transactions with the proposed approach

Technique	Approach	Accuracy (%)	Detection Time	False Positive Rate	Explainability	Adaptability	Limitations
Rule-based Systems	Manual definition of rules based on transaction characteristics (amount, location, etc.) to identify suspicious transactions.	70-80	Fast	High	Easy-to-understand rules	Low - Requires frequent rule updates to keep pace with evolving fraud tactics.	High false positives, static, and require continuous rule updates.
Statistical	Analyses transaction attributes (mean, standard deviation) for deviations from historical patterns.	80-85	Moderate	Moderate	They are limited - It is difficult to pinpoint specific reasons for flagging transactions.	Moderate - Adapts to changing patterns within a defined statistical framework.	May miss complex fraud schemes and be prone to false positives.
Machine Learning	Trains models (e.g., decision trees) to classify transactions as fraudulent or legitimate based on historical data.	85-90	Variable (depends on model complexity)	Moderate	The model can be further improved with interpretable decision tree models.	High - Learns from new data and adapts to evolving fraud patterns.	It requires extensive training data, is complex to implement, and has interpretable models that are not always achievable.
Proposed Approach (Graph Mining + Statistical Methods)	It utilizes Louvain's algorithm to construct a transaction network, analyzes connections, and applies Benford's Law to identify anomalies.	96.57	Fast (real-time analysis possible)	Low	Moderate - Requires understanding of network structure and interpretation of statistical deviations.	High-network analysis can adapt to new types of fraud involving connections between entities.	Requires domain expertise for network construction; computational cost can be high

The Louvain algorithm is used to construct a transaction network in the system, and it assists in identifying the abnormal patterns and connections between entities, such as accounts, merchants, and devices. The ability of this analysis to be in the form of graphs enables the users to have a good perspective of the entire network. The structure identifies the points of suspicious activity and normal work, a fact that would otherwise pass unnoticed. The accuracy is enhanced, which is achieved by integrating graph mining, and sophisticated statistical techniques. The combination of these methods reveals difficult patterns and relationships in transaction data, and rigid statistical analysis indicates anomaly. Moreover, the framework can be used to identify any emerging fraud scheme by analyzing the network, which gives it a considerable advantage over other previous methods. Even though it may be resource-intensive to initialize, the trained, parameter-optimized system provides close to real-time detection, once running, particularly on multi-core or GPU-accelerated systems. The flexibility to change and identify emerging trends is critical to financial institutions in international environments where fraudsters are using more advanced methods. The proposed study proves to be of positive contribution towards detection of fraud in banking transactions; it offers a powerful tool which is a combination of graph mining and powerful statistical methods. Financial institutions have been able to put their systems in place and ensure customers are not in danger of financial fraud due to its dynamic nature.

5 Conclusion and future scope

To enhance the level of effectiveness in terms of fraud, we offer an integrated system that combines the AntiBenford subgraph structure with the Louvain algorithm. It is a technique that uses graph mining and advanced statistics to identify patterns of fraudulent transactions. The AntiBenford subgraph structure marks the presence of anomalous subgraphs using the Benford principle, the description of frequency expectations of leading digits in real data. These subgraphs point at the areas of potential fraud. Unsupervised Louvain algorithm then deletes irregular transactions of the flagged AntiBenford subgraphs, irrespective of frequency. The blend of a graph and statistical analysis can produce a strong detection function that is able to detect intricate fraud patterns, which could be overlooked through a traditional method. We apply parallel processing so that we can get maximum performance by allocating work among several processors. This accelerates the analysis process and also utilizes resources in an efficient manner enhancing accuracy and reliability. In our test on the IBM data we were sure of a good score of 96.57 % accuracy. This finding demonstrates that the suggested plan is efficient in exposing fraudulent transactions in the sophisticated financial environment. Our framework decreases false positives as compared to the traditional methods that end up disrupting valid transactions.

The strategy is also simple to operate in real time to avoid possible losses. Although domain knowledge is useful in the interpretation of network structures and statistical abnormalities, explainability is generally better than what high-level complex machine-learned models can achieve. Most crucially, the flexibility of the strategy is one of its strengths. Network analysis helps the system to detect new and emerging fraud schemes with the help of the relationship between entities. This research presents a strong argument as to why the use of graph mining and fined statistical instruments should be applied in detecting fraud in banks. The solution is very precise, with a low rate of false positive, it responds very fast, and can adjust to emerging trends in fraud cases. The strengths have made it an important instrument to financial institutions. Ensuring security is also a better way to safeguard customers and ensure confidence in the banking system. The research has provided a good base that can be used to develop the solution to wider uses of fraud-detection using network analysis. The researchers identified several promising directions for future research, which would have led to significant improvements in the system's capabilities and practical use. Network feature engineering is one area that can be focused on in the future.

