I. J. Computer Network and Information Security, 2026, 3, 63-84
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.03.04

Forensically Interpretable Graph Descriptors for Improved Illicit Bitcoin Transaction Detection

Medet Shaizat*
Faculty of Information Technology and Artificial Intelligence, Al-Farabi Kazakh National University, 050040, Almaty, Kazakhstan
E-mail: m10.invictus@gmail.com
ORCID ID: https://orcid.org/0000-0002-1651-8205
*Corresponding author

Shynar Mussiraliyeva
Faculty of Information Technology and Artificial Intelligence, Al-Farabi Kazakh National University, 050040, Almaty, Kazakhstan
E-mail: Shynar.Musiralieva@kaznu.edu.kz
ORCID ID: https://orcid.org/0000-0001-5794-3649

Ihor Tereikovskyi
System Programming and Specialized Computer Systems Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 03056, Kyiv, Ukraine
E-mail: terejkowski@ukr.net
ORCID ID: https://orcid.org/0000-0003-4621-9668

Received: February 1, 2026; Revised: April 2, 2026; Accepted: May 15, 2026; Published: June 8, 2026

Abstract: The Elliptic dataset is widely used in Bitcoin anti-money laundering research, yet its original anonymized features have limited forensic interpretability. Much of the existing Elliptic-based literature relies on these opaque benchmark variables, leaving insufficient attention to semantically explicit and interpretable graph representations for illicit transaction detection. To address this gap, this article proposes a combined approach that integrates transaction-level feature reconstruction with interpretable forensic descriptor engineering. First, the benchmark's original feature space is replaced with a semantically explicit reconstructed representation derived from public on-chain transaction data and metadata after resolving benchmark node identifiers to transaction hashes. Second, the proposed approach extends this reconstructed representation with interpretable forensic descriptors that capture local transaction abnormality, outgoing value redistribution behavior, and deviations from upstream transaction history. The empirical design isolates the contribution of the proposed descriptors by comparing the reconstructed representation against its descriptor-augmented variant. Across eight classifiers evaluated under a whole-snapshot train-test protocol that preserves within-snapshot graph structure, the descriptor-augmented representation consistently improves illicit class retrieval. CatBoost achieves the best results, increasing the area under the precision recall curve for the illicit transaction class from 85.10% to 90.27%, precision from 77.49% to 87.04%, recall from 75.57% to 81.11%, and F1-score from 76.44% to 83.90%. The article also discusses how the predictive component can be embedded into a hybrid analytical framework that separates machine learning classification from address-level forensic interpretation. This structure supports explainable prioritization and expert review while preserving the distinction between predictive evidence and forensic interpretation. Overall, the findings demonstrate that semantically explicit and forensically interpretable representations can substantially improve illicit transaction retrieval while supporting transparent post hoc analysis in Bitcoin anti-money laundering research.

Index Terms: Cryptocurrency, Bitcoin, Anti-Money Laundering, Illicit Transaction Detection, Forensic Features, Graph Descriptors, Interpretable Machine Learning

1. Introduction

Virtual assets have evolved from experimental digital instruments into a durable component of contemporary financial infrastructure, supporting trading, settlement, and cross-border value transfer across blockchain-based

ecosystems. Their legitimate utility, however, coexists with persistent exposure to illicit financial activity. The same properties that enable global accessibility, rapid transfer, reduced reliance on intermediaries, and pseudonymous addressing also facilitate money laundering, sanctions evasion, ransomware monetization, fraud, and related forms of financial crime [1,2]. This policy relevance is reflected in the Financial Action Task Force (FATF) 2025 targeted update, which identifies continuing implementation gaps in areas such as virtual asset service provider (VASP) licensing and registration, offshore provider oversight, and Travel Rule compliance [2]. Industry reporting points in the same direction: Chainalysis estimates that illicit cryptocurrency addresses received at least USD 154 billion in 2025, with sanctioned entities accounting for a substantial share of this activity [1]. The money-laundering dimension is especially important, as the on-chain laundering ecosystem reportedly expanded from roughly USD 10 billion in 2020 to more than USD 82 billion in 2025 [3]. Taken together, these developments indicate that crypto-enabled illicit finance remains a significant anti-money laundering (AML) governance problem for regulators, financial institutions, exchanges, and investigators.

Bitcoin is a particularly important setting for studying this problem because its public unspent transaction output (UTXO) ledger records value flows in a transparent and temporally ordered form. Suspicious behavior is rarely evident from a single transaction considered in isolation. A transaction may appear ordinary as a ledger record yet become analytically significant when interpreted through upstream provenance, downstream redistribution, local graph structure, and the behavior of participating addresses. In the research literature, the Elliptic dataset has become the canonical public benchmark for illicit Bitcoin transaction detection. Weber et al. introduced Elliptic as a time-series transaction graph with licit, illicit, and unknown labels, providing a shared basis for supervised learning and graph-based benchmarking in crypto AML research [4]. However, the benchmark has an important limitation from a forensic interpretability perspective: its original node features are anonymized and lack explicit semantic meaning. This limits the benchmark's direct applicability in operational AML contexts, where model outputs need to be explainable, auditable, and actionable.

This representational limitation motivates the present study. Two considerations are central to the proposed reformulation of Elliptic-based illicit transaction detection. First, the original anonymized variables do not encode transaction semantics in a form that can be directly verified, audited, or linked to recognizable on-chain behavior, consequently, model outputs are difficult to connect to transaction-level investigative meaning, even though the Elliptic benchmark remains useful for comparative model evaluation. Second, the benchmark exhibits a marked imbalance between licit and illicit labels, so model evaluation should prioritize illicit class retrieval rather than aggregate accuracy. Accordingly, this study formulates illicit transaction detection as supervised node classification on a transaction graph, replacing the original anonymized feature space with a reconstructed transaction-level representation and extending it with interpretable forensic descriptors.

To operate this reformulation, this study uses Elliptic as the benchmark transaction graph, temporal partitioning scheme, and source of original class annotations, but not as a fixed feature benchmark. After resolving benchmark node identifiers to public Bitcoin transaction hashes, the study reconstructs a semantically explicit transaction-level representation from public on-chain data and derived metadata [5]. The original anonymized Elliptic feature space is excluded from the empirical pipeline. On top of this reconstructed representation, the study defines a set of forensically interpretable descriptors capturing local transaction abnormality, outgoing value redistribution, temporal dispersion, and deviations from upstream transaction history. The empirical design compares two nested representations, namely a reconstructed base feature set and a descriptor-extended variant, to isolate the incremental contribution of the proposed descriptors. Model assessment follows AML priorities by emphasizing illicit class PR-AUC, precision, recall, and F1 across eight classifiers under a whole-snapshot train-test protocol, rather than relying on aggregate accuracy alone. In parallel, the study conservatively extends label coverage by assigning labels to previously unknown transactions only where an external blockchain analytics source [6] provides explicit licit or illicit designations, while ambiguous cases remain unlabeled.

