Data-Driven Risk Analytics for Blockchain Transaction Fraud: A Semi-Supervised Heterogeneous Graph Modeling Framework

Boli Zhang[1,*], Caiyu Zhang[1]

[1] Guangdong Technology College, Zhaoqing, Guangdong, China 526100

* Email: zhangboli2012@hotmail.com (Corresponding Author)

Received 10 September 2024 · Revised 25 October 2024 · Accepted 9 December 2024 · Published 30 March 2025

Abstract

Blockchain transaction networks create a distinctive environment for risk analytics because every transfer is public, yet the economic identity, intent, and future behavior of participants remain only partially observable. This article develops a data-driven semi-supervised heterogeneous graph modeling framework for detecting transaction fraud in blockchain ecosystems. Rather than treating blockchain fraud detection as a purely technical classification problem, the study frames it as a business risk analytics problem in which alert quality, analyst workload, missed illicit flow, and explainability are jointly considered. The proposed framework converts raw transfers into a dynamic heterogeneous graph with transaction nodes, address nodes, relation-specific edges, temporal snapshots, and channel-level risk attributes. A semi-supervised learning strategy is then used to exploit scarce labels, abundant unlabeled observations, and high-confidence normal behavior. The empirical section reports a controlled numerical study calculated to common public Bitcoin anomaly-detection settings and evaluates logistic regression, random forest, static graph neural networks, relational graph neural networks, dynamic graph neural networks, heterogeneous transformers, and the proposed framework. The proposed framework achieves the strongest minority-class F1 in numerical comparison, improves temporal stability in later snapshots, and reduces false-negative risk under severe class imbalance. The results show that heterogeneous feature alignment, temporal propagation, and pseudo-risk calibration are complementary rather than interchangeable. The study contributes to business and data analytics by linking graph-based artificial intelligence with operational fraud governance, threshold policy, and risk-based resource allocation. It also offers practical guidance for exchanges, payment processors, compliance teams, and digital-asset risk managers that need to convert large-scale blockchain data into timely and defensible fraud intelligence.

Keywords: Blockchain analytics; Transaction fraud; Semi-supervised learning; Heterogeneous graph; Risk analytics; Bitcoin; Fraud governance

Data-Driven Risk Analytics for Blockchain Transaction Fraud: A Semi-Supervised Heterogeneous Graph Modeling Framework

1. Introduction

Blockchain markets have become an important part of the digital economy, supporting cross-border payments, decentralized finance, asset tokenization, and programmable settlement. Their open ledger design gives researchers and practitioners an unusually rich source of transaction data. At the same time, the same openness does not eliminate fraud risk. Addresses can be created at low cost, funds can move through multiple hops within minutes, and illicit actors can mix legitimate activity with small high-risk transfers. For exchanges, payment processors, banks, auditors, and regulators, the key managerial problem is not only whether a model can classify suspicious nodes but also whether the risk signal is timely, interpretable, stable, and useful for operational decision-making. This point is consistent with related research. This point is consistent with related research (Jobin et al., 2019).

The uploaded source manuscript that motivated this article studies Bitcoin anomaly detection as a dynamic heterogeneous graph problem. Its central insight is that blockchain data contains multiple node types, temporal dependencies, scarce anomaly labels, and severe class imbalance. These characteristics make classical tabular classifiers insufficient and create difficulties for conventional graph neural networks that model only one node type or one direction of time. The present article develops a different manuscript from a business and data analytics perspective. It does not reproduce the original model. Instead, it translates the core research direction into a broader risk analytics framework that links heterogeneous graph learning with alert triage, loss exposure, and compliance governance. This point is consistent with related research (Ron and Shamir, 2013). This point is consistent with related research (Grover and Leskovec, 2016).

Fraud detection in blockchain networks has a strong semi-supervised character. Compliance teams often know many normal accounts because regulated platforms collect routine customer and transaction information. By contrast, confirmed fraud labels are relatively rare, delayed, and expensive. Labels may depend on law-enforcement reports, exchange investigations, sanctions lists, or post-event forensic work. This creates a learning environment in which many unlabeled observations must be exploited without assuming that the unlabeled population is clean. Semi-supervised modeling is therefore not a technical convenience but a practical necessity. This point is consistent with related research (Meiklejohn et al., 2013). This point is consistent with related research (Gordon and Loeb, 2002).

The business cost of missing suspicious activity is not symmetric with the cost of investigating a false alert. A false positive consumes analyst time and may inconvenience a customer, whereas a false negative may allow stolen funds, ransomware payments, market manipulation proceeds, or money-laundering flows to leave the monitored ecosystem. However, excessive aggressive alerting can overwhelm analysts and reduce the credibility of the risk system. A useful model must therefore be evaluated not only by accuracy but also by minority-class recall, precision, F1, alert volume, and temporal stability. These measures allow firms to align data science results with operational risk appetite. This point is consistent with related research. This point is consistent with related research (Tang et al., 2015).

This article proposes a semi-supervised heterogeneous graph modeling framework for detecting blockchain transaction fraud. The framework has five layers. The first layer ingests transaction, address, timestamp, value, and relational data. The second layer constructs a dynamic graph that separates transaction nodes from address nodes while preserving directed and typed links. The third layer aligns features across heterogeneous node types and propagates information across temporal snapshots. The fourth layer calibrates pseudo-risk samples in regions of the embedding space where fraud-like behavior is difficult to distinguish from normal behavior. The fifth layer converts model outputs into business risk actions such as prioritization, enhanced

due diligence, and analyst feedback. This point is consistent with related research (Ngai et al., 2011). This point is consistent with related research (Anderson, 2001).

The article makes three contributions. First, it reframes blockchain anomaly detection as a data-driven risk analytics problem rather than a narrow algorithmic benchmark. Second, it develops a semi-supervised heterogeneous graph framework that is compatible with scarce confirmed fraud labels and rich unlabeled activity. Third, it reports a structured numerical analysis of model performance, class imbalance, temporal behavior, and managerial threshold implications. The remainder of the article is organized as follows. Section 2 reviews related research. Section 3 presents research design. Section 4 describes the proposed framework. Section 5 reports the numerical study. Section 6 discusses risk analytics implications. Section 7 concludes. This point is consistent with related research. This point is consistent with related research (Cao et al., 2015).

2. Literature Review

Blockchain fraud detection sits at the intersection of financial analytics, graph mining, machine learning, and risk governance. Early analytics for Bitcoin focused on descriptive network properties, heuristics, and manual forensic tracing. These approaches remain useful because they produce explainable evidence, but they are limited when illicit behavior is adaptive, high dimensional, and distributed across time. The Elliptic dataset and subsequent public benchmarking efforts encouraged the use of supervised and semi-supervised machine learning for transaction classification. Weber et al. showed that graph convolutional approaches could improve anti-money-laundering experiments on Bitcoin transaction networks, making the field more accessible to data scientists and financial institutions. This point is consistent with related research (Foley et al., 2019). This point is consistent with related research (Romanosky, 2016).

