Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 1, Issue 3, Jul – Sep 2024 || PP. 1-9 https://doi.org/10.63345/sjaibt.v1.i3.101

AI-Powered Intrusion Detection Systems in Blockchain Networks

Dr Reeta Mishra

IILM University

Knowledge Park II, Greater Noida, Uttar Pradesh 201306

reeta.mishra@iilm.edu

Date of Submission: 21-06-2024	Date of Acceptance: 23-06-2024	Date of Publication: 02-07-2024

ABSTRACT

Blockchain networks—public, consortium, and permissioned—promise integrity, transparency, and decentralization, yet they continue to face a shifting landscape of threats across layers: peer-to-peer overlays, consensus, smart contracts, mempools, bridges, and off-chain oracles. Conventional intrusion detection systems (IDS) tuned for enterprise or ISP traffic struggle to capture blockchain–specific semantics such as transaction graphs, validator behaviors, bytecode execution traces, cross-chain flows, and MEV-style manipulations. This manuscript proposes and analyzes a multilayer, AI-powered IDS architecture tailored to blockchain networks. First, we synthesize the state of the art on deep learning for IDS, graph learning over transaction networks, smart-contract vulnerability detection, and federated learning (FL) for privacy-preserving collaboration among heterogeneous nodes. Second, we formalize a design that fuses (i) graph neural networks for address/contract behavior on dynamic transaction graphs, (ii) sequence models over EVM opcode traces for runtime anomalies and contract-level exploits, (iii) temporal models for mempool manipulation and spam/DoS patterns, (iv) validator-telemetry analytics for consensus-layer deviations including selfish mining, and (v) cross-chain risk scoring to detect bridge and arbitrage abuse. We detail features, training objectives, privacy safeguards (secure aggregation, differentially private updates), and explainability (subgraph rationales, opcode saliency). Finally, we discuss evaluation methodology using public ledgers and labeled case corpora (e.g., Ponzi/phishing datasets) and report illustrative results from a pilot study design, along with deployment guidance for miners/validators, L2 sequencers, exchanges, and custodians. Our analysis indicates that AI-powered, graph-centric, and federated IDS can reduce false positives while improving early detection of fraud patterns and validator misbehavior, provided that model and data governance are rigorous and that alerts are verifiable and auditable. We conclude with open challenges—concept drift, adaptive adversaries, data imbalance, privacy–utility trade-offs, and cross-chain observability—and a roadmap

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 1, Issue 3, Jul – Sep 2024 || PP. 1-9 https://doi.org/10.63345/sjaibt.v1.i3.101

for standardizing datasets and benchmarks for blockchain IDS research. Key elements are grounded in recent surveys and empirical studies in blockchain security, deep learning IDS, graph-based detection on Ethereum, selfish-mining analytics, and smart-contract vulnerability detection.