The system's capability to detect complex patterns signaling fraud can be enhanced by investigating other network-based characteristics, including measures of network centrality, network communities, and the flow of information within the transaction network. It would be beneficial to extend the framework to manage networks, which are subject to change, as this would increase its practical use. Further validations and generalizations of the proposed approach will be carried out in future studies, where extended statistical significance testing, such as Wilcoxon or t-tests, and solid evaluations will be performed. It is also devoted to the wideness of the given AntiBenford-Louvain framework, achieved through the practical use of the temporal GNN structure to design dynamic transaction graphs, the meta-learning approaches aimed at fast adaptation to new types of fraud, and the semi-supervised detection of anomalies through the effective utilization of unlabeled transactional data. The model will need to continually adjust to new patterns of transactions and entities to stay ahead of the fraudsters. It can be effectively used to prevent fraud schemes by creating algorithms that constantly update the network structure, keeping it efficient in protecting the financial system. The combination of the machine learning model and the proposed approach represents a very promising opportunity. Furthermore, comparing the effectiveness of models in the real-world, large-scale datasets of banking transactions would be an important step towards implementation. This can provide valuable information regarding the scalability and effective performance of the system in a production environment. The directions will allow these researchers to narrow and broaden their focus on the suggested direction, eventually leading to a stronger and more secure financial ecosystem for all stakeholders.

References

[1] Somkunwar, R. K., Pimpalkar, A., Katakdound, K. M., Bhide, A. S., Chinchalkar, S. P., & Patil, Y. M., (2023). A Fraud Detection System in Financial Networks Using AntiBenford Subgraphs and Machine Learning Algorithms. In International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, pp. 1-6. https://doi.org/10.1109/AIKIIE60097.2023.10390325.

[2] Deckert, J., Myagkov, M., & Ordeshook, P., (2011). Benford's Law and the detection of election fraud. Political Analysis, 19(3), 245-268. https://doi.org/10.1093/pan/mpr014

[3] Dyck, A., Morse, & A., Zingales, L., (2010). Who blows the whistle on corporate fraud?. The Journal of Finance, 65(6), 2213-2253. https://doi.org/10.1111/j.1540-6261.2010.01614.x

[4] Ziming, Z., Zhaoxuan, L., Zhihao, Z., Jiongchi, Y., Zhuoxue, S., Xiaofei, X., Fan, Z., & Rui, Z., (2024). DDoS family: A novel perspective for massive types of DDoS attacks. Computers & Security, 138, 103663. https://doi.org/10.1016/j.cose.2023.103663

[5] Yajing, L., Zhengya, S., & Wensheng Z., (2023). Improving fraud detection via hierarchical attention-based Graph Neural Network. Journal of Information Security and Applications, 72, 103399. https://doi.org/10.1016/j.jisa.2022.103399

[6] Jianping, L., Yanpeng, C., Yinghui W., & Xiaoqian, Z., (2023). Tracking down financial statement fraud by analyzing the supplier-customer relationship network. Computers & Industrial Engineering, 178, 109118. https://doi.org/10.1016/j.cie.2023.109118

[7] Xiaodong, Z., Jian, G., Xinchang, Z., & Guiqing, L., (2023). Attack scenario reconstruction via fusing heterogeneous threat intelligence. Computers & Security, 133, 103420. https://doi.org/10.1016/j.cose.2023.103420

[8] Li, J., & Yang, D., (2023). Research on Financial Fraud Detection Models Integrating Multiple Relational Graphs. Systems, 11, 539. https://doi.org/10.3390/systems11110539

[9] Lingfei, R., Ruimin, H., Dengshi, L., Yang, L., Junhang, W., Yilong, Z., & Wenyi, H., (2023). Dynamic graph neural network-based fraud detectors against collaborative fraudsters. Knowledge-Based Systems, 278, 110888. https://doi.org/10.1016/j.knosys.2023.110888

