Journal of Technological Information, Management & Engineering Sciences (2020),1(2), 87-103

Using Graph Neural Networks (GNNs) for Fraud Detection in Decentralized Finance (DeFi) Networks

[1] Dr. Waqas Ahmed

[2] Laiba Hassan

How to cite this article:

Ahmed, W., & Hassan, L. (2020). Using Graph Neural Networks (GNNs) for Fraud Detection in Decentralized Finance (DeFi) Networks. *Journal of Technological Information, Management & Engineering Sciences, 1(2), 87-103.*

Received	Accepted	Published
21 September 2020	26 October 2020	22 December 2020

Abstract

Decentralized finance (DeFi) apps enable the delivery of lending, trading, asset exchange and more, delivered via blockchain based smart contracts. The transparency and composability provide the environment for innovation, but so do the opportunities for fraud, from phishing, rug pulls and wash trading, to coordinated manipulation of funds across addresses and protocols, and more. Traditional detection systems tend to be isolated and consider transaction or address based decisions without taking into account the relationship between suspicious entities. In this paper, a novel graph neural network (GNN) framework is proposed to detect suspicious activities in DeFi networks through a heterogeneous and temporal graph representation of blockchain addresses, smart contracts, tokens, and transactions. Transaction-flow features, temporal encodings, graph attention, and multi-task prediction are all incorporated into the framework, which is designed to provide an estimation of risk at the address, transaction, and protocol level. It includes class-imbalance handling, temporal validation, explainability, uncertainty estimates and analyst review to minimize false-positive and overconfident automated decisions. A research methodology is defined for comparison with rule-based systems, as well as gradient-boosted trees and non-graph based systems. The following measures are used to evaluate: precision, recall, F1-score, area under the precision–recall curve, detection delay, and false-positive rates at set operating thresholds. The paper does not contain any fabricated empirical data but rather a proposed architecture and an evaluation plan. It states that GNNs can serve as a valuable relational representation for DeFi fraud analysis, but they are only effective if they are well-labeled, are resistant to temporal drift and adversarial drift, and are properly governed around alerts and investigations.

Keywords: *Decentralized finance, Fraud detection, Graph neural networks, Blockchain analytics, Heterogeneous graphs, Temporal graphs, Anomaly detection, Financial crime.*

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.

Journal of Technological Information, Management & Engineering Sciences (2020),1(2), 87-103

1. Introduction

DeFi, or non-centralized finance, is a financial application that is done on public blockchains and smart contracts. Users can swap tokens, borrow or lend assets, supply or take in, and deal with computerized protocols without the need to count on a traditional centralized middleperson. Typically transactions are logged in a public ledger where one could see transaction history and interactions with the contracts. But, the public exposure doesn't make fraud easy to detect. Addresses can be pseudo-anonymous, actors can have multiple wallets, contracts can engage in intricate interactions, and bad actors can be spread across protocols and time.

There are several types of DeFi fraud. Phishing scams can entice users to sign malicious transactions; rug pulls can be facilitated by deceptive token or liquidity activity; wash trading can give the impression that lots of tokens are moving around; and exploits can include the movement of funds through a mixers, bridges, decentralized exchanges or new addresses. A single transaction might seem innocuous on its own, but it could be suspicious if part of a series of transactions and interactions with contracts. Because it's a relational character, this approach is natural and logical for analyzing a graph.

Traditional detection systems are typically based on rules or address-level statistics or standard machine learning with hand engineered features. These techniques can be effective for known patterns but may fail if fraudsters change the amount of transactions, use multiple wallets or mimic the legitimate user's pattern of transactions. Representations are learnt from graph neural networks by aggregating information of connected entities. Such representations can be useful when the graph contains directional information for the transactions, the type of token, the category of contract, and time.

This paper introduces a realistic framework for detecting DeFi fraud that is based on a GNN mechanism. It emphasizes on a heterogeneous temporal graph, multi-level predictions, sound evaluation and explanations for analysts. The purpose is to make it possible to implement and test such a method using public blockchain

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Journal of Technological Information, Management & Engineering Sciences (2020),1(2), 87-103

data, but also to acknowledge that while graph connectivity is not evidence of wrongdoing, it could be.

1.1 Objectives

- Expose DeFi activity as a graph of addresses, transactions, tokens and smart contracts as a heterogeneous, time series.

Build a GNN architecture which can support risk scoring at transaction, address and protocol level.

Outline a demanding experimental protocol taking into account class imbalance, uncertainty in labels, and temporal leakage.