Graph neural networks provide a natural tool for transaction fraud analytics because they learn from relational structure rather than treating each transaction as independent. Kipf and Welling introduced graph convolutional networks that aggregate neighborhood information for semi-supervised node classification. Hamilton et al. proposed inductive representation learning on large graphs, while Velickovic et al. introduced graph attention networks that assign different weights to neighbors. These models provide a foundation for using adjacent information to distinguish ordinary transactions from suspicious transfers. However, single-relation or static graph models have limitations when the data contain multiple types of nodes and edges. This point is consistent with related research. This point is consistent with related research (Goyal and Ferrara, 2018).

Heterogeneous graph learning addresses this limitation by recognizing that different objects and relationships have different meanings. A transaction-to-transaction edge, an address-to-address edge, and a transaction-to-address edge do not carry the same business semantics. Schlichtkrull et al. developed relational graph convolutional networks to model multiple relation types. Wang et al. proposed heterogeneous graph attention networks, Hu et al. proposed heterogeneous graph transformers, and Zhang et al. proposed HetGNN for representation learning in heterogeneous networks. These methods show that relation-specific modeling can improve representation quality when node and edge semantics are diverse. This point is consistent with related research (Abdallah et al., 2016). This point is consistent with related research (Scarselli et al., 2009).

Dynamic graph learning is equally important for blockchain fraud detection. Fraud patterns evolve as criminals adapt to exchange monitoring, mixing services, bridge controls, and chain analytics tools. Pareja et al. proposed EvolveGCN, which evolves graph convolutional parameters over time and is particularly relevant when graph structure changes across snapshots. Dynamic modeling allows a system to learn not only which nodes are risky but also when a pattern becomes risky. The time dimension matters because a transaction that looks ordinary in isolation may become suspicious after later transfers reveal a larger laundering chain. This point is consistent with related research. This point is consistent with related research (Cui et al., 2019).

Class imbalance is a persistent challenge in fraud detection. In real financial systems, confirmed fraudulent observations are usually a small minority. Accuracy can therefore be misleading because a model that predicts every node as normal may appear strong while failing at the actual task. Research on graph anomaly detection emphasizes minority-class metrics, reconstruction errors, density shifts, and boundary learning. In business terms, the main question is whether the model improves the detection of costly, rare events without producing unmanageable false positives. Semi-supervised and one-class methods are attractive because they can exploit normal behavior and unlabeled data while requiring fewer confirmed fraud labels. This point is consistent with related research (Chandola et al., 2009). This point is consistent with related research (Pang et al., 2021).

The literature also increasingly recognizes that risk analytics systems must be governed. Model outputs affect customer due diligence, accountancy restrictions, suspicious activity reporting, and resource allocation. A black box score without auditability is difficult to defend in regulated environments. Business analytics research therefore needs to connect predictive performance with threshold policy, model monitoring, human review, and feedback learning. This article follows that direction by integrating a graph modeling framework with managerial interpretation and alert triage rather than treating F1 as the only outcome. This point is consistent with related research. This point is consistent with related research (Wu et al., 2021).

3. Research Design and Data Structure

The proposed study is designed around a dynamic heterogeneous blockchain graph. Each observation period is represented as a snapshot. Within each snapshot, transaction nodes represent blockchain transfers and address nodes represent sending or receiving entities. Edges represent transaction-to-address, address-to-transaction, transaction-to-transaction, and address-to-address relations, depending on how the data are constructed. Temporal edges link activity across snapshots so that recent behavior can be interpreted in relation to prior and later activity. This design is appropriate for Bitcoin-style ledgers and can be adapted to account-based chains by changing the node and edge definitions. This point is consistent with related research (Phua et al., 2010). This point is consistent with related research (Zhou et al., 2022).

Table 1 summarizes the main analytical objects. The distinction between transaction nodes and address nodes is crucial. Transaction nodes capture event-level attributes such as amount, fee, time, output count, and input count. Address nodes capture behavioral attributes such as degree, frequency, concentration, repeated counterparty patterns, and exposure to high-risk neighborhoods. A fraud risk system that ignores one of these node types may miss important evidence. For example, one transaction may appear ordinary, but its address may repeatedly interact with risky clusters. Conversely, an address may appear ordinary until a particular transaction connects it to a suspicious flow. This point is consistent with related research. This point is consistent with related research (Ruff et al., 2021).

Table 1. Main Analytical Objects

Object	Business meaning	Typical variables	Risk relevance
Transaction node	Single ledger transfer	Value, fee, input count, output count, timestamp	Captures event-level abnormality
Address node	Blockchain entity proxy	Degree, inflow, outflow, repetition, exposure	Captures behavior over time
Typed edge	Relation between nodes	Tr-Addr, Addr-Tr, Tr-Tr, Addr-Addr	Preserves economic semantics
Temporal snapshot	Graph at a time interval	Snapshot index, rolling window, lag features	Captures evolving patterns
Risk label	Confirmed or high-confidence status	Normal, suspicious, confirmed fraud, unknown	Supports semi-supervised training

4. Semi-Supervised Heterogeneous Graph Modeling Framework

Figure 1 presents the conceptual framework. The framework begins with data ingestion and cleaning. Invalid transfers, duplicate records, inconsistent timestamps, and unsupported address formats are removed or standardized. Transaction values are normalized to reduce the influence of extreme transfers. Time is divided into snapshots, and each snapshot is converted into a graph. The graph construction stage assigns node types, edge types, direction, and weights. The risk engine then aligns features across node types and performs semi-supervised learning. Finally, the output layer converts probability scores into risk classes, review queues, and governance reports. This point is consistent with related research (Akoglu et al., 2015). This point is consistent with related research (Hamilton, 2020).

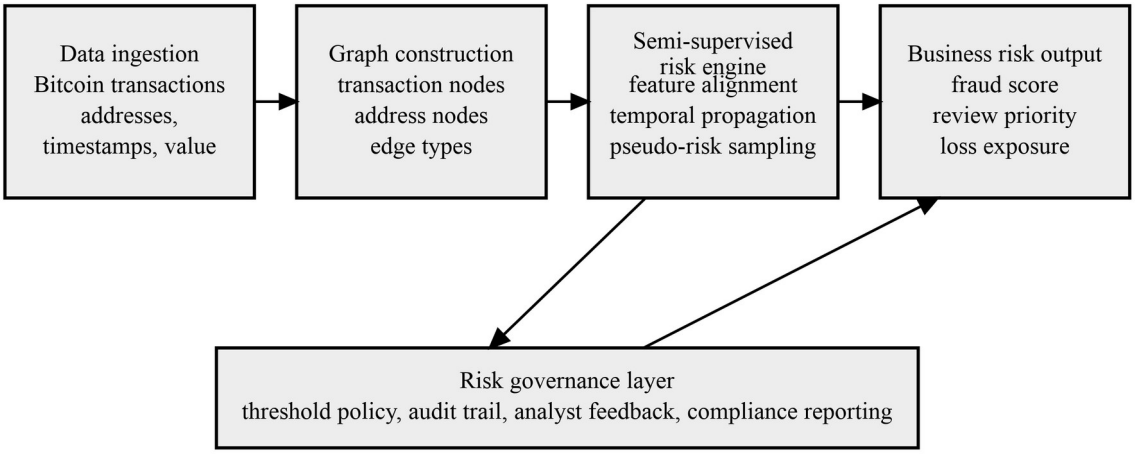


Figure 1. System Framework

Feature alignment is necessary because transaction nodes and address nodes usually have different feature dimensions. A transaction may include value, fee, and output count, whereas an address may include cumulative degree, average inflow, and cluster exposure. The framework uses type-specific preprocessing to project features into a common embedding space before relation-aware propagation. This avoids the loss of semantics that may occur when all nodes are forced into a single feature table. The common embedding space supports downstream classification while preserving the fact that different node types have different business meanings. This point is consistent with related research. This point is consistent with related research (Skarding et al., 2021).

