

Original Article

Explainable Graph Learning For Anomaly Detection in Distributed Financial Data Systems

Jospeh Kuruvilla

Software Engineer, USA.

Received: 06-07-2026

Revised: 06-22-2026

Accepted: 07-03-2026

Published: 07-10-2026

ABSTRACT

The rapid growth of digital banking, electronic payments, and financial technology has increased the complexity of financial anomaly detection. Traditional rule-based and machine learning methods often struggle to identify sophisticated fraud involving multiple accounts, connected entities, and time-dependent transactions. Graph neural networks (GNNs) can capture these relationships but face interpretability challenges in regulated financial environments. This paper proposes an explainable graph learning framework for anomaly detection in distributed financial systems. The framework integrates heterogeneous temporal graph construction, GNN-based representation learning, anomaly scoring, explanation generation, and distributed model monitoring. It identifies transaction-level, account-level, and subgraph-level anomalies by analyzing suspicious transactions, coordinated account behavior, temporal patterns, and structural deviations. A research methodology is outlined to evaluate detection accuracy, explanation fidelity, computational efficiency, robustness, and latency against conventional and graph-based methods. The framework aims to support transparent, scalable, and accountable financial anomaly detection, with its effectiveness requiring validation through empirical studies on reliable financial datasets.

KEYWORDS

Explainable Artificial Intelligence, Graph Neural Networks, Financial Anomaly Detection, Distributed Data Systems, Financial Fraud Detection, Temporal Graphs, Graph Explainability, Transaction Monitoring, Model Interpretability, Financial Cybersecurity.

1. INTRODUCTION

The digital transformation of financial services has created complex and interconnected transaction processing environments. These include traditional commercial banks, payment processors, digital wallets, merchant platforms, lending institutions, and cross-border financial networks. Transactions are being processed on geographically dispersed servers, cloud-based infrastructure, message queues, and heterogeneous database systems. While the use of these technologies enhances the availability, scalability, and processing efficiency of the financial services industry; it also presents many opportunities for fraudsters to hide their fraudulent activity. Fraudsters can distribute fraudulent transactions among many different accounts, utilize existing relationships with legitimate businesses, or perform a series of seemingly unrelated transactions in a coordinated manner. Therefore, effective detection of fraudulent activity will require an understanding of how individual transactions relate to each other as well as how each transaction relates to the various entities involved in performing those transactions.

Traditional anomaly detection includes statistical profiling, rule-based monitoring, logistic regression, decision trees, support vector machines, and ensemble learning methods. All of these techniques continue to have use due to being more understandable, more mature in operational terms, and better able to handle structured transaction attributes than other types of techniques. However, the effectiveness of these traditional techniques will be reduced when suspicious behavior develops as a result of complex relational dependencies as opposed to unusual individual values. For example, multiple transactions may be within expected monetary limits but when combined into a sequence of transfers via intermediate accounts create a suspicious sequence. A feature-based classifier will not recognize such a pattern unless the underlying relationships among the various transactions can be explicitly represented.

The use of graph learning provides a solution to this problem by viewing financial transactions in the context of a network. Nodes represent individual customers, bank accounts, merchants, devices, payment terminals and financial institutions. Edges represent the relationships between these nodes (e.g., transfers, shared IDs, account ownership, etc.). Graph neural network models develop representations based on information from neighboring nodes and their relationships. This allows the model to capture the structure of the data which might be lost using a traditional independent-observation model. Hamilton et al. (2017) proposed GraphSAGE, an inductive representation learning method that creates node representations using neighborhood sampling and aggregation. GraphSAGE can provide a base for applications with dynamic entity creation.

In addition to having high predictive power, predictive ability alone is not enough to effectively identify financial anomalies. Financial organizations will need to know why a transaction has been flagged, what connections have caused the alert, if the model used reliable information, and how investigators can verify the model's findings. Black-box models may make it difficult to complete compliance reviews, internal audits, dispute resolutions, and prioritize the use of

investigative resources. Thus, Explainable Artificial Intelligence (XAI) is now considered to be a core element of graph-based financial monitoring systems rather than simply a value added visualization tool.

The difficulty of distributed financial data systems is greater. Financial transaction data can be divided among databases, regions, organizations, and processing services. Delays in data capture, different timestamp conventions, incomplete connections between entities, and limited communication capabilities will make it difficult to build graphs and use models to make predictions. In addition to the fact that the way people behave financially changes over time, fraud patterns change as well based on new security measures. Therefore, a good fraud detection system needs to be able to predict accurately, explain itself clearly, protect private information, do so at a reasonable price, and provide results quickly.

The proposed explainable graph learning framework presented in this study, which integrates heterogeneous temporal graph modeling, graph-based anomaly scoring, explanation generation, and distributed monitoring, has as its primary goal the development of a technically sound design for detecting complex financial anomalies and providing supporting evidence that can be examined by humans investigating the issue.

The specific objectives are to:

- Develop a graph representation for distributed financial transactions and their associated entities.
- Integrate graph neural network learning with transaction, behavioral, temporal, and structural anomaly indicators.
- Generate human-interpretable explanations that identify influential entities, transactions, and relationships.
- Define an evaluation framework covering detection quality, explanation fidelity, scalability, latency, and robustness.
- Identify implementation challenges involving privacy, concept drift, distributed graph consistency, and responsible financial AI.

The main research question is: How can explainable graph learning be used to identify and investigate complex anomalies within distributed financial data systems, without compromising on operational efficiency and providing reliable decision-making support?

The primary focus of this work is the development of a comprehensive architecture and evaluation process for evaluating financial models, as opposed to demonstrating the superior performance of a novel algorithm on an actual-world financial data set.