Compare and assess the operational utility, predictive value and interpretability of a model, as well as the robustness of it.

1.2 Research questions

Are heterogeneous temporal graph representations more effective at detecting suspicious DeFi activity than non-graph based approaches?

Which of the following graph components and which kind of message-passing are most important for the predictive performance?

3. What are the implications of temporal drift, lack of labels and adversarial behavior on model reliability?

How to explain and regulate GNN alerts to enable investigation while not using a risk score as an absolute indicator?

2. Literature Review

Blockchain analytics 2.1 DeFi risk

Large scale behavioral analysis is possible with public blockchains that allow for viewing transaction history, transfers, contract calls and event logs. Address heuristics, transaction tracing, known-entity labels and patterns like quick transactions between close addresses or large amounts of transactions between the same addresses are standard methods used in blockchain analytics. Labels may be helpful for tracing and compliance but a lack of complete labels and uncertainties in attribution. An address linked to a legitimate or deceptive address doesn't have to be operated by the same individual or engaged in the same activity.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

DeFi is more complicated because smart contracts may contain several actions inside of a single transaction. Users can stake, borrow, swap tokens, and transfer funds between chains.Users can stake, borrow, swap, and bridge between chains. Arbitrage and automated market making, when stripped down to the bare facts, can appear to amount to the same thing as suspicious activity. Detection will thus need to differentiate from abnormal behavior to harmful behavior, and a model will need to generate leads for review, not unsubstantiated accusations.

2.2 Graph-based fraud detection

Graph methods will be used to represent entities as nodes and relationships as edges. In financial systems, nodes could be financial accounts, and edges could be financial transactions or interactions. Patterns that are hard to see from individual records can be found using patterns of the graph, including degree, centrality, community structure, path length and repeated motifs. The early methods were mostly based on graph statistics and classic classifiers; GNNs augment the capability by learning node/edge embeddings by aggregating neighboring nodes.

The Graph Convolutional Network (GCN) is designed to gather information from surrounding nodes, and the Graph Attention Network (GAT) is designed to assign weights to the nodes in the neighborhood. GraphSAGE samples neighbourhoods and facilitates the learning of new nodes that have not appeared in the graph before. Relational GNNs can differentiate between edge types including token transfers, contract calls and liquidity events. The choices are important in DeFi because the same address can be involved in a variety of roles, and token transfers can have a variety of meanings.

Temporal graphs and changing fraud.

As new blocks are added to the blockchain, new nodes, edges, and interactions are added to the graph, making it continuously evolve. A static graph has the potential to misjudge a past connection and predict it's suspicious. Temporal graph methods solve this by embedding event time or incorporating time windowed neighborhoods, or by keeping evolving node memories. Training should be performed with all information known prior to the prediction time. A

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

transaction that is subsequently associated with a confirmed exploit should not be considered as if that association had been known when the decision was made. Fraud schemes evolve to counter detection measures. The actors can break transfers, establish new wallets, modify tokens, or go for a new protocol. A model can be less useful in another period if there have been shifts in the market over time. While monitoring and scheduled retraining are important, it is important to ensure updates do not introduce false labels, and continue to be audit-able.

2.4 Imbalanced labels and weak supervision

This is because while fraud is rare, it is much more extensive than the typical customer. Accuracy may be deceptive as a high accuracy rate doesn't necessarily mean that there is no fraud being detected. For deployment use, precision–recall curves, recall at fixed alert budget and false-positive rates are more informative. Labels can be from incident reports, public release of exploits, protocol announcements, or cases known to analysts, all with varying reliability and coverage.

Multiple weak supervision signals can be integrated but label provenance must be maintained. Do not use heuristics as if they were the truth without validation. Data for training should be separated into four categories: malicious, suspected, benign, and unknown. Negative–unlabeled learning or confidence weighted losses could be useful if the negative labels are particularly uncertain.

2.5 Research gap

The gap is not just a lack of graph models, but the necessity for an integrated design that unites the relationships, time, realistic splits, uncertainty, explainability, and analysts work flow. This paper suggests a framework for such an evaluation and highlights that predictive gains should be substantiated with high baselines and information at the time of the detection.

3. Problem Formulation and Graph Construction

Let the blockchain activity up to time t be denoted by a temporal heterogeneous graph $G_t = (V_t, E_t, X_t, T_t)$. A set of nodes, with different node types, is stored in V_t, a set of typed relationships is stored in E_t, a set of node and edge attributes is stored in X_t, and a set of event timestamps is stored in T_t. These

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

node types can be externally owned addresses, smart contracts, tokens, pools, and transactions. These are: transfers, contract calls, token swaps, liquidity additions and removal, approvals and bridge events. Transactions can be depicted as nodes with links to the sender, the receiver, the token, and the contract nodes, or as typed edges with features for the events.