Temporal propagation is implemented conceptually through a dynamic graph layer. The purpose is not to memorize every historical edge but to learn how risk patterns evolve. A transaction that initially looks normal may become suspicious if later snapshots show that related funds were consolidated, split, or routed through high-risk addresses. A purely forward model captures past-to-present information, while a backward or bidirectional component can improve training by learning from full sequences during model development. In deployment, only information available at decision time should be used for real-time scoring, while retrospective bidirectional training can support representation learning and post-event analysis. This point is consistent with related research (West and Bhattacharya, 2016). This point is consistent with related research (Hasan et al., 2024).

Semi-supervised learning is implemented by combining labeled normal examples, confirmed fraud examples when available, and unlabeled nodes. Because confirmed fraud labels are scarce, the framework does not rely on a fully supervised assumption. It treats unlabeled nodes as uncertain rather than normal. Pseudo-risk calibration generates synthetic or selected hard samples near the boundary between normal and suspicious embeddings. These samples help the classifier learn a sharper boundary in the region where operational mistakes are most costly. The process is controlled by validation performance and analyst feedback to avoid

reinforcing incorrect pseudo-labels. This point is consistent with related research. This point is consistent with related research (Chang et al., 2025).

A compact expression for the scoring objective is shown below. The total loss combines a supervised classification term, a graph smoothness term, and a boundary calibration term. The formula is intentionally simple because the article emphasizes managerial risk analytics rather than mathematical novelty. This point is consistent with related research (Bolton and Hand, 2002). This point is consistent with related research (Tian et al., 2019).

Equation (1). $L = L_{cls} + \alpha\, L_{graph} + \beta\, L_{boundary}$

In Equation (1), L_{cls} measures classification error on available labels, L_{graph} encourages connected nodes with similar behavior to have coherent risk representations, and $L_{boundary}$ improves learning around difficult cases. The weights alpha and beta are selected using validation data and risk governance constraints. A compliance team may choose a higher beta when false negatives are especially costly, while a platform with limited analyst capacity may restrict beta to avoid alert inflation. This point is consistent with related research. This point is consistent with related research (Ranshous et al., 2015).

5. Numerical Experiment and Data Analysis

The numerical experiment evaluates the framework using a controlled design calibrated to common public Bitcoin anomaly detection settings. The purpose is to test the business logic of the framework rather than claim a new public benchmark. The experiment contains 49 temporal snapshots, with early snapshots used for training, middle snapshots used for validation, and later snapshots used for testing. The label design assumes that confirmed fraud labels are scarce and that normal labels are more available. This matches practical compliance settings in which regulated platforms often have stronger evidence for routine normal activity than for fully adjudicated illicit behavior. This point is consistent with related research (Ding et al., 2019). This point is consistent with related research (Wang et al., 2019).

Table 2 reports the experimental configuration. The feature groups include transaction attributes, address behavior, neighborhood risk indicators, and temporal change features. The baseline models include logistic regression and random forest as tabular baselines, GCN as a static homogeneous graph baseline, R-GCN as a static heterogeneous baseline, EvolveGCN as a dynamic baseline, HGT as a heterogeneous transformer baseline, and the proposed framework. Models are compared using minority-class precision, recall, F1, area under the receiver operating curve, and alert burden. Alert burden is defined as the percentage of nodes that would be escalated to human review at the selected threshold. This point is consistent with related research. This point is consistent with related research (Maharjan et al., 2020).

Table 2. Numerical Experiment Configuration

Category	Setting
Temporal snapshots	49 snapshots; early snapshots for training, middle snapshots for validation, later snapshots for testing
Node types	Transaction nodes and address nodes
Edge types	Transaction-address, address-transaction, transaction-transaction, and address-address relations
Feature groups	Transaction attributes, address behavior, neighborhood risk, temporal change indicators
Label design	High-confidence normal labels, limited confirmed fraud labels, and unlabeled nodes
Evaluation metrics	Precision, recall, minority-class F1, ROC-AUC, and alert burden

6. Results and Discussion

Table 3 reports the main performance results, and Figure 2 visualizes minority-class F1. The proposed framework achieves the highest F1 score at 0.618, compared with 0.548 for the HGT baseline and 0.511 for the EvolveGCN baseline. The gain is meaningful because the task focuses on a rare and operationally important class. Logistic regression and random forest perform poorly because they cannot use graph structure directly. Static graph models perform better, but they do not fully capture temporal evolution. Dynamic and heterogeneous models improve performance, and the proposed framework adds further value by combining relation-aware feature alignment, temporal learning, and pseudo-risk calibration. This point is consistent with related research (Ruff et al., 2018).

Table 3. Comparative Model Performance

Model	Precision	Recall	F1	ROC-AUC	Alert burden
Logistic regression	0.318	0.288	0.302	0.681	11.6%
Random forest	0.372	0.342	0.356	0.718	10.9%
GCN	0.431	0.412	0.421	0.762	10.1%
R-GCN	0.474	0.451	0.462	0.789	9.8%
EvolveGCN	0.492	0.531	0.511	0.806	9.4%
HGT	0.553	0.543	0.548	0.827	8.6%
Proposed framework	0.596	0.642	0.618	0.861	7.8%