[10] Chen, T., & Tsourakakis, C., (2022). AntiBenford Subgraphs: Unsupervised Anomaly Detection in Financial Networks. KDD '22: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2762 – 2770. https://doi.org/10.1145/3534678.3539100

[11] Xiaobing, S., Wenjie, F., Shenghua, L., Yuyang, X., Siddharth, B., Bryan, H., Wenhan, W., & Xueqi, C., (2022). MonLAD: Money Laundering Agents Detection in Transaction Streams. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22). Association for Computing Machinery, New York, NY, USA, 976–986. https://doi.org/10.48550/arXiv.2201.10051

[12] Gridley, J., & Seneviratne, O., (2022). Significant Digits: Using Large-Scale Blockchain Data to Predict Fraudulent Addresses. In IEEE International Conference on Big Data, Osaka, Japan, pp. 903-910. https://doi.org/10.1109/BigData55660.2022.10020971.

[13] Shigang, W., Jianping, L., Xiaoqian, Z., & Mingxi, L. (2022). Analysis of financial fraud based on manager knowledge graph. Procedia Computer Science, 199, 773-779. https://doi.org/10.1016/j.procs.2022.01.096

[14] Xuting, M., Hao, S., Xiaoqian, Z., & Jianping, L., (2022). Financial fraud detection using the related-party transaction knowledge graph. Procedia Computer Science, 199, pp. 733-740. https://doi.org/10.1016/j.procs.2022.01.091

[15] Pourhabibi, T., Kok-Leong, O., Booi, H. K., & Yee, L. B., (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Decision Support Systems, 133, 113303. https://doi.org/10.1016/j.dss.2020.113303

[16] Kurshan, E., Shen, H., & Yu, H., (2020). Financial Crime & Fraud Detection Using Graph Computing: Application Considerations & Outlook. In Second International Conference on Transdisciplinary AI (TransAI), 125-130. https://doi.org/10.1109/TransAI49837.2020.00029.

[17] Ganguli, R., Mehta, A., Debnath, N., Aljahdali, S., & Sen, S., (2020). An integrated framework for friend recommender system based on graph theoretic approach. In Gordon Lee and Ying Jin (editors). Proceedings of 35th International Conference on Computers and Their Applications, 69, pp 242-255. https://doi.org/10.29007/4bwn

[18] Ozili, P., (2020). Advances and issues in fraud research: a commentary. Journal of Financial Crime, 27(1), 92-103. https://doi.org/10.1108/JFC-01-2019-0012

[19] Wang, D. et al., (2019). A Semi-Supervised Graph Attentive Network for Financial Fraud Detection. 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, pp. 598-607. https://doi.org/10.1109/ICDM.2019.00070.

[20] Bernaschi, M., Bisson, M., Mastrostefano, E., & Vella, F., (2018). Multilevel Parallelism for the Exploration of Large-Scale Graphs. In IEEE Transactions on Multi-Scale Computing Systems, 4, 3, pp. 204-216. https://doi.org/10.1109/TMSCS.2018.2797195.

[21] Kiljan, S., Vranken, H., & Eekelen, M., (2018). Evaluation of transaction authentication methods for online banking. Future Generation Computer Systems, 80, pp. 430-447. https://doi.org/10.1016/j.future.2016.05.024

[22] Nica, I., Delcea, C., & Chiriță, N., (2024). Mathematical Patterns in Fuzzy Logic and Artificial Intelligence for Financial Analysis: A Bibliometric Study. Mathematics. 12(5):782. https://doi.org/10.3390/math12050782

[23] Askari, S., & Hussain, A., (2020). IFDTC4.5: Intuitionistic fuzzy logic-based decision tree for E-transactional fraud detection. Journal of Information Security and Applications, 52, 102469. https://doi.org/10.1016/j.jisa.2020.102469

[24] Chinchalkar, S.P., & Somkunwar, R.K., (2024). An innovative keylogger detection system using machine learning algorithms and dendritic cell algorithm. Revue d'Intelligence Artificielle, 38, 1, pp. 269-275. https://doi.org/10.18280/ria.380128

[25] Pham, T., & Lee, S., (2017). Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. https://doi.org/10.48550/arXiv.1611.03941

[26] Kamble, N., & Mishra, N., (2024). Hybrid optimization enabled squeeze net for phishing attack detection. Computers & Security, 144, 103901. https://doi.org/10.1016/j.cose.2024.103901

