International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-June-2026, ISSN (Online): 2395-566X

CryptoTrack: A Data-Driven Framework for Detecting and Explaining Cryptocurrency Laundering

Gaurav A. Bagul, Parth P. Jadhav, Pratik S. Rahane, Assistant Professor Vipin K. Wani
Computer Engineering
MET BKC Institute Of Engineering

Abstract- Cryptocurrencies have rapidly grown into a pop- ular medium of digital exchange, offering speed, security, and borderless transactions. While these benefits have driven global adoption, the pseudony- mous and decentralized nature of cryptocurrencies also makes them highly vulnerable to misuse in ille- gal activities such as money laundering, terrorism financing, and fraud. Recent reports highlight bil- lions of dollars being laundered annually through cryptocurrency channels, often using techniques like mixers, peel chains, and cross-chain transfers. Traditional Anti-Money Laundering (AML) sys- tems, designed mainly for conventional banking transactions, struggle to handle the complexities of blockchain-based transactions. They often func- tion as black boxes, providing risk scores without clear reasoning, and they are reactive rather than proactive in detecting suspicious activities. To address these challenges, the proposed sys- tem CryptoTrack: A Data-Driven System for De- tecting Cryptocurrency Laundering. The system leverages advanced analytics to identify suspicious accounts and transactions, while integrating Ex- plainable Artificial Intelligence (XAI) to provide transparent justifications for every detection. Un- like existing systems that only flag activities, Cryp- toTrack enables users and compliance officers to understand the exact reasons why a transaction is considered risky, thereby increasing trust and re- ducing false positives. A visualization dashboard further supports users by providing intuitive in- sights into detected suspicious activity. The proposed framework bridges the gap be- tween opaque detection models and the practical requirement for interpretability in financial mon- itoring. By combining data- driven detection, ex- plainability, and transparency, CryptoTrack offers a more reliable and effective approach to combating financial crimes in the rapidly evolving landscape of cryptocurrency.

Keywords— Cryptocurrency, Money Laun- dering, AML, Explainable AI, Blockchain Analyt- ics, Transparency.

I. INTRODUCTION

Cryptocurrencies function on decentralized, dis- tributed ledger systems where transactions occur without central intermediaries. This decentraliza- tion, combined with pseudonymous address struc- tures, creates a transaction environment in which financial flows can be rapidly executed and glob- ally propagated while remaining difficult to trace to real-world identities. These characteristics give rise to complex money laundering methodologies such as peel chains, high-frequency layering, recur- sive mixing, cross-chain arbitrage laundering, and multi-hop obfuscation involving sets of cooperating wallet clusters.

Traditional Anti-Money Laundering (AML) mechanisms are insufficient because they rely on static rules (threshold triggers, velocity checks, or counterpart risk flags) that fail to generalize to evolving laundering patterns. Such rule-based sys- tems treat transactions independently and disre- gard topological and temporal dependencies inher- ent to blockchain transaction graphs. Additionally, they offer limited transparency, producing binary risk indicators without explaining the underlying behavioral signals.

Recent technical advances—namely Graph Neural Networks (GNNs), sequence models such as BiLSTMs, and post-hoc interpretability techniques such as SHAP—enable the modeling of multi- hop dependencies, behavioral trajectories, and ex- plainable reasoning signals. CryptoTrack leverages these capabilities to construct a unified framework for relational, temporal, and interpretable AML detection, overcoming limitations of conventional systems by grounding detection in mathematically encoded behavioral signatures.

Objectives and Motivation

CryptoTrack aims to establish a technically rigor- ous AML detection framework capable of modeling blockchain transactions as high-dimensional graph- temporal structures while providing interpretable outputs that align with regulatory expectations. The system is intended not merely to flag suspi- cious addresses but to identify structural launder- ing behaviors, quantify temporal inconsistencies, and attribute risk to measurable features.