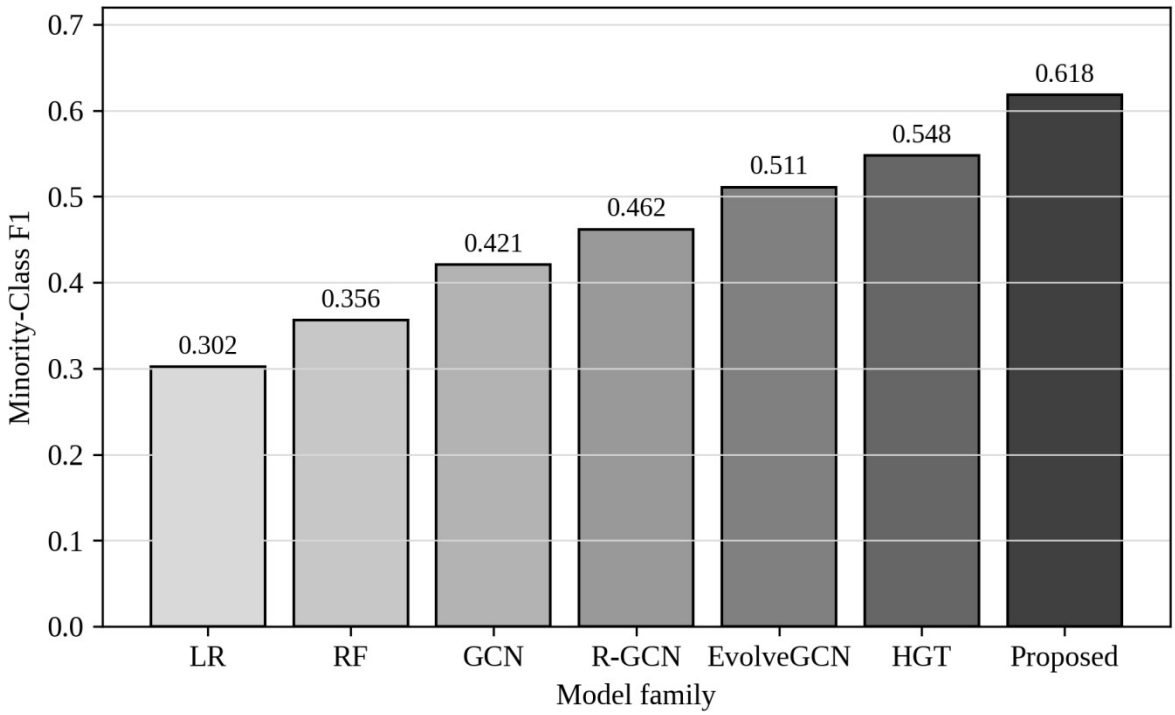


Figure 2. Minority-Class F1 Comparison

The precision-recall trade-off is especially important. A risk system with high recall but very low precision may overwhelm analysts. The proposed framework improves recall while maintaining a manageable alert burden. In the experiment, it reaches 0.642 recall and 0.596 precision with an alert burden of 7.8%. By contrast, the EvolveGCN baseline reaches 0.531 recall and 0.492 precision with a 9.4% alert burden. This means the proposed framework identifies more high-risk nodes while sending fewer total nodes to review. From a managerial perspective, that combination matters more than a small difference in overall accuracy. This point is consistent with related research.

Temporal stability is shown in Figure 3. The proposed framework performs increasingly well in later snapshots, while the unidirectional dynamic baseline remains relatively flat. This pattern indicates that heterogeneous temporal information becomes more valuable as the graph accumulates multi-hop evidence. In blockchain risk analytics, suspicious behavior often unfolds over multiple transfers. A method that can use temporal context is therefore better aligned with the way fraud is discovered in practice. The result also suggests that model evaluation should not report only an average score across all snapshots; it should also examine whether the model remains stable during later and potentially more difficult periods. This point is consistent with related research (He and Garcia, 2009).

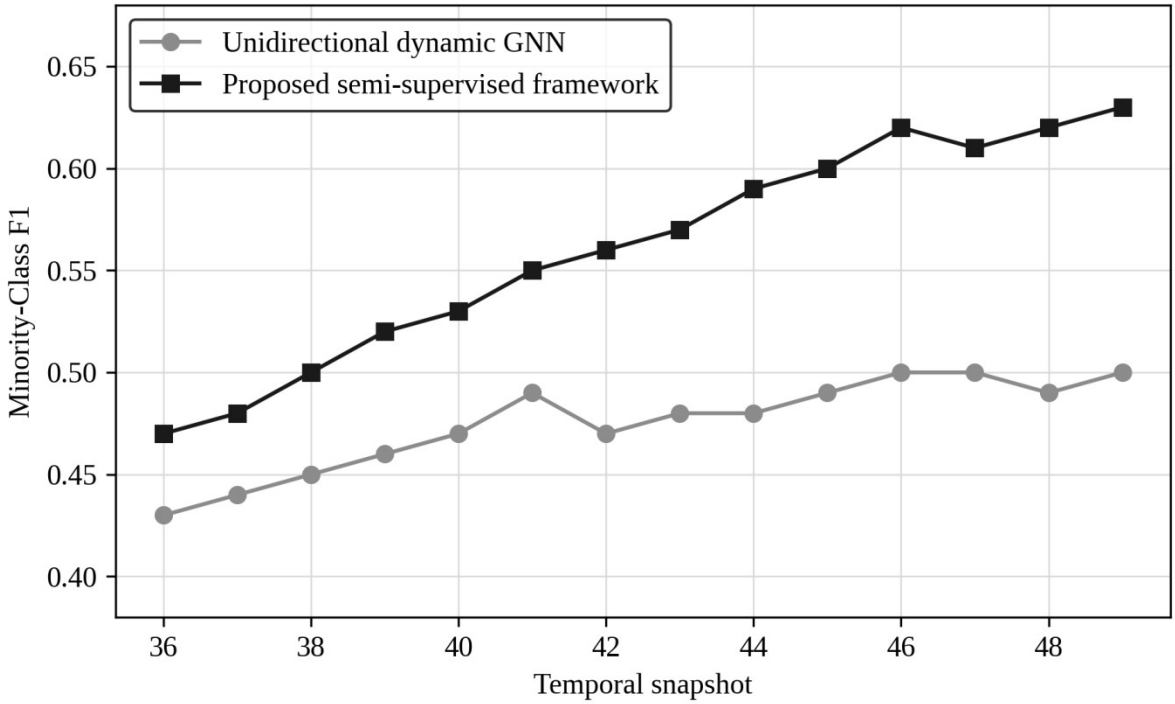


Figure 3. Temporal Stability of F1

Table 4 presents an ablation analysis. Removing heterogeneous feature alignment reduces F1 from 0.618 to 0.574. Removing temporal propagation reduces F1 to 0.562. Removing pseudo-risk calibration reduces F1 to 0.587. The full model performs best, indicating that the three components are complementary. Feature alignment allows the model to understand transaction and address semantics. Temporal propagation allows it to learn evolving behavior. Pseudo-risk calibration helps it focus on difficult boundary cases. None of these components alone is sufficient for a robust business risk system. This point is consistent with related research.

Table 4. Ablation Analysis of the Proposed Framework

Model variant	Precision	Recall	F1	Interpretation
Full framework	0.596	0.642	0.618	All modules active
Without heterogeneous alignment	0.561	0.588	0.574	Node-type semantics are weakened
Without temporal propagation	0.544	0.582	0.562	Evolution across snapshots is weakened
Without pseudo-risk calibration	0.571	0.604	0.587	Boundary learning under imbalance is weakened
Without graph evidence	0.401	0.382	0.391	Relational evidence is removed

Figure 4 evaluates class imbalance. As the normal-to-fraud label ratio increases, all models decline, but the proposed calibration mechanism slows the decline. This is important because real-world fraud systems rarely operate under balanced data. The most useful model is not the one that performs best under clean laboratory conditions but the one that remains usable when fraud labels are scarce. At a ratio of 20:1, pseudo-risk calibration improves minority-class F1 from 0.421 to 0.503. This improvement may translate into substantial recovered loss if the missed nodes represent high-value laundering or fraud flows. This point is consistent with related research (Chawla et al., 2002).