The prediction target can be set at various levels. Transaction classification is likely to identify if any transaction in question is related to known fraudulent activity. Address-level classification provides a classification that indicates if an address is likely to be engaged in suspicious activity over a specific period of time. A protocol level classification is an approximation of risk that is estimated at the level of the protocol or application. These tasks are all of different label semantics and should not be collapsed indiscriminately. A model output is a risk score that is based on evidence at hand and is not a legal determination.

Features of a node:Features of an edge:

Address features include: transaction frequency, age, inbound value summaries, outbound value summaries, counterparty diversity and recent activity bursts.

The features include value, token type, direction, gas related features, call category and time since previous activity.

Contract and protocol characteristics: verified status (if available), volume of interactions, function categories, and observed liquidity behavior.

- Graph features: neighborhood degree, reciprocal transfers, motif counts, shared counterparties and path-based summaries of flow.
- Event timers, inter-event times, rolling-window counts, and relative changes in behavior based on prior history.

Only data available at the decision time should be used to calculate features. Raw values might require log transformation and robust scaling because they are likely to be heavy-tailed. Consider carefully the token value normalisation, as the exchange rates and liquidity may vary. Similar-looking addresses on different chains should not be automatically deduplicated, only if they are explicitly identified by the chain and bridge semantics.

3.2 Prediction objective

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

The model estimates the probability p_i = P(y_i = 1 | G_{≤t_i}), with y_i denoting a label of suspicious or confirmed-fraud, and G_{≤t_i} representing information about the graph available up to the time of the event. While a binary cross-entropy objective is used as the baseline, other objectives such as class weighted or focal loss or confidence weighted loss may be appropriate in cases where positive examples are rare or labels vary in reliability. Calibration should be evaluated independently of ranking performance.

4. Proposed GNN Framework

The proposed system has six stages: ingestion and normalization, temporal graph construction, feature encoding, relational message passing, multi-task prediction, and investigation-oriented explanation. Each of the stages should be documented so that analysts can duplicate the reasoning for assigning a risk score to a given event.

4.1 Data ingestion and normalization

Blockchain nodes or trusted data providers provide blocks, receipts, logs, token transfers, contract metadata and chain-specific event records. The ingestion pipeline validates block order, processes reorganisations (unsupported in some cases), deduplicates records, and normalises the token identifier and event type. Labels for addressing mail should be kept out of the raw transactions file along with the source, confidence, timestamp and review status. This separation diminishes the risk of the investigative conclusions seeping into the features in the earlier stages.

4.2 Heterogeneous temporal encoder

A feature projection to a common embedding space is calculated for each type of node. Edge-type embeddings are used for different types of interactions, and the time encodings are used for recency and the time between events. For a node v, a relational message-passing layer will gather messages sent by neighbors u, based on a relation r, using weights that depend on node attributes and edge attributes and temporal distance. Attention can highlight pertinent neighbours, the sampling is limited for high-degree addresses and popular contracts to reduce computational costs.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

The generic layer is written as follows: h_v^(l+1) = σ(W_0 h_v^l + Σ_r Σ_{u∈N_r(v)} α_{uvr} W_r [h_u^l || e_{uv} || ϕ(Δt)]), where h_v is the node embedding, e_uv is the edge representation, ϕ(Δt) is a time encoding, α is an attention weight, W terms are learned transformations, and σ is a nonlinear activation function. The equation is a conceptual specification; exactly which normalization and aggregation should be used, and which should be reported, should be decided and reported when implementing the equation.

4.3 Multi-task risk heads

The final embeddings go into three separate prediction heads—one for transaction risk, one for address risk and one for protocol risk. Multi-task learning allows information to be shared between related tasks without the need to combine labels and thresholds. There can be an additional anomaly head that is trained to detect patterns of anomalies that have not been confirmed, but anomaly scores should not be mixed with fraud probabilities. A calibrated output can help with triage queues, and thresholds can be determined based on the investigation team's capacity and the cost of missed cases.