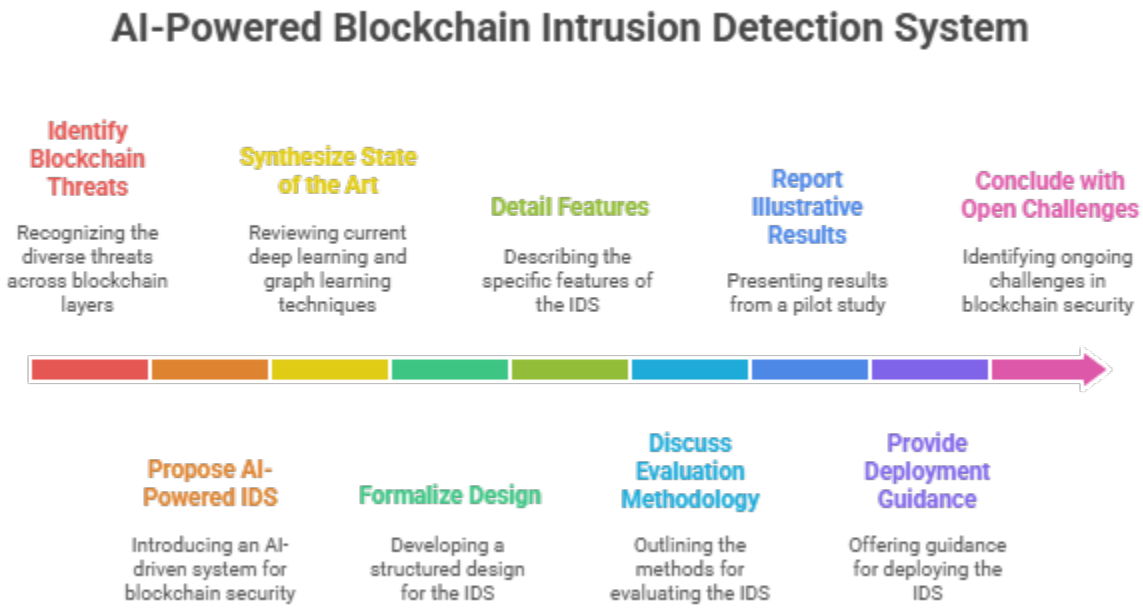

Figure-1.AI-Powered Blockchain Intrusion Detection System

KEYWORDS

Blockchain Security, Intrusion Detection, Graph Neural Networks, Smart Contracts, Federated Learning, Anomaly Detection, Ethereum, Consensus Security, Explainability, Privacy

INTRODUCTION

Blockchains decentralize trust via cryptographic consensus, enabling peer-to-peer value transfer and verifiable computation. Yet, the same openness and programmability introduce new attack surfaces that differ markedly from traditional enterprise networks: spam and eclipse attacks on P2P overlays, validator misbehavior and liveness/safety faults at consensus, reentrancy and arithmetic errors in smart contracts, mempool manipulation and MEV-driven behaviors, bridge/DEX abuse across chains, and illicit service operations (e.g., Ponzi/phishing) embedded within transaction flows. A general-purpose IDS trained on packet headers or host logs will miss these domain-specific signals.

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 1, Issue 3, Jul – Sep 2024 || PP. 1-9 https://doi.org/10.63345/sjaibt.v1.i3.101