This study seeks to address the following research question: To what extent can semantically explicit and forensically interpretable graph representations improve illicit transaction retrieval in Bitcoin AML classification while supporting post hoc investigative interpretation?

The main contributions of this paper are:

1) It proposes a hybrid forensic graph representation approach for the Elliptic benchmark, in which transaction-level reconstruction is used as a foundation for descriptor-based graph and forensic feature engineering. The approach resolves anonymized benchmark nodes to public Bitcoin transaction hashes, excludes the original opaque Elliptic features from the empirical pipeline, and builds a semantically explicit representation from public on-chain metadata to support interpretable illicit transaction detection.
2) It develops a set of forensically interpretable descriptors that characterize illicit-transaction-relevant behavior across multiple analytical dimensions. These descriptors include transaction-level attributes, graph structural topology features, neighborhood forensic features based on robust abnormality and energy measures, outgoing temporal dispersion features, and graph autoregressive lineage features and residuals that capture deviations from upstream transaction history.

3) It demonstrates the empirical value of the proposed interpretable descriptors through a controlled comparison between a reconstructed transaction-level baseline and its descriptor-extended variant across eight classifiers. Under a whole snapshot train and test protocol, the descriptor-augmented representation consistently improves illicit class retrieval, with CatBoost achieving the best results and increasing illicit class PR-AUC from 85.10% to 90.27%, precision from 77.49% to 87.04%, recall from 75.57% to 81.11%, and F1-score from 76.44% to 83.90%.
4) It strengthens post hoc interpretability and forensic applicability by combining conservative label extension, global and local feature attribution analysis, and a hybrid analytical framework that links supervised transaction classification with address level forensic interpretation.

The remainder of the article is organized as follows. Section 2 reviews related work on Bitcoin graph analysis, illicit transaction detection, explainable AML modeling, and blockchain forensics. Section 3 describes the proposed methodology. Section 4 presents experimental results. Section 5 discusses the implications and limitations of the proposed approach. Section 6 concludes the article.

2. Literature Review

Research on illicit Bitcoin transaction detection has developed around the idea that blockchain activity can be represented as a transaction network in which risk is shaped not only by isolated transaction attributes, but also by value-flow dependencies, temporal behavior, and local graph structure. From a forensic perspective, however, blockchain analysis is not simply a graph mining task, its evidentiary value depends on integrating transaction topology, value-flow behavior, temporal context, and, where available, attribution-related information. The Elliptic benchmark introduced by Weber et al. [4] remains the central public reference point for this line of research because it provides a temporally ordered Bitcoin transaction graph with licit, illicit, and unknown labels. At the same time, its anonymized feature space creates a persistent interpretability limitation: although the benchmark supports supervised learning and graph-based evaluation, the original node variables cannot be directly mapped to investigator-facing transaction semantics. This limitation is important because blockchain forensic analysis requires not only predictive discrimination, but also explanations that can be related to observable transaction behavior. Turner et al. [7] emphasize that illicit Bitcoin analysis depends on combining transaction-flow reasoning, graph analysis, and contextual forensic interpretation, which reinforces the need for models whose inputs and outputs are understandable in investigative settings.

Several subsequent datasets and benchmark extensions have attempted to broaden the empirical basis of Bitcoin AML research. Elliptic++ extends the original Elliptic setting by incorporating transaction-, wallet-, address-, and entity-level perspectives, thereby supporting the analysis of illicit transactions, illicit addresses, and user entities in the Bitcoin network [8]. Other public resources, such as HBTBD, introduce heterogeneous transaction and wallet behavior representations to support anti-money laundering analysis beyond a single homogeneous transaction graph [9]. These datasets are valuable because they expand the available information for supervised blockchain forensics. However, they do not fully resolve two structural constraints that remain central to crypto AML research: reliable illicit/licit labels are scarce, and many benchmark features or learned representations remain difficult to interpret in direct forensic terms.

Intermediate research from 2020 to 2023 addressed several practical limitations of Elliptic-style illicit transaction detection. Lorenz et al. [10] studied AML detection under label scarcity and showed that active learning can reduce the amount of labeled data required for effective model training. Alarab et al. [11] compared supervised learning methods for Bitcoin AML classification and confirmed that classical machine learning models remain relevant for this benchmark setting. Alarab and Prakoonwit [12] later introduced a graph-based LSTM approach using temporal graph convolutional representations, highlighting the importance of sequential dependencies in Bitcoin transaction graphs. Lo et al. [13] proposed Inspection-L, a self-supervised GNN-based framework for learning node embeddings for money-laundering detection in Bitcoin. Mohan et al. [14] investigated evolving graph convolutions combined with deep neural decision forests for improving Bitcoin AML classification. FraudLens further shifted attention toward graph structural learning and the effects of graph topology and imbalance on illicit activity identification [15]. Together, these studies show that the post Weber literature evolved along several main directions: label efficient learning, supervised benchmarking, temporal graph modeling, self-supervised representation learning, and graph structural correction.

The theoretical foundation of these approaches is closely related to graph representation learning. Graph convolutional networks formalize the aggregation of node attributes and local neighborhood structure for semi-supervised node classification [16]. GraphSAGE extends this principle through inductive neighborhood aggregation, enabling node representations to be generated from sampled local neighborhoods rather than only from a fixed transductive graph [17]. These ideas are highly relevant to Bitcoin AML because transaction risk may emerge from upstream provenance, downstream redistribution, local connectivity, and temporal flow behavior. In Bitcoin, directed transaction dependencies define upstream provenance and downstream redistribution patterns, making lineage-aware and redistribution-aware descriptors particularly relevant for AML analysis. However, learned embeddings and deep

graph representations do not automatically provide analyst-facing explanations. Even when they improve predictive performance, their internal representations may remain difficult to translate into explicit forensic concepts.