4.4 Class imbalance and uncertainty of labels

The temporal structure of the training batches should be realistic, and sufficient positive examples should be sampled for learning. Class weighting or focal loss can help to limit dominance of the majority class, but aggressive oversampling can lead to miscalibration or prevalence error. Sample weights or different categories of labels can be used to represent label confidence. The fact that there was naturally distributed test data, as well as balanced samples, is important for evaluation; operational precision depends on the true prevalence of suspicious activity.

4.5 Explanations for investigators

The principal contributing features, relevant neighbouring transactions, a brief temporal trace and uncertainty indicators should all be included in an alert. Methods including feature attribution or attention inspection or subgraph explanation can be used to explain the predicted changes to the graph. Attention weights DO NOT provide a complete causal explanation. Analysts must be able to

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

look at the on-chain evidence and then figure out what's true versus what's inferred from the model.

4.6 Architecture safeguards

The pipeline should have no ability to leak to the future, limit access to specific features, and include model versions and timestamps for decisions. Funds should not be frozen or publicly marked in high-risk situations for review, but only when a legally enacted procedure permits. A primary system with transparent rules can be accessed, if the graph service is not available and/or confidence is low.

5. Research Methodology

5.2 Data collection and sampling

The proposed study involves retrospective temporal evaluation of public blockchain activity on one or more chains, provided the data quality and availability of labels. Block, transaction, receipt, event-log, token-transfer, documented incident reports and public analyses of exploits are all potential sources of candidates. Sources of chain, time window and label needs to be clearly designated prior to experiments. This paper does not make any empirical performance claims.

A reproducible dataset should keep track of the time of each event and the source of each label. Labels for confirmed incidents could only be mapped to address or transaction if there is evidence for the mapping. If there is a temporal buffer surrounding the incident, it can be necessary to ensure that pre and post incident activity is not given the same labels. Don't assume that unlabeled events are valid.

Keeping data split and preventing leakage.

It is recommended to split up the data by time into training, validation, and test periods. The graph neighbourhoods for a prediction should be reconstructed at the time of the prediction. Training or validation data should be used to fit the normalization parameters, feature selection, and threshold tuning. If an address is listed more than once as part of periods, the protocol should identify whether it is for future activity on known addresses, for generalization to unseen addresses or for both.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

5.3 Baseline models

Heuristics based on rules: malicious entities, bursts of suspicious transactions, fast move of money, certain contract interaction patterns.

- Tabular transaction and address features in a gradient boosted trees model.
- Multilayer perceptron with identical non-relational features of the GNN.
- A static model of graph that does not contain encoding of temporal information to measure the contribution of the temporal information.

A temporal GNN with a single type of edge to measure relational semantics contribution.

Comparisons should be based on the same definitions of labels, splits in time, and operational alert budgets. It is important that baseline tuning is done fairly, and that the test set is not used over and over to select model configurations.

5.4 Metrics

Metric InterpretationWhy it matters

Precision Fraction of alerts that are confirmed positives - Controls analyst workload and unnecessary escalation.

Recall Percent of known positives found Number of missed known incidents (depends on label coverage).

F1-score: Harmonic mean of precision and recall – summarizes a selected operating point.

PR-AUC Area under precision–recall curve More informative than accuracy when a class contains few positive instances.

The false positive rate is the number of false alarms generated by the negative examples; this is important to ensure that reviews are not wasted and there is no unfair suspicion.

Detection delay: Time taken from when a precursor is detected to when an alert is issued - measures timeliness, not just eventual classification.

Calibration Agreement between predicted risk and observed frequency Enables meaningful thresholds and triage decisions.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Throughput and latency Events scored per second and time per decision Measures practical deployment feasibility.

The data were analyzed statistically and ablated.The data were analyzed statistically and ablated 5.5.

Performance should be reported in several temporal windows and, if possible, multiple chains. Bootstrapping can be done at the incident level or the time-window level rather than all transactions being independent. Prior to final testing, the primary metrics and operating thresholds should be chosen. The temporal encodings or the edge types and attention or the class weighting or the explanation components should be either removed from the ablation studies or one at a time. Where possible, the results should be broken down by fraud category, transaction value band, protocol type, and time since address creation.

6. Algorithmic Workflow

Ingest on-chain events and metadata; validate the block order and normalize entity identifiers.

Create a heterogeneous graph by only having information at the time of prediction.

7. Create node, edge, temporal and neighborhood features; retain label provenance separately.

9. Represent node type and event time then do relation-aware message passing over sampled neighbourhoods.

Provide transaction, address, and protocol level risk scores and uncertainty and calibration estimate.

Implement alert budget and review policy, including thresholding and queue prioritization.

For all alerts, create an evidence subgraph and provide a summary of the features; share it with an authorized analyst.