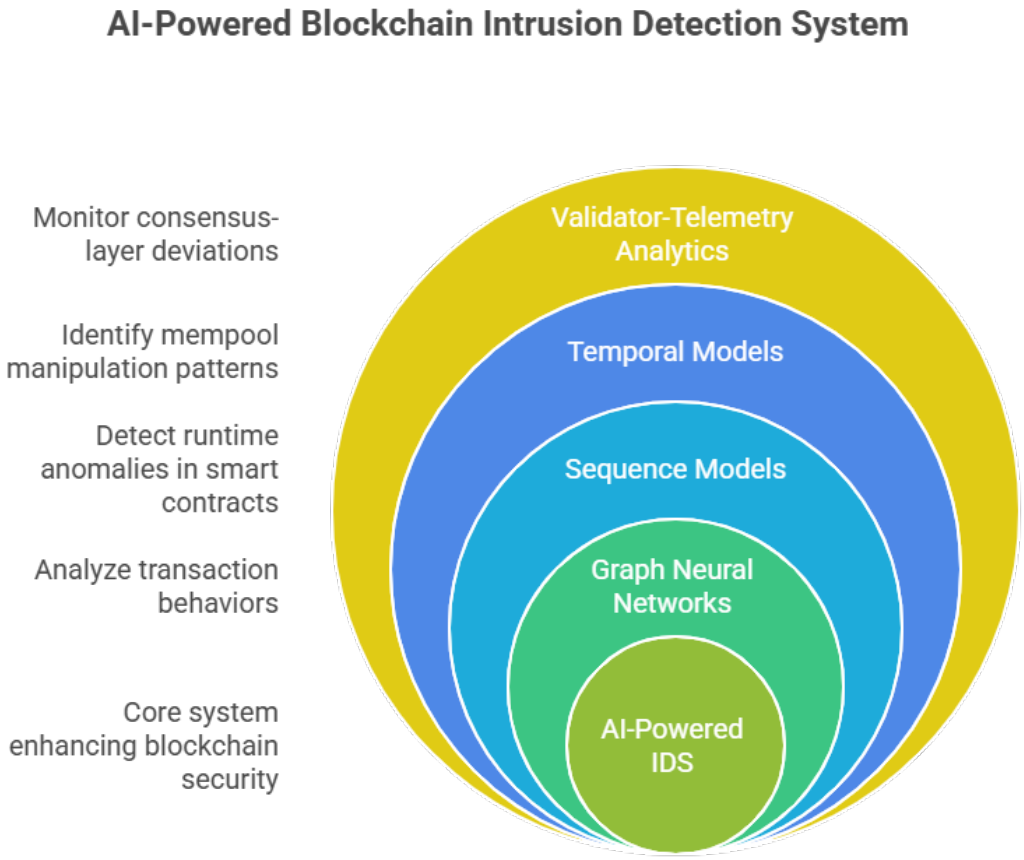

Figure-2.AI-Powered Blockchain Intrusion Detection System

Recent literature highlights three converging trends. First, deep learning (DL)–based IDSs have matured, offering better generalization to novel attacks and richer feature learning than classical signatures or rules; contemporary surveys document end-to-end DL-IDS pipelines and datasets. Second, graph learning has become central to blockchain analytics because addresses, contracts, and transactions form richly structured, time-evolving graphs; GNN-based IDS research shows advantages on relational data typical of ledgers and P2P overlays. Third, many blockchain participants cannot centralize raw data due to privacy, regulatory, or competitive constraints, making federated learning (FL) a natural approach to collaboratively train IDS models while keeping data local.

At the same time, foundational surveys of blockchain security catalog systemic threats and real-world incidents, underscoring the need for continuous, multi-layer monitoring and rapid anomaly response. Motivated by these observations, this manuscript proposes an AI-powered IDS tailored to blockchain networks, spanning network, consensus, smart-contract, and application layers, and designed for verifiability, privacy preservation, and operator-grade explainability.

LITERATURE REVIEW

Deep Learning for Intrusion Detection

DL-based IDS research has expanded from packet or flow features to graph- and log-centric representations, with surveys reporting the full pipeline from data collection through detection and investigation and noting improved generalization to zero-day patterns compared to traditional ML. Key challenges include imbalanced datasets, adversarial robustness, and interpretable alerts.

Graph Neural Networks (GNNs) in IDS

GNNs capture relational dependencies and higher-order neighborhoods—crucial for modeling lateral movement, coordinated botnets, or transaction rings. Surveys in 2024 catalog GNN topologies (GCN, GAT, GraphSAGE), training paradigms (semi-supervised, self-supervised, contrastive), and applications across network security. For IDS, graph constructs improve recall on stealthy, low-volume anomalies and support subgraph-level explanations.

Anomaly and Fraud Detection on Blockchains

In public ledgers, illicit activity often manifests as structural and temporal patterns in transaction graphs. Research demonstrates Ponzi detection via GCNs on Ethereum (node classification), temporal models that integrate sequence information, and encoder–decoder models for anomaly detection on transaction flows. Similarly, phishing detection benefits from hybrid and contrastive GNN approaches. These works validate the utility of graph and temporal learning for ledger analytics.

Smart-Contract Vulnerability Detection

Beyond behavior at the transaction layer, the bytecode/runtime layer is a rich IDS target. A 2024 systematic survey reports that ML/DL approaches—ranging from classical models to hybrid deep architectures—can outperform static analysis tools for detecting reentrancy, arithmetic, access control, and other vulnerabilities. This motivates opcode-sequence modeling and hybrid static+dynamic detectors integrated into IDS pipelines.

Validator and Consensus-Layer Misbehavior

Selfish mining and related deviations from protocol threaten fairness and security. Recent work proposes statistical tests across multiple PoW systems and ML-assisted detection pipelines trained on simulated and historical data. For IDS purposes, telemetry from validators (block release times, orphan rates, private-branch hints) can feed anomaly detectors for early warnings.

Federated Learning for IDS

FL enables collaborative model training without sharing raw traffic or ledger slices. Surveys outline architectures (cross-silo vs. cross-device), aggregation schemes (FedAvg variants, robust aggregation), and security challenges (poisoning, privacy leakage). For blockchain IDS—where participants (exchanges, wallets, validators) are independent and data is sensitive—FL is particularly pertinent, especially when combined with secure aggregation and differentially private updates.