The motivation arises from three unresolved technical gaps: (1) the absence of systems integrat- ing graph-level and temporal-level behavioral mod- eling, which is crucial because

laundering strate- gies evolve both structurally and chronologically;

(2) the lack of inductive, scalable architectures capable of scoring millions of addresses without recomputation; and (3) insufficient explainability in existing ML-based AML tools, which restricts operational deployment due to regulatory trans- parency requirements. CryptoTrack directly ad- dresses these issues through an architecture that fuses GNN relational encoders, BiLSTM temporal encoders, and SHAP-based post- hoc interpretabil- ity.

II. PROBLEM STATEMENT

Blockchain money laundering detection is funda- mentally a graph-temporal learning problem. Each wallet participates in a dynamic, directed trans- action graph where illicit behavior is distributed across multi-hop flows, emergent subgraph mo- tifs, and burst-like temporal patterns. Traditional AML systems and isolated ML classifiers treat data as static tables, discarding graph connectivity and temporal evolution, resulting in poor detection of layered laundering and adversarial obfuscation.

The core problem addressed in this work is the absence of a unified, technically grounded AML detection pipeline that: (i) captures multi-hop topological relationships between wallets; (ii) mod- els sequential behavioral evolution within trans- action histories; (iii) scales to production-grade blockchain data sizes; and (iv) provides quantifi- able, transparent explanations. Without such ca- pabilities, AML detection remains brittle, opaque, and reactive rather than proactive.

III. LITERATURE SURVEY

The increasing integration of Graph Neural Net- works (GNN), temporal deep learning, and Ex- plainable AI (XAI) has significantly shaped re- search efforts focused on detecting cryptocurrency- based money laundering. To develop a detailed un- derstanding of these approaches, more than twenty peer-reviewed studies, frameworks, and datasets were examined. These works cover graph-based blockchain forensics, hybrid deep learning archi- tectures, interpretability methods, and real-world AML datasets, forming the foundation of the pro- posed CryptoTrack framework.

Graph-Based and Relational Approaches Several researchers have demonstrated that blockchain transactions can be effectively mod- eled as graph structures capturing wallet-to-wallet relationships. Raja et al. (2025) highlighted how combining deep learning with graph analysis im- proves the identification of high-risk laundering clusters [1]. Kute et al emphasized the role of graph-centric learning in uncovering multi-hop laundering paths that traditional systems over- look Irshad et al. further advanced this field through GCF-MLD, a graph-clustering fusion model capable of detecting suspicious communities in large-scale blockchain networks [3]. Weber et al. demonstrated the use of Graph Convolutional Networks (GCN) for Bitcoin forensics, proving that relational modeling significantly boosts AML detection accuracy [7]. These studies collectively validate graph-based learning as a foundation for laundering detection.

Temporal and Sequential Modeling
Money laundering schemes often rely on timed behavioral patterns such as rapid layering, fund dispersal, and high-frequency transaction bursts. Alarab and Prakoonwit introduced a Graph-LSTM hybrid capturing both relational and temporal de- pendency structures within Bitcoin data [4]. Pham and Lee showed that sequence-based models effec- tively detect temporal anomalies within blockchain networks [8]. Various studies conclude that com- bining BiLSTM models with graph embeddings im- proves sensitivity to evolving laundering chains, re- inforcing the need for hybrid structural-temporal approaches used in CryptoTrack.

Explainable AI (XAI) in AML Interpretability is essential in AML due to regu- latory requirements demanding clear justification for every flagged transaction. Lundberg and Lee introduced SHAP, an influential interpretability framework providing consistent feature-level expla- nations for complex ML models [5]. Eddin et al. implemented XAI principles for AML alert opti- mization and demonstrated the value of transpar- ent risk scoring in operational environments [6]. Xu and Chen expanded XAI to graph models, en- abling explanation of GNN predictions for AML tasks [?]. Rahman et al. emphasized the impor- tance of integrating XAI in blockchain AML for analyst trust and compliance readiness [14]. These contributions directly influence CryptoTrack's ex-plainability engine.