2. LITERATURE REVIEW

2.1. Conventional Financial Anomaly Detection

Traditionally, there have been three approaches used in financial anomaly detection: rule-based systems; statistical analysis; and supervised classification. Rule-based systems detect transactions which violate pre-defined criteria (e.g., numerous transfers occurring over a brief period of time; transactions in unexpected geographic areas; or activities that are inconsistent with previous behavior). Statistical analysis can be used to determine when current activities deviate from those experienced historically. Supervised machine learning uses training data with known labels to differentiate legitimate and non-legitimate activity.

These methods are foundational to how financial monitoring can occur. For example, traditional classification models are typically very successful if there is enough information about each transaction (attributes) and enough labeled examples of what constitutes "fraudulent" activity to train a model. However, financial fraud poses several unique difficulties. First, only a fraction of all transaction activity is often determined to be fraudulent. Second, label generation often occurs long after an investigation has concluded. Third, new forms of fraudulent activity can occur at any time, which means that previous training data may not contain examples of such activity.

Akoglu et al. (2015) studied anomaly detection in complex networks and found that anomalous behavior can occur when there are unusual nodes, relationships, or sub-graphs. The study is applicable to financial transactions since fraud can be expressed in terms of coordinated relationships rather than as deviations from the normal value of a single transaction. Thus, the authors recommend that anomaly detection in finance should focus on the overall pattern of financial activities in addition to the values of individual transactions.

2.2. Graph Learning for Financial Fraud Detection

The use of graph-based learning for the detection of financial anomalies is an important step forward. This is primarily due to the fact that graph-based learning provides the ability to model the relationships between different entities. For example, a transaction network can identify common accounts, multiple transactions with the same counterparty, circular money movements, groups of individuals engaged in suspicious behavior, and anomalous patterns of interaction. These types of relationships provide insight into why some transactions may appear legitimate, yet collectively they represent abnormal behavior.

Graph neural networks can generate node embeddings (i.e., representations) based on both node features as well as the connectivity of those nodes within the graph structure. Kipf and Welling (2017) developed a popular graph convolutional network architecture that has been utilized in semi-supervised node classification tasks. Veličković et al. (2018) developed graph attention networks, which use attention mechanisms to assign varying levels of significance to each neighbor node. Hamilton et al. (2017) showed that inductive neighborhood aggregation can be effective when generating representations for nodes that have never been seen before.

Graph convolutional networks are the basis of a strong foundation to combine information from the surrounding area. Graph attention networks can focus on potentially useful relationships. Inductive methods allow the use of dynamic graphs that continually add new users and transactions. However, standard graph neural networks do not have a built-in mechanism to explain why they make certain predictions. Attention weights, for example, can show us which neighboring nodes have had computational priority; however, this does not mean that those same nodes were the reason why a prediction was made.

The recent literature explicitly discusses the use of graph neural networks for financial fraud detection. Motie and Raahemi (2024) systematically reviewed graph neural network techniques, identifying methodological variations among different graph types, learning methods, and financial domains. Cheng et al. (2025) provided additional analysis on the use of graph neural networks for financial fraud detection, analyzing model architecture, implementation challenges, and ongoing research gaps. Both reviews demonstrate the significance of graph learning; however, they also emphasize the need to address implementation issues in conjunction with architectural design.

2.3. Explainable Artificial Intelligence for Graph Models

In the context of understanding how models make predictions, explainability is focused on producing clear explanations of how a model generates a prediction. For example, in the field of financial anomaly detection, a good explanation might describe which specific transactions, entities, features, and/or relationships generated an alert. The explanation needs to be sufficiently detailed so as to assist in directing investigations, yet avoid making unfounded assertions concerning criminal intent.

In 2019, Ying et al. developed GNNExplainer, a technique to identify which parts of a graph neural network (GNN) prediction are most important by identifying a compact subgraph and/or feature subset. This idea can be used in the context of financial monitoring. An investigator could use this to identify a few key transactions and/or relationships rather than having to look at all of the transactions and/or relationships within the larger transaction network. It is important to remember that although the explanations created using the model's associations help us understand why the model made a certain decision, they do not necessarily create a causal link. Additionally, the effectiveness of the explanation will depend upon several factors including the model used, the data used, and what you want the explanation to accomplish.

Lundberg and Lee (2017) created SHAP, a general framework for explaining model predictions using feature attribution. Although SHAP supports explanations based upon individual transaction level features, as well as auxiliary models, there are additional considerations needed when using feature attribution on a graph model due to the potential influence of relationships and neighborhoods on the prediction. Ribeiro et al. (2016) developed LIME, a technique that approximates a model's behavior at each point to provide explanation for each prediction. Both techniques can assist in financial investigation; however, as the graph underlying the model evolves,

or as various perturbations result in different local approximations, the explanations provided by either technique could be unstable.

In financial application, it is essential to distinguish between the accuracy of a model's prediction and the quality of its explanation. While a very accurate model can generate unstable or confusing explanations, a model that is understandable may not be able to find complicated relationships. Therefore, a good framework should assess both of these aspects separately. Khoozani et al. (2026) studied how graph learning and explainable models have been used to model financial risk and identified some issues with respect to interpretability, scalability, concept drift, and governance.