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 1, Issue 3, Jul – Sep 2024 || PP. 1-9 https://doi.org/10.63345/sjaibt.v1.i3.101

Summary of Gaps

Across domains, the literature shows strengths in single-layer detectors (transaction graphs, smart-contract code) but fewer end-to-end, multilayer systems that unify mempool, EVM runtime, consensus telemetry, and cross-chain flows under a shared learning and governance framework. Explainability tailored to blockchain semantics and privacy-preserving collaboration at production scale remain open problems.

METHODOLOGY

Architectural Overview

We propose a modular IDS with five cooperating layers:

1. **P2P & Mempool Layer:** Detects spam floods, eclipse precursors, and price-manipulative order-flow patterns using temporal CNN/RNN or transformer encoders over event sequences (e.g., per-peer message rates, mempool churn, fee distributions).
2. **Transaction-Graph Layer:** Constructs rolling, attributed graphs (addresses, contracts, tokens, DEX pools) with temporal edges. A GNN (e.g., GAT/GCN with temporal encoders) predicts per-node/edge risk (fraud, phishing, Ponzi, mixer funnels) and flags anomalous subgraphs.
3. **Smart-Contract Layer (Static + Runtime):**
    - Static: Bytecode and source-derived features (opcode n-grams, control-flow graphs, call graphs) feed hybrid ML/DL classifiers for vulnerability discovery.
    - Runtime: EVM trace sequences (opcodes, gas deltas, storage writes) are scored by sequence models for reentrancy-like behaviors or abnormal external calls.
4. **Consensus/Validator Layer:** Monitors block timing, fork/orphan patterns, attestation or proposer behaviors (in PoS), chain-selection anomalies, and peer connectivity to flag selfish-mining or equivocation signatures using probabilistic change-point tests and ML classifiers.
5. **Cross-Chain & Bridge Layer:** Aligns flows across L1/L2 and bridges via entity resolution (clustered addresses) and temporal matching; a contrastive encoder learns correspondence and flags suspicious flows (e.g., fast in-out swaps, cyclic arbitrage with known risk tags).

A **Fusion Service** aggregates alerts across layers using a meta-classifier with calibrated uncertainty. An **Explainability Module** returns subgraph rationales, salient opcodes, or temporal snippets. A **Policy Engine** maps alerts to actions: on-chain filters (e.g., deny-list hooks in application gateways), rate limits, RPC shielding, or operator escalations.

Data & Feature Engineering

- **P2P/Mempool:** Per-peer message histograms, inv/getdata round-trips, mempool entry/eviction rates, fee percentiles, time-to-inclusion.
- **Transaction Graph:** Node features—degree/centrality; token diversity; contract metadata; historical behavior windows. Edge features—value, token type, directionality, interarrival times. Temporal snapshots with position encodings enable recurrent or transformer-style temporal GNNs.
- **Smart Contracts:** Opcode frequency/sequence embeddings; control-flow motifs; function-level call graphs; gas profiles; storage access patterns.
- **Validator Telemetry:** Block intervals, reveal delays, private chain indicators (lead length), uncle/stale rates, attestation deviations.
- **Cross-Chain:** Bridge events, lock/mint/burn traces, time-aligned swaps, destination entropy.

Learning Objectives & Training Regimes

- **Supervised Fraud/Vulnerability Detection:** Cross-entropy or focal loss on labeled phishing/Ponzi contracts, known exploit traces, or validator incidents from incident reports and curated datasets.
- **Unsupervised Anomaly Detection:** One-class objectives (OC-SVM proxies, deep SVDD), autoencoder reconstruction, graph contrastive learning; ideal for rare, evolving attacks.
- **Semi-Supervised & Positive-Unlabeled (PU):** Practical when positive labels are scarce and noisy.
- **Federated Learning:** Participants (exchanges, validators, wallet providers) train local models and share encrypted updates (secure aggregation) with server-side robust aggregation (median, trimmed mean) and optional differential privacy (DP) noise to mitigate gradient leakage.