Interpretability is therefore a central requirement in AML-oriented machine learning. General explainable AI methods such as LIME and SHAP provide influential post hoc mechanisms for explaining individual predictions through locally faithful surrogate approximation and additive feature attribution [18,19]. AML-focused reviews also emphasize that transparency, auditability, and human-understandable decision support are essential when machine learning systems are used for suspicious activity detection [20]. In finance more broadly, explainable AI is increasingly treated as a condition for trust, regulatory accountability, and responsible model deployment [21]. Nevertheless, post hoc explanation alone is not sufficient when the original feature space is anonymized or weakly aligned with investigative transaction semantics. In such cases, explanation methods may identify influential variables without clarifying what those variables mean as observable transaction behavior. This distinction is especially important for crypto AML, where explanations must be connected not only to model internals, but also to transaction-level forensic reasoning.

Recent applied work has also examined feature selection to improve illicit Bitcoin transaction classification. Shaizat and Mussiraliyeva [22] proposed feature selection approach based on a genetic algorithm for identifying informative attributes in Elliptic and Elliptic++ settings, showing that carefully selected feature subsets can improve detection performance while reducing model complexity. This direction is relevant because it highlights the importance of feature quality and dimensionality control in Bitcoin AML classification. However, feature selection and learned representation methods do not fully address the representational problem created by anonymized benchmark variables. A remaining gap is the design of semantically explicit, graph-aware, and forensically interpretable descriptors that can both improve illicit class retrieval and support post hoc investigative reasoning.

This gap motivates the present study. Existing literature has made important progress in supervised classification, temporal graph learning, self-supervised embeddings, graph structural preprocessing, and feature selection [10-15,22]. Comparatively less attention has been given to replacing anonymized benchmark variables with reconstructed on-chain transaction attributes and extending them with explicit forensic descriptors. The present work addresses this gap by formulating illicit transaction detection on a semantically explicit feature representation and by introducing interpretable descriptors that capture local transaction abnormality, outgoing value redistribution behavior, temporal dispersion, and deviations from upstream transaction history.

3. Methodology

This section describes the methodological pipeline adopted in the study. It first introduces the data sources, the reconstruction of semantically explicit transaction-level features, and the conservative label reconciliation procedure applied to the Elliptic benchmark. It then presents the transaction graph formulation and the proposed feature engineering framework, followed by the comparative model training strategy, explainability analysis, experimental protocol, and computational environment used for evaluation.

3.1. Dataset

This study uses the Elliptic dataset [4,8] as the benchmark transaction graph. It comprises 203,769 transactions (nodes) and 234,355 directed payment edges and is organized into 49 discrete time steps. In the original release, each node is assigned to one of three classes: licit, illicit, or unknown. Specifically, 4,545 transactions (approximately 2%) are labeled as illicit and 42,019 (approximately 21%) as licit, while the remaining transactions are unlabeled [4]. The original benchmark also provides 166 anonymized numerical node-level attributes [4]. These anonymized variables are not used in the present study. Instead, the analysis uses the Elliptic graph structure, node identifiers, temporal partitioning, and original class annotations, while replacing the benchmark's original feature space with a reconstructed semantically explicit representation.

To recover public transaction identifiers, we use the Kaggle dataset *Deanonymized 99.5 pct of Elliptic transactions*, which provides a mapping between Elliptic node IDs and Bitcoin transaction hashes and covers 99.5% of graph nodes [23]. After resolving node identifiers to transaction hashes, we reconstruct explicit transaction-level variables from public blockchain data. Using the Blockchair API, we collect on-chain attributes such as transaction timestamp, block height, fee, the number of inputs and outputs, and network-level indicators describing the state of the blockchain at the time of inclusion [5]. These variables define the reconstructed base feature set used in the empirical analysis.

On top of this reconstructed base representation, we compute an additional set of forensically interpretable descriptors capturing neighborhood deviation, outgoing temporal dispersion, and graph autoregressive lineage dynamics with residuals. The experimental design therefore compares two nested feature sets: the reconstructed base representation and a descriptor-extended representation that appends these forensic features. This comparison isolates the incremental contribution of the added descriptors independently of the original Elliptic variables, which are excluded from all experiments.

The study also extends label coverage for previously unlabeled transactions. Using the Crystal Blockchain service, we retrieve explicit licit/illicit designations where available [6]. Label transfer is performed conservatively and applies

only to transactions assigned to the unknown class in the original Elliptic benchmark. For transactions already labeled as licit or illicit, the original Elliptic annotations are preserved regardless of whether an external assessment is available. Likewise, transactions for which Crystal provides only intermediate or otherwise ambiguous risk indications, such as a medium risk designation without an explicit licit/illicit classification, remain in the unknown class. Under this protocol, 2,304 previously unknown transactions receive explicit labels. As table 1 shows, the resulting graph contains 48,868 labeled transactions in total, including 44,206 licit and 4,662 illicit observations, while the remaining 154,901 transactions retain the unknown label.

Table 1. Label reconciliation rules.

Case	Original Elliptic label	Crystal assessment	Decision rule	Final label	Count
1	licit	any / missing	retain original Elliptic label	licit	42,019
2	illicit	any / missing	retain original Elliptic label	illicit	4,545
3	unknown	explicit licit attribution	relabel from Crystal	licit	2,187
4	unknown	explicit illicit attribution	relabel from Crystal	illicit	117
5	unknown	ambiguous / intermediate / no explicit class	keep as unknown	unknown	154,901

The original chronological Elliptic split remains methodologically appropriate when the objective is forward in time robustness, particularly because performance may deteriorate after major external events such as the dark market shutdown reported in the original study [4,23]. In the present work, however, the predictive setting is different and does not reproduce the original Elliptic benchmark one-to-one. The final representation does not directly reuse the original Elliptic feature matrix. Instead, the original graph structure, labels, and timestep assignment are combined with externally collected transaction-level attributes and graph-derived descriptors computed separately for each snapshot. The timestep index itself is not used as a predictor, and raw block height is excluded as a standalone input variable, although several engineered descriptors retained in the final representation remain defined in block units, such as age-related transaction characteristics and local lag-based quantities derived from block height differences between adjacent transactions. Accordingly, the data are more appropriately interpreted as a collection of 49 schema consistent graph snapshots rather than as a strict chronological forecasting benchmark. The implications of this representation for the train-test partitioning protocol are discussed in the Experimental Setup section.