Datasets, Benchmarks, and Practical Constraints
Blockchain AML research faces challenges includ- ing limited labeled datasets and complex multi- chain environments. Chen et al. surveyed AML datasets and demonstrated the importance of graph-based benchmarks for evaluating detection models [12]. Li et al. explored cross-chain launder- ing detection, showing the need for models capable of adapting to heterogeneous blockchain ecosys- tems [13]. Chainalysis' 2024 Crypto Crime Report remains the most up-to-date real-world source for laundering trends and darknet activity [15]. Reg-ulatory bodies such as the Financial Conduct Au- thority (FCA)

also provide compliance constraints that shape AML system design [?]. These insights highlight why CryptoTrack emphasizes scalability, interpretability, and multi-chain readiness.

In comparison to existing work, persistent methodological gaps are evident. Prior mod- els rarely combine structural and temporal en- coders, resulting in incomplete behavioral repre- sentations. Most datasets capture limited chains and lack cross- chain semantics, reducing general- izability. Many systems detect laundering only after full pattern formation rather than identify- ing early-stage behavioral drift. Further, exist- ing GNN-based solutions typically lack localized explainability, making them unsuitable for reg- ulated AML workflows. These gaps substanti- ate the technical necessity of CryptoTrack's fused graph–temporal–explainable framework. In com- parison to existing work, persistent methodologi- cal gaps are evident. Prior models rarely combine structural and temporal encoders, resulting in in- complete behavioral representations [9, 10]. Most datasets capture limited chains and lack cross- chain semantics [11, 12]. Many systems detect laundering only after full pattern formation rather than identifying early-stage behavioral drift [13]. Further, existing GNN-based solutions typically lack localized explainability [?, 14], making them unsuitable for regulated AML workflows. These gaps substantiate the technical necessity

IV. METHODOLOGY

The methodology of CryptoTrack is designed to model blockchain transactions as a multi-relational dynamic graph and apply learned representations to detect laundering signatures. The pipeline con- sists of six tightly integrated layers: data ingestion, preprocessing, graph construction, feature model- ing, dual-encoder architecture, and explainability.

Data Ingestion Layer: Blocks, logs, and transaction receipts are pulled from chain-indexed RPC endpoints. Each on-chain event is decom- posed into atomic transfer operations and enriched with metadata such as token type, timestamp delta, and execution trace flags. Reorg-aware in- gestion ensures data consistency under chain forks.

Preprocessing and Address Cluster- ing: To account for wallet fragmentation, Cryp- toTrack applies heuristic and clustering-based ag- gregation techniques to map multiple operational addresses into unified entities. Time-normalization transforms timestamps into monotonic sequences and filters micro-dust spam that inflates graph de- gree artificially.

Graph Construction: A directed, at- tributed multigraph G = (V, E) is formed. Each edge stores features such as normalized value, transfer rate, inter-arrival time, and transactional entropy. Node features track rolling-window ag-
gregates (inflow variance, counterparty entropy, H- index of transaction graph), enabling the capture of behavioral locality and volatility.

Relational Modeling via GNN: The relational encoder applies GraphSAGE or GAT layers to compute structural embeddings. Unlike static node2vec embeddings, the GNN propagates feature signals over k-hop neighborhoods, enabling the detection of laundering motifs such as hub- spoke patterns, peel sequences, and rapid redistri- bution flows.
transformed into graph representations where wal- lets are nodes and transactions act as directed edges.
$h(k) = \sigma(W(k) \cdot AGGREGATE(\{h(k-1), xuv\} | u \in N(v)))$
v u

Temporal Modeling via BiLSTM: Each wallet's chronological transaction series is modeled to identify abrupt behavior changes, burst anoma- lies, and structured layering sequences. The BiL- STM captures forward and backward dependen- cies:

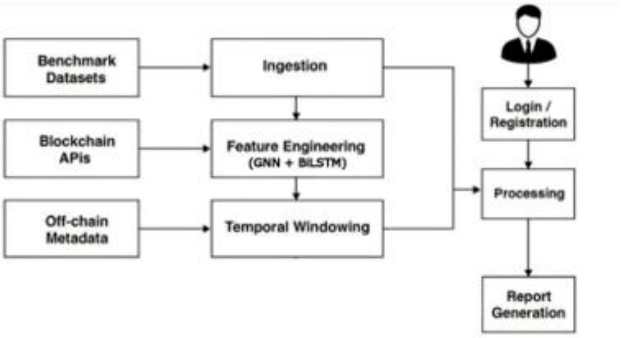

Figure 1. System Architecture of CryptoTrack

Below we expand each architectural component with implementation-level details.