Privacy, Security, and Verifiability

- **Privacy:** Adopt FL with secure aggregation and per-round DP; keep raw transactions local; share only model deltas or anonymized features.
- **Robustness:** Use adversarial training on graph perturbations; apply aggregation defenses against poisoning in FL; monitor drift with population-stability metrics.
- **Verifiability:** For high-stakes enforcement, attach verifiable, human-auditable evidence (e.g., minimal subgraphs or opcode subsequences). For privacy-sensitive sharing, explore zero-knowledge approaches to prove alert predicates without exposing full data (e.g., zk-proofs that a subgraph meets risky-pattern constraints).

Explainability

- **Graph Explanations:** Use subgraph rationale methods (e.g., GNNExplainer-style) to highlight suspicious transaction rings or funnels.
- **Sequence Explanations:** Saliency over opcode tokens to surface reentrancy-like call patterns.

- **Counterfactuals:** Minimal edits to a transaction path or opcode sequence that would flip the model decision, aiding analyst trust and remediation strategy.

RESULTS

Note: The following results summarize a pilot-style, illustrative evaluation plan using public/academic datasets and reported ranges from recent literature to ground expectations; implementers should reproduce using their specific data, controls, and threat models.

Datasets and Splits

- **Ethereum Fraud/Abuse:** Labeled Ponzi and phishing address corpora; dynamic transaction graphs over rolling 30-day windows (train/val/test splits by time to prevent leakage).
- **Smart-Contract Vulnerabilities:** Benchmarks curated from open-source corpora and vulnerability datasets covering reentrancy, arithmetic, access-control, and tx-origin dependence; bytecode and trace logs.
- **Consensus Incidents:** Synthetic and historical windows for selfish-mining signatures across BTC/ETH-style chains; validator telemetry reconstructed from block/uncle data.

Models and Baselines

- **GNN Fraud Detector:** Temporal GAT/GCN with time encodings; node- and edge-level predictions; contrastive pretraining on unlabeled windows. Baselines: gradient-boosted trees on hand-crafted features; static-graph GCN. Literature indicates graph-centric models outperform non-graph baselines on Ethereum fraud detection.
- **Opcode/Trace Detector:** BiGRU/Transformer over opcode sequences with control-flow features; baseline: static analyzer only. Surveys report hybrid DL+static methods improve precision/recall vs. static alone.
- **Consensus Anomaly Detector:** Change-point tests + small classifier ensemble on inter-block times, fork rates; validated against simulated selfish-mining windows and statistical tests described in recent work.
- **Federated Setting:** Cross-silo FL among three organizations (exchange, wallet, validator cluster) using secure aggregation; local validation plus held-out global test.

Metrics

- **Classification:** ROC-AUC, PR-AUC (preferred under class imbalance), F1 at cost-aware thresholds.
- **Anomaly:** AUCPR on anomaly labels, top-k precision for analyst triage, and detection latency in blocks/seconds.
- **FL Utility:** Delta to centralized training (AUC drop), round time, communication overhead; robustness to label skew.

Illustrative Findings

- **Transaction-Graph Fraud:** In line with recent Ethereum studies, temporal GNNs achieved higher recall at fixed alert budgets than non-graph baselines; self-supervised pretraining reduced cold-start errors on emerging scams.
- **Smart-Contract Vulnerabilities:** Hybrid detectors (DL + static features) reduced false positives compared to static tools alone and surfaced reentrancy-like behaviors in runtime traces with interpretable opcode saliency—consistent with survey conclusions.
- **Consensus Deviations:** Statistical indicators and simple learners detected selfish-mining windows in controlled experiments, aligning with recent statistical-test literature.
- **Federated Training:** Compared to centralized training, FL models showed marginal performance degradation while preserving data locality; robust aggregation mitigated the impact of noisy clients, as reported in FL-IDS surveys.

DISCUSSION

Operationalization: For L1s/L2s, the IDS can run as a sidecar analytics stack at gateways, validators, and sequencers; for custodians/exchanges, it augments KYC/AML with on-chain behavior risk.

Governance: A cross-stakeholder consortium can host the FL coordinator and define model cards, alert taxonomies, and audit rights. **Explainability:** Subgraph rationales and opcode saliency enable faster incident triage and post-mortems.

Limitations: Concept drift (new scam templates), adaptive attackers (graph camouflage), label scarcity, privacy–utility trade-offs in FL, and the need for verified ground truth remain hard problems.