3.2. Transaction Graph Model and Feature Engineering

Feature family design was guided by a forensic-oriented exploration analysis of recurrent motifs in the labeled illicit subset and by prior empirical studies of Bitcoin transaction behavior and illicit finance typologies [4,24,25]. Specifically, we observed that illicit activity can manifest as bursty, temporally uneven outward branching multiple spends clustered within short block height intervals motivating outgoing temporal dispersion descriptors. We further exploit the fact that the UTXO spend relation induces a causally ordered provenance structure, consequently, modeling scalar transaction attributes in the context of upstream ancestry can provide a principled baseline against which deviations become measurable [24,25]. This motivates the graph autoregressive lineage (GAR) construction, which propagates node-level attributes along incoming edges to derive multi-hop lineage lags, a lineage-consistent fitted component, and a residual that captures departures from lineage-consistent behavior. Together with structural topology measures, transaction-level attributes, and robust neighborhood deviation statistics, these feature families operationalize domain-informed forensic hypotheses while remaining fully computable from transaction graphs and associated attributes. For readability, this section provides only a conceptual overview of the computation of each feature block, whereas the complete list of features, their functional definitions, and the corresponding formulas are provided in tabular form in Appendix A.

The study addresses illicit activity detection as a supervised node-level classification problem on a directed transaction network observed over discrete temporal slices. Following established practice in blockchain transaction modeling [25], for each time step τ, the transactional system is represented as a directed graph $G_\tau = (V_\tau, E_\tau)$, where each node $v \in V_\tau$corresponds to a transaction and each directed edge $(u \to v) \in E_\tau$encodes a dependency/flow relation between transactions. Each node is associated with a timestamp $h(v)$(implemented as block_height) and, when available, a class label indicating illicit or licit behavior. Labels are mapped to a binary target variable by treating illicit as the negative class and licit as the positive class, i.e., $y(v) = 0$ for illicit and $y(v) = 1$ for licit. Nodes with unknown labels are excluded from supervised training and evaluation, however, they are presented in the graph and can influence purely structural descriptors.

In this study, we propose constructing the feature vector from the following five blocks:

(i) transaction-level attributes
(ii) structural topology features on G_t,
(iii) neighborhood forensic features (robust $z$and energy),

(iv) outgoing temporal dispersion features based on block heights, and
(v) graph autoregressive lineage (GAR) features and residuals.

Formally, for a labeled node $v \in V_t$, the final feature vector is defined as

$$\mathrm{x}(v) = [x_v^{tx}; x_v^{topo}; x_v^{GAR}; x_v^{nbr}; x_v^{temp\text{-}out}] \quad (1)$$

where:

x_v^{tx} denotes transaction-level attributes.
x_v^{topo} denotes structural-topological features on the snapshot G_t.
x_v^{GAR} denotes graph autoregressive lineage features and residuals.
x_v^{nbr} denotes neighborhood forensic features (robust z, energy).
$x_v^{temp\text{-}out}$ denotes temporal dispersion of outgoing transfers in terms of block heights.

3.2.1. Transaction-level and Structural Features

To characterize both the intrinsic properties of a transaction and its structural role within the contemporaneous Bitcoin network, we use two complementary groups of features: transaction-level covariates and structural topology descriptors. This distinction is natural in AML-oriented blockchain analysis, where node-specific behavioral attributes and graph-derived relational patterns often provide complementary evidence for distinguishing licit and illicit activity [4].

For each transaction $v \in V_t$, the transaction-level feature block x_v^{tx} includes numerical covariates describing its economic and operational profile independently of graph connectivity. These attributes comprise transferred value, the numbers of inputs and output addresses, transaction size, fee related measures, concentration indicators, and summary statistics of input/output amounts and age measured in blocks. Such descriptors are widely used in cryptocurrency AML settings because they capture spending regularities and transaction composition that are not directly encoded in network structure [4].

In designing the structural feature block, we adapt graph descriptors that have been extensively used in the analysis of social networks, complex networks, and other large-scale relational systems to characterize node role, connectivity, and embeddedness. This choice is motivated by the fact that Bitcoin transaction graphs, despite their domain-specific semantics, also exhibit heterogeneous local structure and nontrivial mesoscale organization, making established topological measures a principled basis for structural representation [26-30]. The structural feature block x_v^{topo} is therefore designed to capture the position of $v$within the snapshot G_t. Specifically, we compute in-degree and out-degree, PageRank, HITS authority and hub scores, average neighbor degree and weakly connected component size, k-core number, and eigenvector centrality. Directed measures are evaluated on G_t, whereas average neighbor degree, weakly connected component size, k-core number, and eigenvector centrality are computed on the undirected projection of the snapshot. Together, these features provide a compact summary of both transaction behavior and graph embeddedness and serve as the base representation for the subsequent neighborhood forensic and lineage-based feature blocks.

3.2.2. Neighborhood Forensic Features: Robust Deviation and Local Discrepancy

As is well known, bitcoin transactions are embedded in directed spending relations induced by the UTXO model, consequently, the immediate neighborhood of a transaction contains informative short-range evidence on how value is received, consolidated, and redistributed [24]. Building on this intuition, we introduce neighborhood forensic features to characterize each transaction relative to its local directed environment while preserving the distinction between incoming and outgoing relations. This feature family is designed to capture local forensic behavior that is not directly encoded by global graph topology or transaction-level attributes alone. The central idea is to evaluate each transaction relative to its immediate directed neighborhood, while preserving the distinction between incoming and outgoing relation. For a node $v \in V_t$, let $N_{\mathrm{in}}(v)$and $N_{\mathrm{out}}(v)$ denote its incoming and outgoing neighbors in the snapshot G_t, and let all neighborhood aggregations be computed over unique directed relations.

For any numerical transaction attribute $s(v)$, we first quantify contextual deviation using a robust neighborhood-normalized score based on the median and the median absolute deviation (MAD), rather than the conventional mean and standard deviation, since the latter are substantially more sensitive to outliers and heavy-tailed observations [31,32]. For non-empty neighborhoods and $\mathrm{dir} \in \{\mathrm{in}, \mathrm{out}\}$, we define:

$$z_{\mathrm{dir}}^{\mathrm{rob}}(v) = \frac{s(v) - \mathrm{med}_{\mathrm{dir}}(v)}{\max(c\ \mathrm{MAD}_{\mathrm{dir}}(v), \kappa(s))}, \quad (2)$$

where $\mathrm{med}_{\mathrm{dir}}(v)$and $\mathrm{MAD}_{\mathrm{dir}}(v)$are computed over $N_{\mathrm{dir}}(v)$, $\kappa(s)$prevents numerical instability when local dispersion is very small. The constant $c = 1.4826$ is the Gaussian consistency factor for MAD, i.e., the factor that renders the scaled MAD asymptotically consistent for the standard deviation under a normal reference model [32]. This score therefore captures how atypical a transaction is relative to the behavioral center of its direct neighborhood while remaining stable under extreme values and heterogeneous attribute scales.