Data Ingestion and Stream Processing
$h(T)$
$\longrightarrow$
$\longleftarrow$
A fault-tolerant ingestion layer consumes raw
v = [LSTM(x1:m),
LSTM(x1:m)]
blocks/transactions. The system supports two
6. Fusion and Classification: The fused em- bedding
zv = [h(G)||h(T)||xv]
modes:

Batch mode for historical reprocessing (par-allelized via partitioned block ranges). is fed to fully connected layers with dropout reg- ularization and softmax/sigmoid outputs for clas- sification. Weighted losses and focal adjustments counteract extreme class imbalance, which is typi- cal in AML datasets.

Explainability Layer: CryptoTrack in- tegrates SHAP to quantify feature-level attribu- tions and GNNExplainer for graph-level localiza- tion. Explanations identify which edges, temporal segments, or derived features contributed most to a laundering prediction, enabling regulatory-aligned interpretability.

This layered methodology ensures a technically grounded, reproducible, and scalable AML system capable of handling real-world blockchain dynam- ics.

Proposed Architecture

The proposed CryptoTrack framework integrates structural learning, temporal behavior modeling, and explainability to provide a comprehensive AML detection solution. The architecture be- gins with a data ingestion module that collects blockchain transaction data from public APIs and open datasets. After preprocessing, the data is

Stream mode for near-real-time scoring us- ing micro-batches (e.g., 1–5 minute windows) and an event queue (Kafka/RabbitMQ).
The ingestion layer validates block headers, applies canonicalization (unit conversion), and pushes nor- malized events to a feature store and raw ledger archive.

Feature Store and ETL

A centralized feature store materializes node and edge aggregates for given windows. ETL jobs com- pute:

- Rolling aggregates (sums, means, variances) over configurable windows.
- Temporal motifs (e.g., repeated circular transfers).
- Label propagation for weak labels (e.g., mark nodes connected to sanctioned addresses).

Features are versioned to ensure reproducible training and rollback.

Graph Builder and Storage

The graph builder converts event streams into an attributed directed multigraph. For scalability:

- Sharded graph partitions (by address names- pace or hash) are stored in a graph database (e.g., Neo4j, TigerGraph) or in distributed formats (Apache Parquet + adjacency lists).
- An index layer supports neighborhood queries and k-hop extraction for sampled subgraphs.

Model Training Infrastructure

Model training uses distributed frameworks (Py- Torch Geometric / DGL) with:

- Neighbor sampling (GraphSAGE) to limit re- ceptive field and memory.
- Mixed-precision training on GPUs for GNN layers and BiLSTM.
- Checkpointing and experiment tracking (MLflow) for hyperparameter sweeps.

Inference and Scoring Inference supports:

- Online scoring: micro-batch inference that updates node risk scores and pushes alerts.
- On-demand graph queries: investigator- driven subgraph scoring for deep dives.

Latency targets are defined (e.g., sub-second for single-node scoring, sub-minute for full micro- batch).

Explainability and Analyst Dashboard

Explanations are generated asynchronously for flagged alerts to reduce latency. The dashboard integrates:

- Risk timeline for each wallet.
- SHAP bar plots for top contributing features.
- Graph visualization of the influential sub- graph (edges colored by contribution).
- Actionable recommendations and links to SAR templates (automated text via tem- plates or LLM-assisted drafting).

Operational Controls, Security and Privacy Operational considerations include:

- Role-based access control (RBAC) for ana- lysts.
- Audit logging of all model decisions and ex- planation views.
- Data governance: retention policies, PII con- trols, and encrypted storage.
- Support for federated learning modes when raw data cannot be shared across institutions.

Implementation Scope and Future Work

The implementation of CryptoTrack involves con- structing a scalable pipeline for blockchain data collection, feature extraction, model training, and XAI integration. Real-world datasets such as Elliptic, Bitcoin transaction data, and AMLSim will play a crucial role in training and valida- tion. The system will incorporate graph feature extraction, temporal sequence preprocessing, hy- brid GNN–BiLSTM modeling, and SHAP-based explainability.