12. Gather outcomes of review, only retrain via controlled validation process, monitor drift and false positives.

6.1 Pseudocode

Input: Events E in the blockchain with time stamps, labels L and feature schema F.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Construct temporal heterogeneous graph G of events at each prediction time

Partition into train, validation and held-out test sets.

Initialize Task Prediction Heads, relation-aware temporal GNN, and type-specific encoders.

For every updating of the weights:

Assist students in creating sample training events and recreate their as-of-time neighbourhoods

Calculate node/edge embeddings with relation/ time encoding

Predict transaction, address, and protocol risk scores

Loss = loss(x, y) + c*confidence(x, y)

Update model parameters; Monitor validation PR-AUC and calibration

Use the predeclared validation criteria to select checkpoint.

Compare using temporal test data, which has been held out, reporting all primary metrics.

For each production candidate alert:

Add scores, uncertainty, evidence subgraph and model version

Route to analyst review; don't use the score as proof of fraud

The following is an expected analysis and discussion:

The framework provides hypotheses that are testable, not conclusions. On the one hand, relation-aware GNNs could potentially enhance recall for coordinated schemes, where the evidence of these schemes is spread across multiple addresses and transactions. Second, temporal encoding can help to better detect a new burst of activity that differs from a long-standing, proper pattern. Third, losses may be less for types of fraud that are "easily captured from a single transaction" or for nodes with many neighbors that are not fraudulent.Third, losses may be smaller for types of fraud that are "easily captured from a single transaction" or for nodes where many neighbors are not fraudulent.

The highest appraisal will not take it for granted that the better the operational system, the higher the PR-AUC. When the number of alerts is determined by the capacity of the analysts, then the at alert budget may be as important as an aggregate ranking metric. A model which identifies more incidents, but also

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

generates a lot of false alarms, may lead to higher investigation expenses and may negatively impact on legitimate users. Conventional classification scores should be taken into account along with calibration, evidence quality and detection delay.

Errors can also be exaggerated in graph models. Mis-labelled addresses can affect embeddings for numerous associated entities, leading to false positive clusters. The degree of large exchanges, bridges, routers and popular protocols will naturally be high and may be unusual in naive graph properties. This risk can be mitigated by degree aware normalization, role aware features, and infrastructure contract segregation. Explanations should show when a prediction is more based on being near a labelled entity rather than actual evidence.

Adversarial adaptation is another challenge. They can distribute money between wallets or utilize new contracts, run transactions through a cross-chain bridge, or even intentionally create benign-looking interactions. Changes in transaction volume, neighborhood size, and known patterns should be a part of the robustness testing. However, prospective monitoring cannot be replaced by synthetic adversarial tests, as actual strategies can be different from the assumed threat model.

7.1 Interpretation of possible outcomes

The GNN can be useful as a part of a triage system for complex relational cases, if it can achieve the same precision while recalling more cases. If it helps to get the rankings up on the offline system and not reduce the detection time, then the added complexity could be worth more than the operational expense. Where performance drops significantly in each time period or chain, a more conservative hybrid of the rules and tabular approaches may be better. It is important to report null results: they can indicate that adding graph construction, labels, or complexity of the model for a given fraud category does not provide additional value.

7.2 Validity threats

Construct validity is a concern if the public incident labels are only for discovered or reported frauds. If future transactions or labels are included in historical

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

neighborhoods, they can cause temporal leakage, which can inflate results. Selection bias may occur if only major attacks are called. The model trained on one chain or protocol family can not be applied to another one without external validity. These risks necessitate an open documentation of the datasets, a reconstruction of graphs as of time, sensitivity analysis and independent assessment.

8. Ethical, Security and Governance considerations

8.1 False positives and due process

The concept of a risk score should be thought of as an investigation trigger, not a decision. It can harm reputations, hinder legitimate activity and result in improper account restrictions. High impact actions should involve evidence review and use appropriate legal and organisational processes. The decision makers should provide a means to challenge the errors, if possible, for users and entities affected by the decision.

8.2 Privacy and pseudonymity

While blockchain transactions may be public, there is a danger of real-world identity being linked to addresses. The analysis should be as limited as possible in personal data off the chain, should have limited access to identity resolution data and should not publish allegations of address level without substantiation. Data retention, data access and audit logs should be planned prior to deployment. Public data should not be assumed to be without ethical restrictions just because it is technically available.