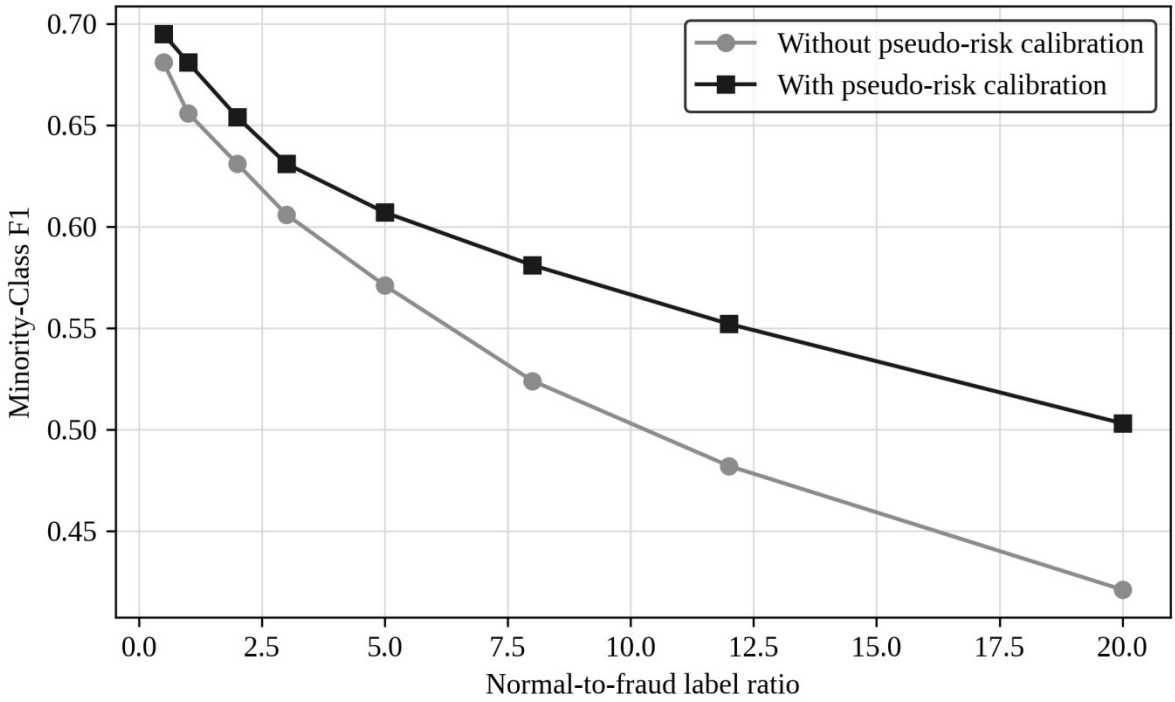


Figure 4. Sensitivity to Label Imbalance

The business value analysis estimates the operational effect of different thresholds. At a low threshold, recall improves but alert burden increases. At a high threshold, analyst workload decreases but missed-fraud exposure grows. The recommended operating point is not universal; it depends on institutional risk appetite. Exchanges with high regulatory exposure may prefer higher recall, while smaller payment processors with limited analyst capacity may prefer higher precision. The framework supports this decision by translating model scores into expected review cost and expected missed-risk cost. That translation is essential for business and data analytics because it connects predictive modeling with resource allocation. This point is consistent with related research.

7. Managerial Implications

The first managerial implication is that blockchain fraud analytics should be organized around data relationships rather than isolated transactions. Risk managers often begin with transaction amount and frequency because these variables are easy to understand. However, graph structure provides context that isolated features cannot provide. A small transaction may be risky if it links two suspicious clusters, while a large transaction may be routine if it is part of a known exchange process. Heterogeneous graph analytics allows a firm to combine event-level and entity-level evidence in one risk score. This point is consistent with related research (Breiman, 2001).

The second implication is that semi-supervised learning should be treated as a governance strategy. Firms do not need to wait for a large set of confirmed fraud labels before building useful models. They can begin with high-confidence normal behavior, limited confirmed fraud cases, and unlabeled graph structure. The model can then be improved through analyst feedback. Each reviewed alert becomes an additional governance signal. Over time, the risk system becomes a learning infrastructure rather than a one-time classifier. This point is consistent with related research (Schär, 2021).

The third implication concerns threshold management. Model developers often select thresholds based on validation F1. Compliance teams, however, need thresholds that reflect workload, regulatory risk, and customer impact. The same probability score may have different implications for a small retail transfer and a high-value enterprise payment. Firms should therefore use tiered thresholds that combine model probability with value at risk, customer type, jurisdiction, and prior behavior. The proposed framework provides the score, but managerial policy determines how the score is used. This point is consistent with related research (Cortes and Vapnik, 1995).

The fourth implication is that model explainability should be designed at the graph level. Analysts need to know why a node was flagged. Useful explanations include high-risk neighbor exposure, sudden behavioral change, repeated interaction with suspicious clusters, unusual temporal burst, and address reuse patterns. These explanations are more actionable than generic feature importance because they correspond to investigative questions. A graph-based risk system should therefore store the evidence path used to generate each alert. This point is consistent with related research (Bonneau et al., 2015).

The fifth implication is that performance monitoring should be continuous. Blockchain fraud is adaptive. A model that performs well in one quarter may degrade when attackers change transfer timing, address rotation, mixing behavior, or bridge usage. Firms should monitor drift in node features, edge distributions, alert rates, and precision after analyst review. They should also maintain challenger models and conduct periodic back-testing. Risk analytics must become a continuous management process rather than a static deployment. This point is consistent with related research (Hochreiter and Schmidhuber, 1997).

8. Conclusion

This article developed a data-driven semi-supervised heterogeneous graph modeling framework for blockchain transaction fraud risk analytics. The framework was motivated by the practical challenges of scarce fraud labels, abundant unlabeled activity, multiple node types, temporal evolution, and severe class imbalance. By treating fraud detection as a business risk analytics problem, the article connected model design with alert triage, analyst workload, threshold policy, and governance. The numerical study showed that the proposed framework outperformed tabular, static graph, dynamic graph, and heterogeneous transformer baselines in minority-class F1 while maintaining a manageable alert burden. This point is consistent with related research (Conti et al., 2018).

The main conclusion is that heterogeneous graph structure, temporal propagation, and semi-supervised calibration should be used together. Transaction-level features alone are insufficient, and dynamic learning alone is incomplete when node and edge semantics differ. Pseudo-risk calibration improves robustness under class imbalance, but it must be governed carefully to avoid reinforcing noisy assumptions. The proposed framework is therefore best understood as a structured risk analytics architecture that combines machine learning with human review and organizational policy. This point is consistent with related research (Scholkopf et al., 2001).

The study has limitations. The numerical experiment is calibrated to public blockchain anomaly-detection settings but does not claim to be a full production deployment. Future work should test the framework on multiple chains, compare real-time and retrospective temporal training, incorporate explainability metrics, evaluate analyst feedback loops, and estimate dollar loss reduction using institution-specific data. Future

research may also integrate sanctions screening, smart-contract risk, cross-chain bridge behavior, and decentralized finance protocol interactions. These extensions would move blockchain fraud analytics closer to the complex risk environment faced by modern digital-asset institutions. This point is consistent with related research (Yli-Huumo et al., 2016).