To complement relative deviation, we further quantify neighborhood discrepancy through the mean squared departure from the same directed neighbor set:

$$E_{\mathrm{dir}}(v) = \frac{1}{d_{dir}} \sum_{u \in N_{\mathrm{dir}}(v)} (s(v) - s(u))^2, \tag{3}$$

This term captures the local heterogeneity of neighboring transactions. Conceptually, it may be viewed as a localized and direction-aware analogue of graph-signal variation or edgewise smoothness measures used in graph signal processing [33,34]. The final vector $\mathrm{x}_v^{\mathrm{nbr}}$is obtained by concatenating robust-deviation and local-discrepancy descriptors across selected transaction attributes for both incoming and outgoing neighborhoods.

3.2.3. Outgoing Temporal Dispersion Features

Bitcoin transactions are naturally embedded in blockchain time through their inclusion in sequentially ordered blocks, making block height a practical monotone proxy for transaction chronology [24,25]. Analyzing transaction activity only in aggregated form may obscure short-term behavioral changes, while the temporal network literature emphasizes that the ordering and spacing of interactions can materially affect the observed dynamics of a system [35]. Moreover, event sequences in complex systems often exhibit bursty rather than uniform timing [36,37]. Motivated by these considerations, we introduce outgoing temporal dispersion features to characterize how rapidly and how diffusely a transaction redistributes value over blockchain time.

Let $O(v)$denote the set of outgoing neighbors of transaction $v$in the snapshot G_t, and let:

$$k_{\mathrm{out}}(v) = \mid O(v) \mid \tag{4}$$

be the number of outgoing transfers. For each outgoing edge (v, u), we compute the block height lag:

$$\Delta_{v \to u} = bh(u) - bh(v) \tag{5}$$

which represents the temporal offset between the source transaction and its downstream transfer in units of block progression. From the empirical distribution of these lags, we extract summary statistics such as the median, standard deviation, and minimum, which capture the typical delay, temporal variability, and earliest downstream propagation, respectively.

To quantify the overall temporal extent of outgoing activity, we define:

$$\Delta t(v) = \begin{cases} \max_{u \in O(v)} bh(u) - \min_{u \in O(v)} bh(u), & k_{\mathrm{out}}(v) \geq 2, \\ 0, & k_{\mathrm{out}}(v) < 2. \end{cases} \tag{6}$$

This quantity measures how widely outgoing transfers are spread over blockchain time. However, direct rate-like statistics based on $\Delta t(v)$can become unstable when multiple transfers are concentrated within a very short block interval. We therefore introduce smooth regularization:

$$\Delta t_{\mathrm{reg}}(v) = \Delta t(v) + \tau \ln\left(1 + (e-1)\exp\left(\frac{\Delta t(v)}{\tau}\right)\right), \tag{7}$$

where $\tau > 0$is a scale parameter controlling the transition between short span stabilization and large span asymptotics. Based on this regularized span, we define a rate-like outgoing temporal intensity:

$$\rho(v) = \frac{(k_{\mathrm{out}}(v) - 1)^{\alpha}}{(\Delta t_{\mathrm{reg}}(v))^{\beta}}, \tag{8}$$

followed by its stabilized logarithmic transform:

$$\phi_{\mathrm{out}}(v) = \ln(1 + \tau\, \rho(v)), \tag{9}$$

The quantities in (7)-(9) are introduced in this study as a stabilized descriptor of outgoing temporal dispersion, conceptually motivated by the temporal ordering of blockchain interactions and the bursty concentration of transactional activity [35-37]. They jointly encode two complementary aspects of outgoing behavior: the temporal span over which downstream transfers are distributed and the intensity with which multiple transfers are concentrated within that span. For sufficiently large temporal spans, the resulting descriptor behaves as a rate type measure, whereas for near zero spans it remains well-behaved and avoids pathological inflation. Accordingly, $\mathbf{x}_v^{\text{temp-out}}$ captures whether value redistribution is temporally concentrated or temporally diffuse, thereby complementing graph-topological and neighborhood-forensic representations.

3.2.4. Graph Autoregressive Lineage (GAR)

In Bitcoin, edges encode directed value transfer rather than generic proximity, so each transaction is embedded in an upstream lineage of fund propagation. This makes incoming provenance especially relevant for financial forensics, where graph-based methods have already shown that relational structure is informative for illicit transaction detection. Motivated by this observation, for a scalar transaction-level signal $x(v)$ (e.g., size, fee, concentration, or value related attributes), we define a graph autoregressive lineage (GAR) representation that models $x(v)$ through its directed incoming ancestry.

$$\hat{x}(v) = b_0 + \sum_{k=1}^{P_{\max}} \varphi_k \, x^{(k)}(v), \tag{10}$$

and the corresponding residual

$$r(v) = x(v) - \hat{x}(v). \tag{11}$$

where $x^{(k)}(v)$ represents an averaged k-hop ancestor signal propagated along incoming edges of the transaction graph.

To construct k-hop lineage lags, we define

$$x^{(0)}(v) = x(v) \tag{12}$$

Then, for $k \geq 1$, the k-step incoming lineage lag is defined recursively as

$$x^{(k)}(v) = \begin{cases} \dfrac{1}{d_v^-} \displaystyle\sum_{u \in N^-(v)} x^{(k-1)}(u), & d_v^- > 0, \\ 0, & d_v^- = 0. \end{cases} \tag{13}$$

where $N^-(v)$ denotes the set of incoming neighbors of node $v, d^-(v)$ denotes the in-degree of node v.

The coefficients φ_k, b_0 are obtained by minimizing the following objective functional [16]:

$$J = \sum_{v \in \mathcal{V}_{\text{train}}} \left(x(v) - b_0 - \sum_{k=1}^{P_{\max}} \varphi_k \, x^{(k)}(v) \right)^2 + \lambda \sum_{k=1}^{P_{\max}} w_k \, \varphi_k^2, J \to \min_{b_0, \varphi} \tag{14}$$

where $\mathcal{V}_{\text{train}}$ denotes the set of training nodes with finite signal and lag values, $\lambda \geq 0$ is the regularization coefficient, and w_k are nondecreasing depth-dependent weights that impose stronger shrinkage on more distant ancestors. Such regularization stabilizes the estimates and limits the influence of increasingly remote lineage information.