2.4. Distributed Data Processing and Temporal Graph Analysis

Distributed financial systems process transactions through several services and data stores. The system's architecture can include: Ingestion Pipelines Event Streaming Feature Computation Graph Storage Model Serving Investigation Systems A model designed to detect anomalies in a distributed financial system will need to handle: Incoming Information Duplicate Events Out-of-Order Messages Incomplete Records Changing Graph Connectivity

Temporal graph modeling is significant for financial institutions since most financial relationships are dynamic. For example, a credit card holder who typically makes a small number of purchases each month can make many more purchases than usual in a given month. Alternatively, a transaction sequence that seems to be a normal pattern of activity can appear to be suspicious if it occurs during an atypically fast time frame. Therefore, temporal data provides insight into behavior patterns that look similar when viewed in a static representation of the network of connections among entities, but vary based upon their sequential or temporal order.

In addition to introducing other concerns, distributed processing also has implications. For example, a single processing service will only have limited visibility into the network when a relationship exists across multiple data partitions. If the construction of a graph is based upon inconsistent or delayed information, then a model will be able to provide explanations which fail to capture significant interactions. Therefore, the architecture being proposed includes; data provenance, event-time consistency, graph snapshot management, and explanation traceability.

2.5. Research Gaps

Existing studies have demonstrated the utility of graph learning in identifying relational fraud and how essential explainability is for making high-stake decisions. However, there are still some implementation issues. First, detection, explanation, and distributed processing have usually been investigated as separate entities. For a complete financial system to work, these components must function collectively so that there is a logical link between the initial transaction, the prediction made by the model, and the explanation generated.

Second, the quality of the explanation is different than how accurate the anomaly score is. Therefore, for financial applications, we will use evaluation metrics that measure the fidelity, stability, sparsity, and investigator usefulness of an explanation. Third, class imbalance and concept drift are two major problems when evaluating financial anomaly detection algorithms. When we randomly split our transactional data into training and test sets, we could create "temporal leakages" or have "related" entities appear in both partitions. This can result in an overstatement of the generalization performance of our algorithm.

Distributed deployment will require us to explicitly consider the following: inference latency, communication overhead, cross-partition relationships, privacy restrictions and incomplete graph observations. Lastly, an anomaly score does not prove that a transaction was fraudulent. Therefore, the system should be able to convey its uncertainty about whether a transaction is fraudulent or not; preserve any evidence related to the potential fraud; and allow for human review. This framework proposes to address the above issues through a combination of detection, explanation, distributed systems design, and operational evaluation within a single research model.

3. RESEARCH METHODOLOGY

3.1. Research Design

This study is conducted using a methodological approach based on design. The study combines existing work on graph learning, anomaly detection, explanations of results, and distributed finance processing to develop an integrated framework. The final design developed in this study will be used for empirical testing of the framework, rather than being demonstrated through a fully completed experimental test.

The suggested evaluation will use time-ordered transactional data, comparing the proposed graph-based method against traditional and graph-learning methods. The evaluation will include assessing the ability to predict future events, as well as the operational performance of each method, thereby allowing for assessment of the entire system as a practical decision support tool rather than simply as a classifier.

3.2. Data Representation and Graph Construction

Let a financial transaction graph at time (t) be represented as

$$G_t = (V_t, E_t, X_t, R_t, T_t)$$

The variables are defined as follows: (V_t) represents financial entities; (E_t) represents connections among entities; (X_t) holds both entity and connection attributes; (R_t) contains the different connection types; and (T_t) holds all of the time-related data.

Nodes are used to model customers, accounts, merchants, devices, payment terminals, and institutions. Edges model money transfers, account ownership, shared devices, common identifiers, and merchant interactions. Examples of edge attributes include transaction amount, currency,

timestamp, channel, and transaction status. Node attributes include historical transaction frequency, account tenure, aggregate behavioral statistics, and previously established risk indicators.

In general, if you want to use a heterogeneous graph for your data representation, then there are times when you should. If you have relationships that mean different things to each other, such as a transfer of funds from one account to another, and also the fact that both accounts share the same device; this will help keep you from treating every relationship equally.

Graph construction must occur utilizing only those data that will exist at the point of making predictions. No future transaction data, results from subsequent investigations, or later confirmations of fraudulent activity can become part of historical feature representations. In addition, the system should use event-time windows and versioning of the graphs as needed to enable reproducible analysis.

3.3. Proposed Model Architecture

The suggested structure includes five primary levels: distributed data ingestion, time-series graph creation, graph-based abnormality rating, reason for finding, and investigator's response.

The ingestion phase of processing collects transactions from authorized financial data providers. During this phase, validation processes identify malformed records, duplicate events, missing fields, and inconsistencies in time stamps. After being validated, a standardized version of each transaction is passed to graph construction services which will update all associated nodes, edges and temporal characteristics.

The graph learning stage produces representations for entities and transactions. For node (v), a generic message-passing operation can be expressed as:

$$h_v^{(l+1)} = \sigma(W_l h_v^{(l)} + AGG_{u \in N(v)} \phi_l(h_v^{(l)}, h_u^{(l)}, e_{uv}))$$

where $(h_v^{(l)})$ denotes the representation of node (v) at layer (l), $(N(v))$ denotes its neighborhood, (e_{uv}) represents edge attributes, (W_l) is a learnable transformation, (ϕ_l) computes relationship-aware messages, (AGG) aggregates neighboring information, and (σ) is a nonlinear activation function.

The anomaly scoring stage combines learned graph representations with transaction-level and temporal indicators. A general scoring function is

$$S(v, t) = f_\theta(h_v(t), x_v(t), r_v(t), d_v(t))$$

Where $(S(v, t))$ is the anomaly score, $(h_v(t))$ is the learned representation, $(x_v(t))$ includes all observed attributes, $(r_v(t))$ describes the relational pattern, and $(d_v(t))$ is the temporal deviation.