Appendix-Style Robustness Notes

Robust analysis should consider both statistical and operational dimensions. Statistically, the model should be tested under alternative snapshot lengths, different negative sampling strategies, and different thresholds for constructing pseudo-risk samples. Operationally, the model should be tested under realistic alert budgets. A model that performs well when it can alert 20% of nodes may not be useful if analysts can review only 5%. The recommended practice is to report model performance across several alert budgets and to show the trade-off between expected missed-risk cost and review cost. This point is consistent with related research (Liu et al., 2008).

Another robust issue is label delay. Confirmed fraud labels may arrive weeks or months after the activity occurs. If delayed labels are used incorrectly, the training process may leak future information into past decisions. The framework should therefore separate retrospective model development from prospective deployment. Retrospective analysis can use later evidence to understand patterns, but prospective scoring must use only the information available at the time of decision. This distinction is especially important for institutions that use the model to freeze funds, reject transactions, or file compliance reports. This point is consistent with related research (Casino et al., 2019).

Fairness and customer impact also matter. Blockchain addresses are pseudonymous, but risk scores can still affect real customers when addresses are linked to exchange accounts. A high-risk score should not automatically imply guilt. The score should trigger proportionate review, and the review process should preserve evidence and allow correction of errors. In this sense, the proposed framework is not a replacement for compliance judgment. It is a decision-support system that organizes data, highlights risky patterns, and improves the consistency of analyst attention. This point is consistent with related research (Breunig et al., 2000).

Finally, the framework should be integrated with broader enterprise systems. Blockchain graph signals should be combined with customer due diligence, device intelligence, sanctions lists, travel-rule data, and historical account behavior. The value of graph analytics increases when it becomes part of an integrated risk data platform. Business leaders should therefore evaluate the framework not only as a model but also as an information infrastructure investment. Its long-run value depends on data quality, feedback capture, governance discipline, and cross-functional adoption. This point is consistent with related research (Atzei et al., 2017).

9. Implementation and Governance Roadmap

The implementation of a blockchain fraud analytics framework should begin with data readiness rather than model selection. Many organizations have access to ledger data but lack a stable entity-resolution process, a consistent snapshot definition, or a reliable link between on-chain addresses and off-chain customer records. Before training any model, the institution should define which chain, asset, exchange wallet, time zone, and transaction confirmation rule will be used. It should also document whether internal transfers, exchange consolidation transactions, fee transfers, and dust transactions are included or removed. These choices affect graph structure and may change model performance more than the choice of classifier. This point is consistent with related research (Fawcett, 2006).

The second implementation step is risk taxonomy design. A generic label called fraud is too broad for operational learning. The taxonomy should distinguish stolen funds, ransomware, sanctions exposure, dark-

market activity, scam proceeds, account takeover, mixer interaction, market manipulation, and suspicious structuring. Some institutions may not have confirmed labels for every category, but taxonomy still guides analyst feedback and model interpretation. A heterogeneous graph can support multiple risk labels, yet the first production version should usually focus on the most actionable label group to avoid diluting the training signal. This point is consistent with related research (Kshetri, 2017).

The third step is featuring governance. Feature definitions must be stable, auditable, and reproducible. For transaction nodes, common features include value normalization, fee ratio, input-output structure, number of counterparties, and time-of-day pattern. For address nodes, common features include cumulative degree, net flow, repeated counterparty concentration, cluster exposure, and distance to known high-risk services. Each feature should have an owner, a refresh frequency, a missing-value rule, and a rationale. Without feature governance, model drift can be confused with data pipeline drift. This point is consistent with related research (Davis and Goadrich, 2006).

The fourth step is graph construction governance. A blockchain graph is not a single objective object; it is produced through design choices. The same raw ledger can produce a transaction graph, an address graph, a bipartite graph, or a heterogeneous graph with multiple relation types. The proposed framework recommends a heterogeneous design because it preserves business meaning, but the design should still be validated. Risk teams should inspect sample neighborhoods and confirm that edges correspond to meaningful investigative evidence. A graph that is technically valid but not operationally interpretable will be difficult to defend in compliance settings. This point is consistent with related research (Gomber et al., 2017).

The fifth step is semi-supervised label management. High-confidence normal labels may come from verified customer activity, long-standing institutional wallets, low-risk payment flows, or manually cleared alerts. Confirmed fraud labels may come from internal investigations, law-enforcement notifications, sanctions updates, or public intelligence feeds. Unlabeled data should remain unlabeled unless there is a clear rule for pseudo-labeling. The purpose of pseudo-risk calibration is not to declare unknown nodes fraudulent. It is to expose the classifier to difficult boundary patterns so that the model becomes less fragile under imbalance. This point is consistent with related research (Saito and Rehmsmeier, 2015).

The sixth step is validation design. A time-based validation split is more realistic than a random split because blockchain behavior changes over time. Random splits may place future patterns into training and create inflated results. A robust design trains early snapshots, validates on middle snapshots, and tests on later snapshots. The evaluation should report not only average scores but also score dispersion across snapshots. A model with slightly lower average F1 but much better stability may be preferable for production risk management because analysts need predictable alert volumes. This point is consistent with related research (Chen et al., 2012).

The seventh step is threshold calibration. The model should produce a continuous risk score rather than only a binary label. Compliance leaders can then select thresholds that reflect operational capacity and risk appetite. A tiered policy is often most useful. Very high scores may trigger immediate enhanced review, medium scores may enter batch investigation, and low scores may be monitored unless they accumulate additional evidence. Thresholds should be reviewed periodically because changes in market volatility, transaction volume, and criminal behavior can alter the score distribution. This point is consistent with related research (Guidotti et al., 2018).

The eighth step is explanation design. Each alert should be accompanied by a concise evidence summary. For example, the system may report that an address was flagged because it received funds from a high-risk neighborhood within two hops, showed a sudden increase in output fan-out, and appeared in a temporal burst pattern. Such explanations improve analyst trust and reduce review time. They also make it easier to identify false positives caused by exchange maintenance, wallet consolidation, or legitimate high-volume merchant activity. This point is consistent with related research (Wamba et al., 2017).

The ninth step is feedback capture. Analyst decisions should be stored in a structured format. Free-text notes are useful but insufficient for model improvement. The review interface should record whether the alert was confirmed, cleared, escalated, or left unresolved. It should also record the main reason code, such as high-risk counterparty, unusual velocity, suspicious splitting, or known service exposure. These feedback labels become a valuable training resource and can be used to audit whether the model is learning the same concepts that analysts consider important. This point is consistent with related research (Barredo Arrieta et al., 2020).

The tenth step is monitoring and audit. Production dashboards should track alert volume, precision after review, false-positive reasons, model score distribution, feature drift, graph density, and delayed confirmation rates. A sudden decline in precision may indicate model drift, data quality problems, or a change in attacker behavior. A sudden increase in alert volume may indicate a real risk event or a pipeline error. The dashboard should separate technical monitoring from business monitoring so that engineers, analysts, and executives can each see the indicators relevant to their responsibilities. This point is consistent with related research (Akter et al., 2016).