In particular, we use the monotone schedule

$$w_k = \left(1 + \frac{k-1}{P_{\max} - 1} (r - 1) \right)^{\gamma}, \quad k = 1, \ldots, P_{\max}, r \geq 1, \gamma \geq 0, \tag{15}$$

which satisfies $w_1 \leq w_2 \leq \cdots \leq w_{P_{\max}}$.

The weighted ridge objective in (14) is a convex quadratic function, for $\lambda > 0$ (with $w_k > 0$) it is strictly convex and therefore admits a unique global minimizer, obtained by solving the corresponding normal equations [38,39].

These quantities encode complementary aspects of transaction behavior. The depth wise lineage lags summarize how the signal propagates through upstream provenance, the fitted value represents the lineage consistent component of

the signal, and the residual captures deviation from lineage expected behavior. Together, they enrich the representation with interpretable multi-hop dependency information that is not captured by transaction-level, topological, or neighborhood descriptors alone.

3.3. Model Training and Comparative Analysis

Another objective of this study is to evaluate whether the predictive benefit of the proposed feature set is consistent across multiple learning algorithms rather than being an artifact of a single favorable model choice. To this end, we conducted a comparative benchmark using eight tree-based ensemble classifiers with different inductive biases and optimization schemes: CatBoost, LightGBM, HistGradientBoosting GradientBoosting, XGBoost, RandomForest, balanced bagging with decision trees, and ExtraTrees. All models were trained and evaluated on identical pooled training and test splits under fixed priori settings to ensure a fair and controlled comparison. Class imbalance was addressed through model-specific weighting or balanced resampling whenever supported by the corresponding implementation. This design allows us to assess whether the proposed feature representation yields stable improvements across related but distinct learners, rather than improving performance for only a single model.

3.4. Explainability and Feature Relevance

Post-hoc explainability was performed for the best-performing model overall, selected according to the primary comparison criterion reported in Results section. In our experiments, this model was CatBoost. Global feature relevance was assessed using CatBoost PredictionValuesChange importances. In parallel, SHAP values [19] were computed for the same trained model on up to 20,000 observations sampled from the pooled held-out test set in each run. Class-specific SHAP attributions were obtained for both the illicit and licit outputs and aggregated into mean absolute SHAP scores, which were further visualized using summary and bar plots. Relevance estimates were finally summarized across runs as mean ± standard deviation and are interpreted as associations with the fitted decision function rather than as causal effects.

3.5. Experimental Setup

All supervised experiments were performed on the labeled subset obtained after the inner join with the external transaction-level feature table, transactions with the unknown label were excluded. The evaluation protocol comprised ten repeated random splits of the 49 graph snapshots using split seeds 42–51. In each run, whole snapshots rather than individual nodes were assigned to the training or test partition with a 70/30 ratio, and both licit and illicit classes were required to be present in both partitions. Since the original chronological Elliptic split was not enforced, the per snapshot analysis is interpreted as cross-snapshot stability rather than temporal forecasting.

For each snapshot, graph-derived features were computed independently on the full intra-snapshot transaction graph and then restricted to labeled nodes for supervised learning. Unlabeled nodes could contribute to topology, but no class labels were used in feature construction. The final representation combined external transaction-level attributes with structural descriptors, including degree-based measures, PageRank, HITS, average neighbor degree, weakly connected component size, k-core number, eigenvector centrality, neighborhood discrepancy features (G1/G2), outgoing block height lag variables, and GAR (Pmax = 2) lineage features. PageRank, HITS, and eigenvector centrality were approximated with 20 fixed iterations. To limit leakage, timestep and raw block height were excluded from the predictive input, graph-derived features were computed separately for each snapshot, and the GAR coefficients were fitted once on pooled training snapshots only and then applied unchanged to both training and test snapshots. Within each run, model performance was evaluated on a pooled test set formed by concatenating the held-out snapshots, and final comparisons were aggregated across repeated runs. Detailed performance measures are reported in Section 4.

3.6. Computational Efficiency and Deployment Feasibility

Experiments were conducted on a 64-bit Windows laptop with an Intel(R) Core (TM) i7-1065G7 CPU @ 1.30 GHz and 16 GB of RAM. The computational cost of the proposed descriptor-extended AML pipeline can be separated into feature extraction and model scoring. Feature extraction is the dominant stage because the added forensic descriptors require local neighborhood aggregation, per-slice temporal statistics, and bounded-depth lineage propagation over the temporal transaction graph. Under the evaluated configuration, where the descriptor set, neighborhood order, and lineage depth are fixed, these operations do not require unrestricted graph expansion. The extraction cost therefore scales approximately linearly with the processed temporal slice, up to constant factors determined by the selected descriptor parameters, i.e., $O(|V_\tau| + |E_\tau|)$. Once the feature vector is constructed, CatBoost scoring is lightweight and requires no additional graph traversal. In the reference CPU-only profile, feature extraction required 67.14 s in total, GAR calibration required 19.34 s. Thus, the total descriptor-construction cost, including GAR calibration, was 86.48 s, or approximately 1.79 ms per transaction. model training required 23.17 s, single-sample inference after feature extraction was 2.05 ms and peak memory usage reached 341 MB. These measurements indicate that the dominant practical cost lies in descriptor construction rather than in model scoring. The reported single-sample inference time therefore refers only to CatBoost scoring after feature extraction and should not be interpreted as full end-to-end latency for an arbitrary newly arriving transaction. Overall, the proposed approach is

most suitable for offline investigation support, periodic compliance screening, and block-level or micro-batch AML monitoring, where graph context can be updated over blocks or short time windows and only affected transactions need to be rescored. However, the current measurements represent a single CPU-only reference profile rather than a full cross-platform benchmark. Stricter real-time deployment would require indexed graph storage, cached adjacency structures, and incremental feature updates, while evaluation on additional hardware environments remains future work.

4. Results

This section presents an empirical comparison between two feature settings. The baseline representation comprised 26 engineered features organized into two groups: transaction-level attributes and structural topology features defined on the transaction graph G_t. The enhanced representation extended this baseline to 87 features by incorporating 61 additional forensic-operator features, grouped into neighborhood forensic descriptors, outgoing temporal-dispersion features, and graph autoregressive lineage (GAR) features and residuals. To preserve readability, this section focuses on predictive performance, whereas the full feature inventory and formal definitions are provided in Appendix A. Throughout this section, the illicit class is encoded as the negative label (y = 0) and the licit class as the positive label (y = 1), consistent with the methodological setup.