[27] Ngai, E.W.T., Yong, H., Wong, Y.H., Yijun, C., & Xin, S., (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50, 3, pp. 559-569. https://doi.org/10.1016/j.dss.2010.08.006

[28] Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. (2009). Credit card fraud detection: a fusion

approach using Dempster–Shafer theory and Bayesian learning. Information Fusion, 10(4), pp. 354-363. https://doi.org/10.1016/j.inffus.2008.04.001

[29] Saini, D. K. J. B., Shelke, N., Prajwalasimha, S. N., Pimpalkar, A., Kumar G. H., & Monish, L., (2025). Advanced Deep Learning for Real-Time Fraud Detection in Banking: Scalable and High-Accuracy Solutions, In Proceedings of the 6th International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-6. https://doi.org/10.1109/INCET64471.2025.11139964

[30] Prajwalasimha, S. N., Saini, D. K. J. B., Shelke, N., Pimpalkar, A., Kumar G. H., & Shree P. H. R., (2025). Trustworthy and Interpretable AI for Robust Fraud Detection in Financial Transactions, In Proceedings of the 6th International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-6. https://doi.org/10.1109/INCET64471.2025.11140975.

[31] Kumar G. H., Shelke, N., Pimpalkar, A., Saini, D. K. J. B., Prajwalasimha, S. N., & Dileep K., (2025). Zero Trust-Enabled Digital Twins for Real-Time Anomaly Detection in Industrial Cyber Physical Systems," In Proceedings of the Third International Conference on Networks, Multimedia and Information Technology (NMITCON), Bengaluru, India, pp. 1-6. https://doi.org/10.1109/NMITCON65824.2025.11188236

[32] Hiwase, A., Pimpalkar, A., Dange, B., Thakre, N., Jaiswal, S., & Mankar, T. (2025). EBSSPA: Efficient Deep Learning Model for Enhancing Blockchain Scalability and Security Through Fusion Pattern Analysis. Acta Informatica Pragensia, 14 (3), pp. 316-339. https://doi.org/10.18267/j.aip.260

[33] Alatrista-Salas, H., Hancco, J. F. A., & Espinoza-Villalobos L., (2025). Algorithms For Anomaly Detection on Time Series: A Use Case on Banking Data. Informatica. 49, 13, pp. 203–220. https://doi.org/10.31449/inf.v49i13.6243

[34] Gupta, P., Arora, M., & Thakur H. K., (2025). Community Detection in Social Networks: A Deep Learning Approach Using Autoencoders. Informatica, 49, 5, pp. 195–212. https://doi.org/10.31449/inf.v49i5.7018

[35] Lun, L., John, C., Doyle, W. W., & David, A., (2005). Towards a Theory of Scale-Free Graphs: Definition. Properties, and Implication, 2, 4.

[36] Hanghang, T. & Christos, F., (2006). Center-piece subgraphs: problem definition and fast solutions. In Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos (Eds.). ACM, pp. 404-413. https://doi.org/10.1145/1150402.1150448

[37] Berger, A., & Theodore P. H., (2011). A basic theory of Benford's Law. Probab Surveys, 8, 1 – 126. https://doi.org/10.1214/11-PS175

[38] Pereira, A. G., & Kohlsdorf, T., (2023). Repeated evolution of similar phenotypes: Integrating comparative methods with developmental pathways. Genetics and Molecular Biology, 46, 1 suppl 2. https://doi.org/10.1590/1678-4685-GMB-2022-0384

[39] Kanezashi H., & Suzumura, T., (2016). An incremental local-first community detection method for dynamic graphs. 2016 In Proceedings of the IEEE International Conference on Big Data (Big Data), Washington, DC, USA, pp. 3318-3325. https://doi.org/10.1109/BigData.2016.7840991

[40] Erik A., Jovan B., Luc von N., Béni E., Andreea A., & Kubilay A., (2023). Realistic synthetic financial transactions for anti-money laundering models. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23). Curran Associates Inc., NY, USA, Article 1300, pp. 29851–29874. https://dl.acm.org/doi/10.5555/3666122.3667422

[41] Cappi, C., Chapdelaine, C., Gardes, L., Jenn, E., Lefevre, B., Picard, S., & Soumarmon, T., (2021). Dataset Definition Standard (DDS). https://doi.org/10.48550/arXiv.2101.03020