The function (f_{θ}) can either be a supervised classifier, a reconstruction-based detection method, or a hybrid scoring model.

A hybrid method is best used when there is limited data with which to label fraudulent activities. The supervised part of the model would utilize past instances where fraud was identified, and the unsupervised/self-supervised portion of the model would locate deviations from what has been learned about normal behavior. It is important to note that the anomaly score generated by each part of the model needs to be calibrated prior to combining them, otherwise the numerical scale of both may not be directly comparable.

3.4. Explainability Mechanism

The Explanation stage provides the evidence that was used to make each alert. When a transaction is determined to be suspicious, the Explanation stage can provide a small sub-graph of the transaction, the originating account, destination accounts, relevant intermediary accounts and the relationships that led to the prediction. It can also identify temporal anomalies including increases in frequency of transfers, concentrations of transfers in newly connected accounts, etc.

The suggested methodology will employ both graph-level substructure explanations and feature attribution. Using GNNExplainer-style techniques will help us to find out which portions of our graph have the most significant impact on the final decision. Additionally, using feature attribution techniques will allow us to understand how much each transaction attribute contributes to the overall result. Then, we can create a rule-based presentation layer that will allow us to turn those signals into structured evidence. For example, "unusual transfer frequency", "multiple recently connected counterparties", or "abnormal transaction timing".

The explanation should provide the model version, the time at which the score was created, a copy of the appropriate graph, and reference to any supporting data that can be used to generate the same result. Providing this information will help an investigator replicate the decision and determine if the explanation was generated using the same information that was available at the time the alert was triggered.

The critical safeguard is to differentiate between what was actually observed and what could potentially be inferred as a risk. To illustrate this, the fact that an individual repeatedly transfers funds between multiple accounts can provide justification for further investigation, however, it does not provide definitive proof of money laundering or intent to commit fraud.

3.5. Training and Evaluation Protocol

The experimental design should employ a temporal division of training, validation, and testing sets. Training data will consist of older transactions, while model selection data will be composed of subsequent transactions, and the most current transactions will serve as the evaluation

set. If possible, a second level of testing should be conducted using either an entity-disjoint or group-based strategy to evaluate the ability to generalize to new, never-before-seen accounts or institutions.

The baseline methods for this study will consist of the following: (1) a rule-based detection algorithm, (2) a traditional machine learning-based classification algorithm that utilizes features extracted from transactions, (3) a non-explainable graph-based model, and (4) an explainable graph-based model. Each of these models will utilize similar data inputs and be evaluated over the same period of time.

The performance of the model can be evaluated using several metrics, including: precision, recall, F1-score, area under the precision-recall curve (AUPRC), and false-positive rate. AUPRC is particularly useful when there are very few suspicious transactions, however, there is not one single metric that adequately reflects the operational utility of the model. When investigators have a set amount of time to review alerts, we should also measure the precision of those alerts that have been ranked most highly.

Evaluation of explanations needs to assess fidelity, sparsity, stability and how helpful they are to investigators. Evaluation of operations will evaluate median and 95th percentile inference latencies, throughputs, resource usage, communication costs and performance during periods when there is a delay in receiving events or when events are missing.

The methods section must include ablation studies to determine how much the contributions from temporal features, heterogeneous relationship types, graph aggregation, and explanation mechanisms affect the overall results. It should also assess explanation quality separately because there is no direct measurement of explanation quality in terms of anomaly detection metrics.

3.6. Proposed System Workflow



Fig 1 - Explainable Graph Learning Workflow for Distributed Financial Anomaly Detection

Fig 1 shows the end-to-end workflow proposed. Each distributed transaction event is first normalized and then used to construct a temporal heterogeneous graph. After constructing the

graph, it produces an anomaly score for each event. Also, each event has an associated explanation component that provides supporting evidence for why the event was flagged as an anomaly. Results from human investigations may be used to improve future iterations of the model through data governance, label validation, and controlled retraining procedures.

3.7. Distributed Deployment and Governance

The suggested solution should be composed of five distinct service-based systems: Transaction Ingestion, Graph Updating, Model Inference, Explanation Generation, and Case Management. Each service will have its own ability to scale independently based on the amount of work being done. As well as being able to update models without having to rewrite the complete transaction processing pipeline, the separate services must also maintain the integrity of the data that was utilized to create the alert.

Distributed graph processing systems have the ability to partition a graph in a way which keeps frequently communicating entities near each other and allows for cross partition communications as needed. While local neighborhood aggregation can potentially decrease the amount of communication required, it does so at the expense of potentially omitting relationships that exist across different partitions. Therefore, the system must maintain track of graph completeness and indicate if the reason given for explaining something is due to the use of a complete or partial neighborhood.

The same as privacy and access controls, financial data must be processed in a manner that includes adequate authorization, encryption, retention, and access controls. If possible, sensitive identifiers should either be removed or pseudonymized to the point where they can still allow for investigation of potential threats but do not identify individuals. When there is no way for multiple organizations to share their transaction data with each other, other options exist such as using privacy-preserving collaborations or federated learning; however, these alternative methods will require additional communication, governance and security measures.

4. RESULTS AND DISCUSSION

4.1. Analytical Assessment of the Proposed Framework

Since this study creates a framework and does not report on a finished experiment, the findings will be reported as an analysis of what can be expected from it, how it can be measured for success, and the compromises involved. Without experimental proof, there is no claim made regarding numerical increases in fraud detection, latency, or explanation quality.