Because the primary objective of this study is the reliable identification of illicit transactions, the evaluation emphasizes class-specific performance for the illicit class, particularly the area under the precision–recall curve (AUPRC), the F1-score, and recall. Precision for the illicit class is reported as a complementary indicator of false-positive control, whereas overall accuracy is retained for completeness only. For all class-specific evaluation metrics, illicit was treated as the positive evaluation class. Specifically, AUPRC was computed as average precision using the predicted probability assigned to the illicit label, whereas precision, recall, and F1 were computed after recoding predictions so that illicit corresponded to the positive class. In highly imbalanced AML settings, aggregate measures may overstate apparent model quality while masking insufficient recovery of illicit transactions, accordingly, they are not used as the principal basis for model selection.

4.1. Base

Table 2 reports mean values over 10 runs for the baseline models. CatBoost achieved the highest illicit class PR-AUC (85.10%) and recall (75.57%), exceeding LightGBM's recall (72.43%) by 3.14 percentage points and making it the strongest baseline for illicit-case recovery. LightGBM delivered the highest illicit class F1-score (76.96%) while maintaining competitive PR-AUC (84.35%) and recall (72.43%), making it the most balanced baseline in precision-recall terms. HistGradientBoosting was close, with an F1-score of 76.82% and a PR-AUC of 84.29%, but its lower recall (67.11%) suggests a more conservative operating profile. RandomForest and ExtraTrees achieved very high precision (93.52% and 94.04%) at the cost of substantially lower recall (61.60% and 56.80%), an unfavorable trade-off in AML screening where false negatives are especially costly. Although HistGradientBoosting reached the highest overall accuracy (95.82%), it did not deliver the strongest illicit class retrieval, underscoring that aggregate metrics alone are insufficient for model selection in this setting. Overall, the baseline results favor boosting-based methods: CatBoost is preferable when illicit case recovery and ranking quality are prioritized, whereas LightGBM offers the best precision-recall balance.

Table 2. Performance of baseline models using the 26-feature baseline representation. Values are reported as mean over 10 runs.

Model	PR-AUC (illicit class)	Precision (illicit class)	Recall (illicit class)	F1-score (illicit class)
CatBoost	85.10	77.49	75.57	76.44
LightGBM	84.35	82.26	72.43	76.96
XGBoost	82.93	87.99	65.70	75.10
HistGradientBoosting	84.29	90.14	67.11	76.82
GradientBoosting	81.27	88.86	65.02	74.95
Random Forest	83.38	93.52	61.60	74.15
BaggingTreeBalanced	79.92	89.71	62.75	73.74
ExtraTrees	80.19	94.04	56.80	70.74

4.2. Base + Forensic Operators

Table 3 reports mean values over 10 runs for the same classifiers after augmenting the baseline feature space with forensic operators. In this enhanced setting, CatBoost delivered the best illicit class performance, achieving a PR-AUC of 90.27%, an F1-score of 83.90%, and a recall of 81.11%, making it the strongest model for illicit transaction retrieval. LightGBM was the closest competitor, with a PR-AUC of 89.54%, an F1-score of 83.04%, and a recall of 76.50%. It also achieved slightly higher precision than CatBoost (90.96% vs. 87.04%), indicating a more conservative operating profile with tighter false positive control. However, CatBoost retained a clear advantage in minority class retrieval and ranking quality. XGBoost and HistGradientBoosting formed a second tier, with F1-scores of 80.96% and 80.89%, respectively, but remained below the two leading models in PR-AUC and recall. As in the baseline setting, RandomForest and ExtraTrees maintained very high precision (95.88% and 93.61%) at the cost of substantially lower

recall (65.64% and 59.57%), a less desirable trade-off in AML settings where false negatives are especially costly. Overall, the inclusion of forensic operators further strengthened the dominance of boosting methods, with CatBoost offering the most favorable balance of PR-AUC, F1-score, and recall.

Table 3. Performance of the same models using the 87-feature enhanced representation augmented with forensic operators. Values are reported as mean over 10 runs.

Model	PR-AUC (illicit class)	Precision (illicit class)	Recall (illicit class)	F1-score (illicit class)
CatBoost	90.27	87.04	81.11	83.90
LightGBM	89.54	90.96	76.50	83.04
XGBoost	87.81	93.37	71.56	80.96
HistGradientBoosting	87.82	93.98	71.11	80.89
GradientBoosting	85.12	92.73	68.40	78.69
Random Forest	86.67	95.88	65.64	77.85
BaggingTreeBalanced	82.04	92.57	63.12	74.94
ExtraTrees	83.89	93.61	59.57	72.74

4.3. Comparative Analysis: Baseline vs. Baseline + Forensic Operators

A direct comparison between Tables 2 and 3, Fig. 1 demonstrates that the inclusion of forensic operators led to a clear and consistent improvement in illicit class detection. Averaged across all eight classifiers, the area under the precision–recall curve for the illicit class increased by 3.97 percentage points (from 82.68% to 86.65%), the F1-score by 4.27 percentage points (from 74.86% to 79.13%), and the recall by 3.76 percentage points (from 65.87% to 69.63%), while precision for the illicit class rose by 4.52 percentage points (from 88.00% to 92.52%).

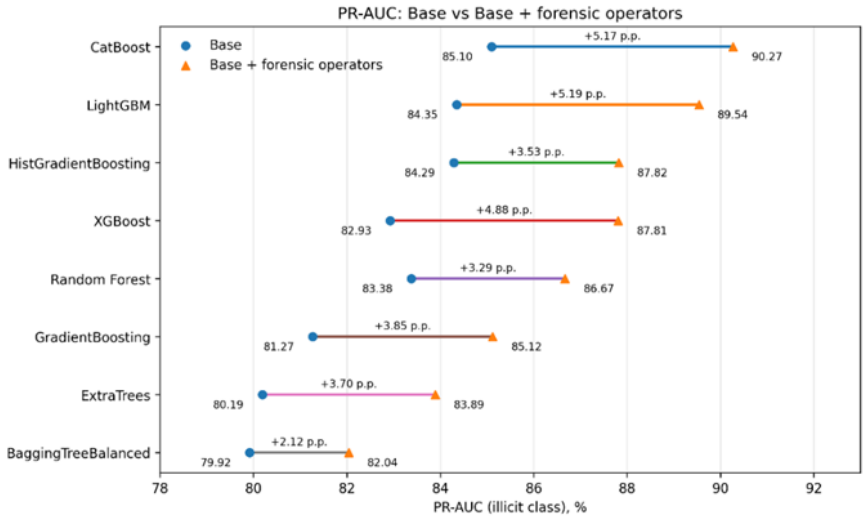

Fig. 1. PR-AUC: Base vs Base + forensic operators.