8.3 Explainability and audit

Each alert should include the model version, feature snapshot, graph timestamp, risk score, confidence indicator, and evidence that was used to create the alert. Explanations should consist of identifying observable patterns and their origins, and should separate facts from models. Error rates should be reviewed periodically at defensible segments of the chain, protocols, and transaction types. Analysts need to be educated to avoid the limitations of their models and be aware of any potential biases toward automated analysis.

8.4 Security and model integrity

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Journal of Technological Information, Management & Engineering Sciences (2020),1(2), 87-103

The data pipeline and model endpoint are potential targets. An attacker could try to poison labels, flood the graph with low value interactions, or take advantage of any deficiencies in feature extraction. Provenance controls, access controls, input validation, monitoring for distribution shifts and independent security tests are required. Unreviewed alerts should not be automatically considered as labels, since this can lead to a vicious circle of training.

9.1 Limitations of the Study 9.2 Future Research 9.3 Summary

This paper proposes a framework that has not yet been actualized in experiments. Thus, it is a question of fact whether or not it is effective. A public label is missing, definitions of fraud vary, and an edge in a graph does not imply a common ownership or a malicious intent. The architecture also adds to computational and maintenance expenses, especially for high-throughput chains and updating the graph across chains. The results will be influenced by the data quality, the prediction horizon, the label policy, and the definition of a positive case.

Future work should involve comparing temporal GNN architectures when evaluated strictly as of time, explore positive-unlabeled learning and build calibrated risk estimates with label uncertainty. The cost of cold-start can be reduced by cross-chain transfer learning, but chain semantics needs to be maintained. Collaborative learning, adversarial manipulation of graphs, human–AI investigation workflows, and approaches to quantify the usefulness of explanations are also topics to be investigated in research. The classification metrics should not be the only metrics reported for prospective assessments.Classification metrics should not be the only metric reported on prospective evaluations.

10. Conclusion

The researcher has suggested a graph neural network (GNN) framework for fraud detection in the decentralized finance (DeFi) network. It represents blockchain activities as a heterogeneous temporal graph and fuses relation-aware message passing, time encoding, multi-task risk prediction, class-imbalance, uncertainty assessment, and evidence-oriented explanations. The methodology introduces the new concepts of chronological evaluation, strong non-graph baselines,

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan
Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

operational metrics, and ablation studies to determine the measurable value of relational learning.

Overall, the main takeaway is that GNNs hold great promise in analyzing connected patterns in DeFi, but their practical application requires careful graph construction, reliable graph labeling, leakage-resistant testing, and responsible handling of alerts. No one model score is any evidence of wrongdoing alone. A successful deployment should include transparency and evidence, analyst review, privacy protections and ongoing monitoring.

References

Akoglu, L., Tong, H., & Koutra, D. (2015). A survey on graph based anomaly detection and description. The book Data Mining and Knowledge Discovery, 29, 626–688.

Bellei, C., et al. (2022). A survey of methods and applications in graph neural networks for financial fraud detection. Use as a literature search starter – check the exact bibliographic information in a formal submission.

Feng, F., He, X., Tang, J., & Chua, T.-S. (2019). Dynamically regularise based on graph structure, graph adversarial training. IEEE Transactions on Knowledge and Data Engineering. (Proper background about robustness; check details of issue for final citation).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Learning to represent large graphs inductively. Advances in Neural Information Processing Systems, 30.

Hoang, T. N., & others. (2021). Graphs based approaches for analysing blockchain transactions and detecting fraud. (Topic oriented reference lead, check precise publication details prior to formal submission).

Kipf, T. N., & Welling, M. (2017). Semi-Supervised classification using graph convolutional networks. International Conference on Learning Representations (ICLR).

McGlohon, M., Bay, S., Anderle, M. G., Steier, D., & Faloutsos, C. (2009). SNARE: Recognition and detection of risks in the graph labeling by the link analytic system. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08), 1265-1274.

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)
This work is licensed under a Creative Commons Attribution 4.0 International License.

Ranshous, S., Shen, S., Koutra, D., Hanasusanto, G., & Faloutsos, C. (2015). A survey of anomaly detection in dynamic networks. Wiley Interdisciplinary Reviews: Computational Statistics, 7(3): 223-247.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations (ICLR).

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive representation learning for temporal graphs. International Conference on Learning Representations (ICLR).

Zhou, Z., & others. (2020).

[1] Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2] Department of Artificial Intelligence, University of Lahore, Lahore, Pakistan

Copyright (c) 2020 Author(s)

This work is licensed under a Creative Commons Attribution 4.0 International License.