The suggested architecture overcomes a major limitation of transaction-level classification by including the relationships between the financial units. Typically, in conventional classification, a single transaction is analyzed based on its amount, place, time, and other features. However, a graph-based model also evaluates the surrounding network, which enables it to find patterns such as coordinated transactions, unusual connections, and suspicious relationships between accounts. The

advantages offered by this approach are consistent with the general conclusions drawn from studies of graph anomaly detection by Akoglu et al. (2015) and financial fraud as reviewed by Motie and Raahemi (2024).

Graph learning should not automatically be assumed to outperform traditional methods in all situations. When the financial dataset has limited useful relational information, then a properly calibrated Gradient Boosted Tree (GBT) or Logistic Regression (LR) model will typically be more computationally efficient, as well as easier to validate. Also, if there is an incomplete or noisy graph construction, this could potentially create false connections which would negatively impact the performance of the model. Thus, the practical benefits of graph learning depend on the quality of the relational information, the type of anomalies present, and the conditions under which models are evaluated.

The comparison that is most relevant does not only compare whether the graph model achieves a better total aggregate score. Rather, the comparison must determine if using graph information improves the ability to detect at a specific false positive rate; if it improves the precision of the top ranked alerts; and if it can identify previously unidentified coordinated anomalies. These outcomes will be much more persuasive of its operational value than a slight increase in the overall accuracy of the model on an imbalanced dataset.

4.2. Comparative Analysis of Detection Approaches

Table 1: Comparison of Anomaly Detection Approaches in Distributed Financial Systems

Approach	Primary Strength	Principal Limitation	Expected Role
Rule-based detection	Transparent and straightforward to audit	Fixed rules may miss evolving patterns	Immediate controls and known-risk indicators
Conventional machine learning	Efficient classification of structured attributes	Limited relational awareness unless graph features are engineered	Strong baseline for transaction-level detection
Graph neural networks	Learns entity relationships and neighborhood patterns	Computational cost and limited inherent interpretability	Detection of relational and coordinated anomalies
Explainable graph learning	Combines graph-based prediction with inspectable evidence	Explanation fidelity and stability require validation	Risk ranking and investigation support
Hybrid graph and rule-based system	Combines known-risk controls with learned relational patterns	More complex calibration and maintenance	Integrated financial monitoring

Note: The comparison is conceptual and summarizes the expected characteristics of each approach. It does not represent results from a controlled benchmark experiment.

Table 1 illustrates that the proposed model is most effective when used in conjunction with existing financial monitoring techniques, and should not be viewed as an overall alternative. Traditional rule-based models continue to serve well as deterministic controls, while traditional

machine learning continues to establish a baseline of performance. Graph neural models allow for additional information to be represented through relationship mapping, and models which include explanations provide greater opportunity to assess the results of such models.

The Hybrid Architecture is of special importance due to the need for Financial Institutions to preserve existing Rules and Reporting Procedures, while providing New Analytical Capabilities. A Graph Model will Prioritize Complex Cases, while Existing Controls will Handle Explicitly Defined Risk Conditions. The Hybrid Architecture provides Layered Detection without having Each Decision Based upon a Single Complex Model.

4.3. Explainability and Investigation Effectiveness

The primary anticipated advantage of explaining how graph-based anomaly detection works is to convert a high anomaly rating into an investigation lead based upon evidence. While simply reporting that an account is suspect provides little direction for an investigator, providing an explanation which identifies a sudden surge in transaction volume, multiple new connections with other entities, and an unusually high concentration of activities will provide the investigator with a means to focus their efforts.

When you think an abnormality is due to more than one thing, it can make things very difficult for you as a investigator. That's why having subgraphs in your explanation will allow you to go through the data faster. However, just because you have a small explanation does not mean it is going to help you find out what went wrong. If there are key pieces of information missing from your explanation then it may actually hurt your ability to understand how something happened. Therefore, both the size of the explanation and how accurate it was compared to the way the original model predicted what would happen should be considered.

In addition to providing explanations based on subgraphs, feature attributions can provide insight into the importance of transaction amounts, frequencies, accounts histories, and deviations from normal temporal patterns. However, feature attributions cannot be used as proof that a specific feature was responsible for causing the anomaly. Model explanations help to illustrate how predictions relate to the input data used in the model under a given explanation methodology; however, model explanations cannot be used independently to establish criminal intent.

Another characteristic that must be met is the consistency of explanations. Minor changes in the input graph should not generate drastically different explanations for no logical reasons. If there are two very similar networks of transactions which generate opposing explanations, then investigators will likely lose confidence in the reliability of the system. An assessment of whether explanations are consistent after being tested multiple times under realistic perturbations can provide information regarding the stability of explanations.

4.4. Evaluation Metrics and Expected Trade-Offs

Table 2: Proposed Evaluation Dimensions and Measurement Criteria

Evaluation Dimension	Recommended Measures	Research Purpose
Detection performance	Precision, recall, F1-score, AUPRC	Assess identification of rare anomalies
Alert prioritization	Precision at fixed review capacity; recall at fixed alert volume	Measure usefulness to investigators
Explanation fidelity	Prediction agreement under explanation-based masking	Determine whether cited evidence reflects model behavior
Explanation stability	Similarity of explanations under controlled perturbations	Assess consistency
Explanation compactness	Number of highlighted nodes, edges, and features	Estimate investigative complexity
Operational efficiency	Throughput, median latency, 95th-percentile latency	Assess real-time suitability
Distributed scalability	Resource utilization, communication overhead, throughput under load	Evaluate scaling behavior
Robustness	Performance under drift, missing edges, delayed events, and noisy attributes	Assess reliability under imperfect conditions
Governance	Audit completeness, evidence traceability, access-control compliance	Assess operational accountability

Note: These measures define a proposed experimental evaluation plan. Their numerical values must be obtained from implementation and testing.