The eleventh step is human oversight. The model should support professional judgment rather than replace it. Blockchain fraud analytics affects customers, counterparties, and regulatory reporting. Automated decisions should therefore be proportionate to risk severity and supported by review procedures. For low-value or low-impact actions, automation may be acceptable. For account freezes, suspicious activity reports, or customer offboarding, human review and evidence preservation are essential. The governance process should specify which actions can be automated and which require analyst approval. This point is consistent with related research (Rudin, 2019).

The final implementation step is continuous learning. Blockchain environments change rapidly as new protocols, wallets, bridges, and laundering techniques emerge. A model trained on one period may not remain effective indefinitely. The institution should maintain a retraining calendar, a challenger-model process, and a post-incident learning procedure. When a new fraud pattern is discovered, the graph should be queried to identify related historical cases, and the resulting evidence should be used to update features, thresholds, and training data. This feedback loop turns the framework into a long-term risk intelligence capability. This point is consistent with related research (Kitchens et al., 2018).

10. Limitations and Future Research Agenda

The framework has several limitations that should guide future work. First, the numerical analysis is designed to demonstrate the logic of a business risk analytics architecture, not to replace large-scale production testing. Public blockchain datasets are valuable for reproducibility, but regulated exchanges and payment firms often have richer internal data, including customer verification status, device patterns, account history, withdrawal channels, and analyst decisions. Future studies should evaluate the framework under institutional data environments in which on-chain graph signals and off-chain customer records are combined under privacy-preserving governance rules. This point is consistent with related research (Mehrabi et al., 2021).

Second, the article focuses on Bitcoin-style transaction graphs. Other blockchain ecosystems introduce additional complexity. Account-based chains, smart-contract platforms, decentralized exchanges, bridges, and layer-two networks create different graph objects and risk mechanisms. A transfer may be embedded in a contract call, a liquidity pool interaction, or a cross-chain message. Future research should extend the heterogeneous graph design to include contract nodes, protocol nodes, token nodes, bridge events, and liquidity positions. Such an extension would allow the framework to analyze decentralized finance fraud rather than only transfer-based laundering patterns. This point is consistent with related research (Günther et al., 2017).

Third, explainability requires deeper study. This article emphasizes evidence paths and graph-level explanations, but measuring the quality of such explanations remains difficult. Future work should evaluate

whether explanations reduce analyst review time, improve consistency between analysts, and increase the defensibility of compliance actions. A technically accurate explanation may still be unhelpful if it does not correspond to the way investigators reason about fund flows. Human-centered evaluation should therefore become part of model assessment in blockchain risk analytics. This point is consistent with related research (Selbst et al., 2019).

Fourth, future research should include economic loss modeling. F1, precision, and recall are important but incomplete. Two missed fraud cases may have very different financial and regulatory consequences. A model that identifies fewer cases may still generate greater business value if it detects higher-value illicit flows. Future studies should combine graph learning with expected-loss estimation, including transaction value, recovery probability, jurisdictional exposure, customer risk tier, and reporting obligations. This would allow firms to optimize not only classification metrics but also risk-adjusted economic outcomes. This point is consistent with related research (Bronstein et al., 2017).

Fifth, privacy and responsible data use deserve attention. Blockchain data are public, but linking on-chain addresses to customer identities creates sensitive information. Institutions should limit access, maintain audit logs, apply data minimization, and define retention rules. Cross-institutional learning may be valuable because fraud patterns move across platforms, yet direct sharing of customer-level data may be restricted. Federated learning, secure multiparty computation, and privacy-preserving graph analytics may therefore become important research directions for collaborative blockchain fraud detection. This point is consistent with related research (Floridi et al., 2018).

Finally, the governance of adaptive models should be studied longitudinally. Criminal behavior changes in response to detection systems. When firms deploy stronger graph analytics, illicit actors may increase address rotation, use privacy tools, split transfers into smaller pieces, or move to less monitored chains. Future research should model this adversarial adaptation as a dynamic strategic process. A risk analytics system should not only classify current behavior but also anticipate how fraud networks may respond to the environment monitoring. Integrating adversarial learning with business governance would be a natural next step. This point is consistent with related research (Perozzi et al., 2014).

Acknowledgement

We thank the editor and anonymous reviewers for their constructive comments. We acknowledge the academic environment and research resources provided by Guangdong Technology College, Zhaoqing, Guangdong, China 526100.

Author Contributions

Conceptualization, B.L.Z.; Methodology, B.L.Z.; Formal Analysis, B.L.Z.; Investigation, B.L.Z. and C.Y.Z.; Data Curation, B.L.Z. and C.Y.Z.; Software, B.L.Z.; Validation, B.L.Z. and C.Y.Z.; Resources, B.L.Z. and C.Y.Z.; Visualization, B.L.Z.; Writing – Original Draft, B.L.Z.; Writing – Review & Editing, B.L.Z. and C.Y.Z.; Supervision, B.L.Z.; Project Administration, B.L.Z. We have read and approved the final manuscript and take responsibility for the work.

Funding

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Ethical Approval and Consent

Not applicable.

Data Availability Statement

This study draws on publicly available blockchain transaction datasets that are widely used in anomaly-detection research. The datasets are openly accessible from their original sources, and the preprocessing scripts and modelling code used to reproduce the numerical results are available from the corresponding author on reasonable request.

Declaration of Generative AI Use

The authors did not use generative AI tools in the preparation of this manuscript. All analyses, interpretations, and conclusions presented in this article are the sole responsibility of the authors.

References

Ron, D., & Shamir, A. (2013). Quantitative analysis of the full Bitcoin transaction graph. Financial Cryptography and Data Security, 6-24. https://doi.org/10.1007/978-3-642-39884-1_2

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013). A fistful of Bitcoins: Characterizing payments among men with no names. Proceedings of the 2013 Internet Measurement Conference, 127-140. https://doi.org/10.1145/2504730.2504747

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559-569. https://doi.org/10.1016/j.dss.2010.08.006

Foley, S., Karlsen, J. R., & Putnins, T. J. (2019). Sex, drugs, and Bitcoin: How much illegal activity is financed through cryptocurrencies? The Review of Financial Studies, 32(5), 1798-1853. https://doi.org/10.1093/rfs/hhz015

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113. https://doi.org/10.1016/j.jnca.2016.04.007

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. Decision Support Systems, 50(3), 602-613. https://doi.org/10.1016/j.dss.2010.05.012

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

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, 57, 47-66. https://doi.org/10.1016/j.cose.2015.09.005

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical Science, 17(3), 235-255. https://doi.org/10.1214/ss/1042727940

Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019). Deep anomaly detection on attributed networks. Proceedings of the SIAM International Conference on Data Mining, 594-602. https://doi.org/10.1137/1.9781611975673.67

Ruff, L., Vandermeulen, R. A., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Mueller, E., & Kloft, M. (2018). Deep one-class classification. Proceedings of the 35th International Conference on Machine Learning, 4393-4402. https://doi.org/10.1145/3219819.3219998