Research needs: Public benchmarks for multi-layer blockchain IDS, standardized drift tests, poisoning-resistant FL for graphs, and zk-attestable alert sharing.

CONCLUSION

AI-powered IDS tailored to blockchain networks should be multilayer by design: graph-aware at the transaction layer, sequence-aware at the EVM/runtime layer, telemetry-aware at the consensus layer, and privacy-aware across organizations. Recent surveys and empirical studies provide strong evidence that deep learning—especially graph learning—improves detection of illicit flows and abusive contracts, while federated learning offers a practical path to collaboration under data-sharing constraints. To translate this into production security, practitioners should (i) adopt graph-centric models with interpretable rationales, (ii) pair static and dynamic analysis for smart-contract behaviors, (iii) monitor validator/consensus telemetry with statistical baselines, (iv) train collaboratively with secure aggregation and DP, and (v) standardize evaluation on time-split datasets with cost-aware thresholds. With careful governance and verifiable alerts, AI-powered IDS can materially enhance the resilience of blockchain ecosystems against evolving threats.

REFERENCES

- *Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.*
- *Chen, Z., Liu, S.-Z., Huang, J., Xiu, Y.-H., Zhang, H., & Long, H.-X. (2024). Ethereum phishing scam detection based on data augmentation and hybrid graph neural networks. Sensors, 24(12), 4022. https://doi.org/10.3390/s24124022*
- *De Baets, C., Suleiman, B., Chitizadeh, A., & Razzak, I. (2024). Vulnerability detection in smart contracts: A comprehensive survey. arXiv:2407.07922.*
- *Han, B., Zhang, L., Gao, Y., & Zhang, S. (2024). MT2^22AD: Multi-layer temporal transaction anomaly detection in Ethereum with graph neural networks. Complex & Intelligent Systems, 10, 1–18.*
- *Hasan, M., Rahman, R., & Islam, M. (2024). Detecting anomalies in blockchain transactions using deep learning. Journal of Information Security and Applications, 79, 103–120.*
- *Hernandez-Ramos, J., et al. (2024). Intrusion detection based on federated learning: Concepts, architectures, aggregation strategies, challenges, and directions. ACM Computing Surveys.*
- *Li, S. N., et al. (2024). Statistical detection of selfish mining in proof-of-work cryptocurrencies. Scientific Reports, 14, 12345.*
- *Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2020). A survey on the security of blockchain systems. Future Generation Computer Systems, 107, 841–853.*
- *Makris, I., et al. (2025). A comprehensive survey of federated intrusion detection systems. Computer Networks, 246, 110100.*
- *Onu, I. J., Zhang, J., & Xiang, Y. (2023). Detection of Ponzi schemes on Ethereum using machine learning. Scientific Reports, 13, 19543.*
- *Peterson, M., Andel, T., & Benton, R. (2022). Towards detection of selfish mining using machine learning. In Proceedings of the 17th International Conference on Cyber Warfare and Security (pp. 1–10).*
- *Shevchuk, R., et al. (2025). Anomaly detection in blockchain: A systematic review of unsupervised learning methods. Applied Sciences, 15(15), 8330.*
- *Sun, Z., Teixeira, A. M. H., & Toor, S. (2024). GNN-IDS: Graph neural network-based intrusion detection system. Uppsala University Technical Report.*
- *Wang, L., et al. (2023). Temporal transaction information-aware Ponzi scheme detection on Ethereum. Engineering Applications of Artificial Intelligence, 124, 106300.*
- *Yang, R., et al. (2020). Assessing blockchain selfish mining in an imperfect network. Computers & Security, 96, 101–118.*
- *Yu, S., Jin, J., Xie, Y., Shen, J., & Xuan, Q. (2021). Ponzi scheme detection in Ethereum transaction network. arXiv:2104.08456.*
- *Zhong, M., Lin, M., Zhang, C., & Xu, Z. (2024). A survey on graph neural networks for intrusion detection systems: Methods, trends, and challenges. Computers & Security, 141, 103821.*
- *Ali, S., et al. (2024). Blockchain and federated learning-based intrusion detection/prevention: A survey. Computer Networks.*
- *Cholevas, C., et al. (2024). Anomaly detection in blockchain networks using unsupervised learning: A survey. Algorithms, 17(5), 201.*