The Area Under Precision-Recall Curve (AUPRC) metric is especially useful for evaluating models that are used to identify transactions that are either fraudulent or anomalous. In cases where there are very few instances of such transactions, then the use of accuracy can be deceiving. For example, if a model identifies nearly all transactions as being legitimate, it will likely have high accuracy; however, it will also miss most of the potentially problematic events. Therefore, both Precision and Recall must be evaluated together, and the operational threshold must take into account the ability to investigate potential anomalies and the costs associated with both false alerts and missing anomalies.

The measurement of latency should occur at the end of each stage of the processing pipeline, not just at the end of the neural network inference step. There are many other steps in the process, such as data ingestion, graph updates, retrieving neighbors, creating explanations, and communicating with other services that can contribute significantly to the total response time. Reporting both the median and 95th-percentile latencies allows for a much better understanding of how a system performs compared to only reporting the average latency.

The evaluation of explanations requires special attention. While prediction agreement following the use of an explanation-based mask may provide insight into whether the highlighted

evidence is related to the model's output, the results will depend on the methods used to replace the removed nodes, edges, or features. As such, masking experiments need to incorporate appropriate control conditions, as well as being supplemented by assessments of stability and structured evaluations conducted by domain experts.

4.5. Proposed Experimental Comparison

The four primary types of systems to be compared through this experimental evaluation are a rule-based detection system, a traditional machine-learning based classifier, a non-explanatory graph neural network, and the proposed explainable graph-learning model. Each system must be evaluated over the same timeline of test data, with similar levels of available information.

A public financial transaction benchmark (e.g., the IEEE-CIS Fraud Detection dataset) can provide a starting point for evaluating transactions at the transaction level. Researchers should verify all relevant relational fields within the transaction dataset prior to developing a graph structure. While a transaction dataset can provide information on transaction-to-transaction relationships, it does not always provide full coverage of account-to-account relationships. Controlled experiments utilizing synthetic graphs are possible as long as researchers make sure to distinguish the findings of these experiments from the performance of the system in the real world.

In the experiment, many types of anomalies can be tested (e.g., higher than usual number of transactions, coordinated money transfers, new and/or suspicious relationships being created, and changes in account activity over time). Unsupervised anomaly detection and expert review of potential cases will provide additional means for evaluating performance when labeled data is not available.

To test the effectiveness of each architectural feature, it is recommended to conduct "ablation studies" by removing one type of temporal information, heterogeneous edge types, explanation generation mechanisms, or graph neighborhood aggregation methods at a time. Through this testing process, we can find out if any of these architectural elements provide measurable benefits. In addition, robust testing will include controlled missing edges, delayed events, and variations in anomaly prevalence.

4.6. Operational Interpretation and Limitations

Figure 2 shows how to turn your model's output into a step-by-step investigation procedure. The first stage of this procedure identifies what you have found (graph evidence) and when it occurred (temporal indications). Next, a quality check will determine if there is enough supporting evidence for your explanation. If there isn't enough evidence, the system should tell you that you don't have enough information to make an accurate statement, rather than tell you that you are sure about something that may not be true.

The new approach also has several disadvantages. If there are many connections among the accounts, then large financial graphs will have high memory and network bandwidth usage, especially if the accounts are linked to each other through different financial institutions. Sampling from the neighborhood can reduce resource utilization, however it could lead to missing some important relationships. Additionally, temporal graph models will require more memory and communication overhead. Lastly, explanation generation will result in higher inference times.



Fig 2 - Evidence-to-Investigation Cycle

A further limitation is the availability of reliable labels. Labels of confirmed fraud may be late, incomplete, or influenced by current investigation priorities. As a result, evaluating the model using only historical labels can lead to underestimating new types of suspicious activities. Therefore, the model needs to differentiate between confirmed outcomes, unresolved alerts, and analyst suspicions.

Finally, how good an explanation is depends on how the model relate to the explanation method. A plausible explanation to a human investigator might not accurately represent the model's logic. On the other hand, a mathematically accurate explanation might be hard to understand by a non-technical investigator. Both of these characteristics need to be evaluated directly.

5. CONCLUSION

Explainable graph learning is an alternative to traditional methods for detecting distributed financial system anomalies. Using a heterogeneous temporal graph representation of the entities within the distributed system (and how those entities interact with each other), graph neural networks have the ability to utilize relational evidence from the graph in addition to traditional attribute-based data. The value of this approach lies in its ability to detect anomalous patterns of behavior such as coordinated fraud schemes, suspicious fund transfer activity, and unusual entity relationships, which are created by interactions between different entities.

This study developed an integrated system for integrating multiple aspects of a financial fraud detection process: distributed event collection; temporal graph development; graph representation learning; anomaly scoring; explanation creation; and investigator feedback. This system was designed to link the decision making process (i.e., whether to investigate) to supporting

evidence that can be examined by investigators (i.e., relevant subgraphs, transaction characteristics, and temporal deviation). In addition, this study created a methodology to assess the performance of the system based on six criteria: quality of detection; fidelity of explanation; stability; computational efficiency; distributed scalability; and governance.

Explainable graph learning is an alternative method to traditional monitoring practices; however, it does not mean that the use of traditional methods will be completely replaced. Traditional rule based controls are still useful in identifying known risk areas and traditional classification algorithms can establish a baseline. Graph based methods can add value when the relationship between data points adds to the ability to identify anomalies. As such, the usefulness of graph based methods has to be demonstrated through a series of experiments with relevant datasets, under actual deployment environments, and with clear metrics for evaluation.