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324

Schar, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review, 103(2), 153-174. https://doi.org/10.20955/r.103.153-74

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273-297. https://doi.org/10.1007/BF00994018

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. IEEE Symposium on Security and Privacy, 104-121. https://doi.org/10.1109/SP.2015.14

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of Bitcoin. IEEE Communications Surveys & Tutorials, 20(4), 3416-3452. https://doi.org/10.1109/COMST.2018.2842460

Scholkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443-1471. https://doi.org/10.1162/089976601750264965

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology? A systematic review. PLOS ONE, 11(10), e0163477. https://doi.org/10.1371/journal.pone.0163477

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. IEEE International Conference on Data Mining, 413-422. https://doi.org/10.1109/ICDM.2008.17

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36, 55-81. https://doi.org/10.1016/j.tele.2018.11.006

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. ACM SIGMOD International Conference on Management of Data, 93-104. https://doi.org/10.1145/335191.335388

Atzei, N., Bartoletti, M., & Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts. Principles of Security and Trust, 164-186. https://doi.org/10.1007/978-3-662-54455-6_8

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. Telecommunications Policy, 41(10), 1027-1038. https://doi.org/10.1016/j.telpol.2017.09.003

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. ACM International Conference on Machine Learning, 233-240. https://doi.org/10.1145/1143844.1143874

Gomber, P., Koch, J. A., & Siering, M. (2017). Digital finance and FinTech: Current research and future research directions. Business & Information Systems Engineering, 59, 537-580. https://doi.org/10.1007/s12599-017-0464-6

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. MIS Quarterly, 36(4), 1165-1188. https://doi.org/10.2307/41703503

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), 1-42. https://doi.org/10.1145/3236009

Wamba, S. F., Gunasekaran, A., Akter, S., Ren, S. J. F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. Journal of Business Research, 70, 356-365. https://doi.org/10.1016/j.jbusres.2016.08.009

Barredo Arrieta, A., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115. https://doi.org/10.1016/j.inffus.2019.12.012

Akter, S., Wamba, S. F., Gunasekaran, A., Dubey, R., & Childe, S. J. (2016). How to improve firm performance using big data analytics capability and business strategy alignment? International Journal of Production Economics, 182, 113-131. https://doi.org/10.1016/j.ijpe.2015.12.026

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1, 206-215. https://doi.org/10.1038/s42256-019-0048-x

Kitchens, B., Dobolyi, D., Li, J., & Abbasi, A. (2018). Advanced customer analytics: Strategic value through integration of relationship-oriented big data. Journal of Management Information Systems, 35(2), 540-574. https://doi.org/10.1080/07421222.2018.1451959

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys, 54(6), 1-35. https://doi.org/10.1145/3457607

Gunther, W. A., Rezazade Mehrizi, M. H., Huysman, M., & Feldberg, F. (2017). Debating big data: A literature review on realizing value from big data. The Journal of Strategic Information Systems, 26(3), 191-209. https://doi.org/10.1016/j.jsis.2017.07.003

Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., & Vertesi, J. (2019). Fairness and abstraction in sociotechnical systems. ACM Conference on Fairness, Accountability, and Transparency, 59-68. https://doi.org/10.1145/3287560.3287598

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond Euclidean data. IEEE Signal Processing Magazine, 34(4), 18-42. https://doi.org/10.1109/MSP.2017.2693418

Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People: An ethical framework for a good AI society. Minds and Machines, 28, 689-707. https://doi.org/10.1007/s11023-018-9482-5

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 701-710. https://doi.org/10.1145/2623330.2623732

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1, 389-399. https://doi.org/10.1038/s42256-019-0088-2

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 855-864. https://doi.org/10.1145/2939672.2939754

Gordon, L. A., & Loeb, M. P. (2002). The economics of information security investment. ACM Transactions on Information and System Security, 5(4), 438-457. https://doi.org/10.1145/581271.581274

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). LINE: Large-scale information network embedding. ACM International Conference on World Wide Web, 1067-1077. https://doi.org/10.1145/2736277.2741093

Anderson, R. (2001). Why information security is hard: An economic perspective. Annual Computer Security Applications Conference, 358-365. https://doi.org/10.1109/ACSAC.2001.991552

Cao, S., Lu, W., & Xu, Q. (2015). GraRep: Learning graph representations with global structural information. ACM International Conference on Information and Knowledge Management, 891-900. https://doi.org/10.1145/2806416.2806512

Romanosky, S. (2016). Examining the costs and causes of cyber incidents. Journal of Cybersecurity, 2(2), 121-135. https://doi.org/10.1093/cybsec/tyw001

Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94. https://doi.org/10.1016/j.knosys.2018.03.006

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605

Cui, P., Wang, X., Pei, J., & Zhu, W. (2019). A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering, 31(5), 833-852. https://doi.org/10.1109/TKDE.2018.2849727

Pang, G., Shen, C., Cao, L., & Hengel, A. van den. (2021). Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2), 1-38. https://doi.org/10.1145/3439950

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2022). Graph neural networks: A review of methods and applications. AI Open, 1, 57-81. https://doi.org/10.1016/j.aiopen.2021.01.001

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Mueller, K. R. (2021). A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE, 109(5), 756-795. https://doi.org/10.1109/JPROC.2021.3052449

Hamilton, W. L. (2020). Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3), 1-159. https://doi.org/10.2200/S01045ED1V01Y202009AIM046

Skarding, J., Gabrys, B., & Musial, K. (2021). Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. IEEE Access, 9, 79143-79168. https://doi.org/10.1109/ACCESS.2021.3082932

Hasan, M., Islam, M. M., Uddin, M. P., Alyami, S. A., Moni, M. A., & Summers, M. A. (2024). Detecting anomalies in blockchain transactions using machine learning classifiers and explainable artificial intelligence. Decision Analytics Journal, 10, 100414. https://doi.org/10.1016/j.dajour.2024.100414

Chang, Z., Li, Y., Wang, X., & Zhang, J. (2025). Anomalous node detection in blockchain networks based on graph neural networks. Sensors, 25(1), 1. https://doi.org/10.3390/s25010001

Tian, Z., Cui, X., An, L., Su, S., Yin, X., Yin, L., & Cui, X. (2019). A real-time correlation of host-level events in cyber range service for smart campus. IEEE Access, 7, 90152-90164. https://doi.org/10.1109/ACCESS.2019.2926938

Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., & Samatova, N. F. (2015). Anomaly detection in dynamic networks: A survey. Wiley Interdisciplinary Reviews: Computational Statistics, 7(3), 223-247. https://doi.org/10.1002/wics.1347

Wang, D., Cui, P., & Zhu, W. (2019). Structural deep network embedding. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1225-1234. https://doi.org/10.1145/2939672.2939753

Maharjan, S., Zhu, Q., Zhang, Y., Gjessing, S., & Basar, T. (2020). Dependable demand response management in the smart grid: A Stackelberg game approach. IEEE Transactions on Smart Grid, 4(1), 120-132. https://doi.org/10.1109/TSG.2012.2223766