Future work focuses on enhancing cross-chain analytics, enabling CryptoTrack to detect laun- dering behavior across multiple blockchain net- works. Additional research directions include applying federated learning to support privacy-preserving collaboration between institutions, de- veloping streaming GNN models for real-time de- tection, and refining XAI techniques tailored to graph-based AML systems.

Applications and Use Cases

CryptoTrack can be applied across cryptocurrency exchanges, blockchain analytics companies, finan- cial intelligence units, and regulatory agencies. Ex- changes can integrate the system into their compli- ance workflows to identify suspicious accounts dur- ing Know-Your-Transaction (KYT) checks. Reg- ulators and auditors benefit from the XAI mod- ule, which provides clear explanations for risk clas- sifications, supporting investigations and report- ing. Blockchain forensic analysts can use Crypto- Track to explore laundering patterns, trace multi-hop transactions, and identify illicit networks such as darknet market operations, ransomware flows, or coordinated fraud schemes.

Challenges and Limitations

Despite its strengths, CryptoTrack faces several challenges. Blockchain data is pseudonymous, making attribution of identities difficult with- out external metadata. Publicly available AML datasets are limited and often imbalanced, caus- ing difficulties in model training and evaluation. Graph-based neural networks can be computation- ally expensive when processing large-scale trans- action networks, requiring significant memory and optimized sampling. Cross-chain laundering re- mains an emerging challenge due to inconsistent data structures across networks. Finally, although XAI models improve interpretability, explaining graph-based deep models remains a complex and evolving problem.

Future Scope

Future advancements for CryptoTrack include im- plementing cross-chain graph embeddings, inte- grating privacy-preserving techniques such as fed- erated learning, and developing real-time AML monitoring using streaming GNNs. Additional op- portunities involve enhancing interpretability for graph-based decisions, incorporating large lan- guage models for automated suspicious activity re- port (SAR) generation, and collaborating with reg- ulatory bodies to refine laundering typologies and improve practical adoption.

V. CONCLUSION

CryptoTrack presents a unified and explainable framework for detecting cryptocurrency laundering by combining graph learning, temporal sequence modeling, and transparent interpretability. The system addresses limitations of traditional AML tools by offering adaptive learning, reduced false positives, and clear justification for flagged trans- actions. As the cryptocurrency ecosystem contin- ues to evolve, the integration of scalable graph an- alytics and XAI will be essential for building trust-worthy and effective AML systems. CryptoTrack advances this vision by bridging the gap between high-performance detection models and the regu- latory need for transparency.

REFERENCES

1. M. R. Raja et al., “Detecting and Preventing Money Laundering Using Deep Learning and Graph Analysis,” IJACSA, 2025.
2. D. Kute et al., “Deep Learning and Ex- plainable AI Techniques for Detecting Money Laundering,” IEEE Access, 2021.
3. F. Irshad et al., “GCF-MLD,” IEEE Access, 2024.
4. I. Alarab et al., “Graph-Based LSTM,” Neu- ral Processing Letters, 2023.
5. S. Lundberg et al., “SHAP,” NeurIPS, 2017.
6. A. Eddin et al., “AML Graph ML,” arXiv, 2021.
7. M. Weber et al., “GCN AML,” KDD, 2019.
8. T. Pham et al., “Bitcoin Anomaly Detection,” Applied Intelligence, 2021.
9. P. Monamo et al., “Bitcoin Fraud Detection,” IEEE ISSA, 2016.
10. H. Atrak et al., “GNN AML,” IEEE TNNLS, 2024.
11. S. Yang et al., “Blockchain GNN,” IEEE TCSS, 2022.
12. G. Chen et al., “AML Survey,” ACM, 2023.
13. Y. Li et al., “Cross-chain AML,” Info Sciences, 2023.
14. M. Rahman et al., “AML-XAI,” Journal of Financial Crime, 2023.
15. Chainalysis, “Crypto Crime Report,” 2024.