One of the most important aspects of the proposed framework is that it focuses on providing explanation quality and distributed system performance as primary design considerations. If an anomaly detection system produces accurate scores, but lacks the ability to produce reliable evidence, then it would be difficult to deploy the system in a regulatory compliant manner. Likewise, if there is a model which is interpretable, however, does not meet the necessary operational latency or scalability requirements, then it would likely not be able to be deployed into production.

The framework provides a base on which to implement and study the results empirically. Ultimately, the success of the framework should be based upon how well it detects anomalies and explains them, how quickly it can prioritize alerts, and how much assistance it provides in supporting human accountability during decision-making.

6. FUTURE SCOPE

Future research should validate this proposed architecture with actual financial data (or financial data that is managed in an appropriate manner). This future research should assess how well graph-based approaches perform relative to existing conventional baseline methods, as well as provide information about the level of certainty in the findings, how well the model can generalize over time, and how well the model can detect anomalies as the frequency of those anomalies changes. Additionally, evaluating the model across several different institutions and transaction types would assist in determining if the framework applies beyond a single data set.

A second direction is developing temporal and heterogeneous graph neural networks which will allow for a more detailed representation of the sequence of financial interactions. It may be possible to combine event-time processing with dynamic graph updates to enhance responsiveness to emerging patterns. However, the resulting computational overhead will need to be quantified accurately.

In addition to the aforementioned research areas, investigators should also examine how to explain graphs in a way that is tailored to the unique characteristics of financial graphs. Some of the most promising avenues for further study involve counterfactual explanations, temporal subgraph explanations, uncertainty-aware evidence summaries, and the stability of explanations as graphs undergo changes. When constructing counterfactual explanations, researchers must ensure they are based upon realistic financial behaviors, and not merely as proof that changing a particular feature within the graph will result in a decrease in fraudulent activity.

Fourth, the potential for privacy-preserving graph learning should be examined. Federated learning, secure computing, and privacy-aware graph representation methods could facilitate collaborative research efforts among organizations while protecting the privacy of individual transactional data. The feasibility of these approaches would depend upon communication expenses, levels of privacy provided, and the degree to which cross-institutional relationships can be maintained.

Finally, additional research regarding potential future deployments should focus on investigator responses to explanations, fairness of the system, resilience to adversarial attacks, changes in the underlying model over time (model drift), and the ability to audit the system. Evaluation based upon human-centric factors may assess if providing explanations increases efficiency in conducting investigations, improves the prioritization of alerts, and provides more consistent decision making among analysts. These types of evaluations will aid in determining if the use of explanation technology results in actual operational advantages over technical measures of interpretability.

REFERENCES

- [1] Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29, 626–688. <https://doi.org/10.1007/s10618-014-0365-y>
- [2] Obannagari, C. K. R. N., & Nangi, P. R. (2021). Predictive Database Performance Optimization Using Machine Learning-Driven Query Workload Modeling. *International Journal of AI, BigData, Computational and Management Studies*, 2(4), 107-116.
- [3] Visweswaran, K. (2026, March). Cross-Layer AI Optimisation Techniques for Thermal Management in Wearable Intelligent Devices. In *2026 International Conference on Machine Learning and Autonomous Systems (ICMLAS)* (pp. 1557-1564). IEEE. <https://doi.org/10.1109/ICMLAS67792.2026.11483953>
- [4] Nangi, P. R., Obannagari, C. K. R. N., & Settipi, S. (2022). Self-auditing deep learning pipelines for automated compliance validation with explainability, traceability, and regulatory assurance. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(1), 133-142.
- [5] Cheng, D., Zou, Y., Xiang, S., & Jiang, C. (2025). Graph neural networks for financial fraud detection: A review. *Frontiers of Computer Science*, 19(9), Article 199609. <https://doi.org/10.1007/s11704-024-40474-y>
- [6] Virendra Jangid. (2024). Cloud-Native Financial Data Platforms for Risk and Compliance Management. *International Journal of Intelligent Systems and Applications in Engineering*, 12(23s), 4389 -. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/8349>
- [7] Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems* 30 (pp. 1024–1034). Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7e00-Abstract.html>

- [8] Nangi, P. R., Obannagari, C. K. R. N., & Settipi, S. (2025). Large Language Model-Driven Intelligent Observability Frameworks for Serverless Applications in Event-Driven Cloud Architectures. *International Journal of Emerging Research in Engineering and Technology*, 6(1), 100-110.
- [9] Vasantha Kumar Hosahalli Seenappa. (2024). NG-iRTS: A Python and Ansible-Based Infrastructure as Code Framework for Enterprise Network Automation. *International Journal of Intelligent Systems and Applications in Engineering*, 12(23s), 4486–4492. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/8533>
- [10] Gantikota, S. (2025). Privacy-By-Design Engineering Under GDPR and CCPA: Practical Patterns for Cross-Border Data Handling In Cloud-Based Applications. *International Journal of AI, BigData, Computational and Management Studies*, 6(1), 227-231. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I1P123>
- [11] Nangi, P. R., Obannagari, C. K. R. N., & Settipi, S. (2022). Enhanced Serverless Micro-Reactivity Model for High-Velocity Event Streams within Scalable Cloud-Native Architectures. *International Journal of Emerging Research in Engineering and Technology*, 3(3), 127-135.
- [12] Khoozani, Z. S., Seera, M., & Lim, C. P. (2026). Towards transparent financial AI: A systematic review of graph learning and explainable methods for credit risk and fraud detection. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-026-11645-z>
- [13] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1609.02907>
- [14] Sunkara, R. (2024). Improving Observability and Stability in Wayland-Based Compositors: Lifecycle Logging, Buffer Validation, and Crash Hardening in Production Display Stacks. *American International Journal of Computer Science and Technology*, 6(1), 60-64. <https://doi.org/10.63282/3117-5481/AIJCSST-V6I1P106>
- [15] Nangi, P. R., & Obannagari, R. N. CK, & Settipi, S.(2022). Predictive SQL Query Tuning Using Sequence Modeling of Query Plans for Performance Optimization. *International Journal of AI, BigData, Computational and Management Studies*, 3(2), 104-113.
- [16] Y. Syed, "Enterprise Observability and Software Delivery Performance: From Telemetry to Engineering Intelligence", *IJADSMC*, vol. 5, no. 2, pp. 01-08, Sep. 2022, doi: 10.67228/30713498/IJADSMC-2022PIIKP2M
- [17] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* 30 (pp. 4765–4774). Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>
- [18] Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*, 240, Article 122156. <https://doi.org/10.1016/j.eswa.2023.122156>
- [19] Syed, Y. (2025). Generative AI Integration and RAG-Based Enterprise Tooling: Architectures, Techniques, and Organizational Impact. *International Journal of Scientific Research in Science, Engineering and Technology*, 12(6), 600-606. DOI: <https://doi.org/10.32628/IJSRSET2613255>
- [20] Nangi, P. R. (2022). Multi-Cloud Resource Stability Forecasting Using Temporal Fusion Transformers. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(3), 123-135.
- [21] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>
- [22] Nangi, P. R., Obannagari, C. K. R. N., & Settipi, S. (2025). Quantum-Enhanced Optimization Models for Large-Scale Security Policy Evaluation in Distributed Cloud-Native Systems. *American International Journal of Computer Science and Technology*, 7(6), 46-56.
- [23] Casadei, A., Bhaduri, S., Malshe, R., Mullapudi, P., Ratan, R., Pole, A., & Rakshit, A. (2025). Scaling Causal Mediation for Complex Systems: A Framework for Root Cause Analysis. *arXiv preprint arXiv:2512.14764*.
- [24] Nangi, P. R., & Obannagari, C. K. R. N. (2024). A Multi-Layered Zero-Trust-Driven Cybersecurity Framework Integrating Deep Learning and Automated Compliance for Heterogeneous Enterprise Clouds. *American International Journal of Computer Science and Technology*, 6(4), 14-27.
- [25] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1710.10903>

- [26] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html>
- [27] Nangi, P. R., Obannagari, C. K. R. N., & Settipi, S. (2025). AI-Driven Data Engineering Pipelines for High-Velocity Analytics Using Serverless Architectures and Event-Stream Intelligence. *International Journal of AI, BigData, Computational and Management Studies*, 6(3), 74-83.
- [28] Syed, Y. (2024). DevSecOps and Enterprise-Scale CI/CD Pipeline Engineering: Governing Delivery Velocity, Embedded Security, and Release Automation Across Polyglot Environments. *International Journal of Scientific Research in Science, Engineering and Technology*, 11(5), 475-484. DOI: <https://doi.org/10.32628/IJSRSET2513156>
- [29] Obannagari, C. K. R. N., & Nangi, P. R. (2021). Predictive Autoscaling for Kubernetes Using Multivariate Time-Series Forecasting. *International Journal of Emerging Research in Engineering and Technology*, 2(2), 73-83.
- [30] Fung, W., Guo, L., Hilliard, D., Casadei, A., Ratan, R., Bhaduri, S., ... & Rakshit, A. (2025). OpComm: A Reinforcement Learning Framework for Adaptive Buffer Control in Warehouse Volume Forecasting. *arXiv preprint arXiv:2512.19738*.
- [31] Nangi, P. R., & Obannagari, C. K. R. N. (2026). Federated ai-driven query optimization for distributed cloud databases. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 7(1), 91-99.
- [32] S. Suganya, Dr. Suresh Kumar Murugesan, and Dr. K. Govind, "Correlates of School and Family Environment on WellBeing," *Journal of Survey in Fisheries Sciences*, vol. 10, no. 2(S), pp. 1714-1720, 2023.
- [33] Nangi, P. R., & Nala Obannagari, C. K. R. (2026). Quantum-Classical Hybrid Algorithms for Complex Query Planning: A Scalable Computational Framework for High-Dimensional Decision Optimization in Heterogeneous Data Ecosystems. *International Journal of AI, BigData, Computational and Management Studies*, 7(2), 328-336. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I2P143>
- [34] Brahmandam, L. M. K. (2024). An Empirical Evaluation of the Medallion Architecture on Databricks and Apache Spark with Snowflake: Throughput, Latency, and Cost for Batch and Real-Time Ingestion Patterns. *International Journal of AI, BigData, Computational and Management Studies*, 5(3), 197-206. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I3P122>
- [35] Syed, Y. (2023). Polyglot Persistence and Enterprise Data Migration: Architectures, Consistency, and Practice. *American International Journal of Computer Science and Technology*, 5(3), 61-66. <https://doi.org/10.63282/3117-5481/AIJCST-V5I3P106>
- [36] Sandra, K. (2024). *THE REGULATED BANKING AI LAKEHOUSE*. INDO-CONTINENTAL ACADEMIC PUBLISHERS.
- [37] Nangi, P. R., & Obannagari, C. K. R. N. (2026). LLM-Driven Database Administration and Natural Language Query Interfaces. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 7(1), 410-418.