As shown in Table 4, all evaluated classifiers improved in terms of the area under the precision–recall curve, F1-score, and recall for the illicit class, whereas precision improved for seven of the eight models, with only ExtraTrees showing a marginal decline. The largest increase in F1-score was observed for CatBoost, the largest gain in recall for XGBoost, and the largest improvement in the area under the precision–recall curve for LightGBM. These findings indicate that the forensic operators improved not only illicit-case recovery, but also the overall quality of minority class discrimination. Moreover, the enhanced representation shifted the ranking of the strongest models: whereas LightGBM offered the best F1-score in the baseline setting and CatBoost led in precision-recall ranking quality and recall, the enhanced setting established CatBoost as the overall best-performing classifier across the principal illicit class metrics. Overall, the results show that the addition of forensic operators yields a substantial and practically meaningful benefit, particularly for boosting-based methods, most notably CatBoost and LightGBM.

Table 4. Comparison of PR-AUC and F1-score before and after adding forensic operators. Base and Base+FO values are reported as mean over 10 runs.

Model	PR-AUC (illicit class)			F1-score (illicit class)		
	Base	Base + FO	Δ	Base	Base + FO	Δ
CatBoost	85.10%	90.27%	+5.17%	76.44%	83.90%	+7.46%
LightGBM	84.35%	89.54%	+5.19%	76.96%	83.04%	+6.08%

XGBoost	82.93%	87.81%	+4.88%	75.10%	80.96%	+5.86%
HistGradient Boosting	84.29%	87.82%	+3.53%	76.82%	80.89%	+4.07%
Gradient Boosting	81.27%	85.12%	+3.85%	74.95%	78.69%	+3.74%
Random Forest	83.38%	86.67%	+3.29%	74.15%	77.85%	+3.70%
BaggingTree Balanced	79.92%	82.04%	+2.12%	73.74%	74.94%	+1.20%
ExtraTrees	80.19%	83.89%	+3.70%	70.74%	72.74%	+2.00%

4.4. Comparative Analysis with Graph-Learning Baselines

To assess the relative position of the proposed method, representative graph-learning and hybrid baselines were evaluated on the Elliptic benchmark under the same experimental protocol. The comparison includes conventional GNN architectures, namely GCN, GraphSAGE, APPNP, and H2GCN, a graph transformer model, SGFormer, and hybrid graph representation approaches, GraphMAE + CatBoost and DGI + CatBoost.

Table 5. Comparative performance of the proposed method and representative graph-learning baselines.

Model	**Model family**	**PR-AUC (illicit class)**	**Precision (illicit class)**	**Recall (illicit class)**	**F1-score (illicit class)**
GCN	GNN	0.5343	0.2743	0.7010	0.3943
GraphSAGE	GNN	0.5522	0.3165	0.7233	0.4403
APPNP	GNN	0.5295	0.2809	0.6938	0.3999
H2GCN	GNN	0.5408	0.3344	0.6986	0.4522
SGFormer	Graph transformer	0.6172	0.4327	0.6922	0.5325
GraphMAE + CatBoost	Hybrid graph embedding	0.8592	0.7826	0.7552	0.7687
DGI + CatBoost	Hybrid graph embedding	0.8347	0.7590	0.7209	0.7395
Proposed Descriptor-Extended CatBoost	**Proposed interpretable descriptor model**	**0.9027**	**0.8704**	**0.8111**	**0.8390**

Table 5 presents comparative results position the proposed descriptor-extended CatBoost model favorably against representative graph-learning and hybrid baselines on the Elliptic benchmark. Standalone GNN models, including GCN, GraphSAGE, APPNP, and H2GCN, achieve relatively high recall for the illicit class but substantially lower precision and F1-score, suggesting that their learned neighborhood representations tend to over-detect illicit transactions under strong class imbalance. The graph transformer model SGFormer improves over conventional GNNs, but its illicit class PR-AUC and F1-score remain considerably below those of the proposed model. Hybrid graph representation approaches, such as GraphMAE + CatBoost and DGI + CatBoost, narrow this gap by combining learned graph embeddings with a strong classifier, however, they still underperform the descriptor-extended representation. The proposed model achieves the highest illicit class PR-AUC and F1-score among all compared methods, indicating that semantically explicit transaction reconstruction combined with interpretable forensic descriptors provides more discriminative signals for illicit transaction detection than purely learned graph embeddings in this experimental setting.

4.5. Ablation Analysis of Forensic Descriptor Groups

To further isolate the contribution of each forensic descriptor group, we conducted a group wise ablation study using CatBoost, which was the strongest performing classifier in the main experiments. The full descriptor-extended model includes three groups of forensic descriptors: neighborhood descriptors, temporal dispersion descriptors, and pedigree dynamic descriptors. We then retrained CatBoost after removing one descriptor group at a time while keeping the reconstructed base representation and the remaining descriptor groups unchanged. This design allows the relative contribution of each descriptor group to be estimated from the performance degradation observed after its removal.

Table 6. Ablation study of neighborhood, temporal dispersion, and gar forensic descriptors using CatBoost.

Configuration	**Removed descriptor group**	**PR-AUC (illicit class)**	**Precision (illicit class)**	**Recall (illicit class)**	**F1-score (illicit class)**	**Δ PR-AUC**	**Δ F1-score**
Full model	None	90.27	87.04	81.11	83.90	—	—
Ablation 1	Neighborhood descriptors	86.97	80.82	77.02	78.79	−3.30	−5.11
Ablation 2	GAR	89.18	86.29	79.24	82.56	−1.09	−1.34
Ablation 3	Temporal dispersion descriptors	89.98	86.62	80.44	83.34	−0.29	−0.56

As shown in Table 6, the ablation results show that all three groups of forensic descriptors contribute positively to illicit class detection, but their contributions are not uniform. Removing the neighborhood descriptor group produces the largest degradation, reducing illicit class PR-AUC from 90.27% to 86.97% and F1-score from 83.90% to 78.79%. This indicates that local graph structural abnormality is the most influential component of the proposed forensic representation. Removing pedigree dynamic descriptors also decreases performance, although more moderately, with PR-AUC falling by 1.09 percentage points and F1-score by 1.34 percentage points, suggesting that upstream lineage dynamics provide additional discriminatory information beyond static transaction and neighborhood structure. The smallest performance reduction is observed when temporal dispersion descriptors are removed, with PR-AUC decreasing by 0.29 percentage points and F1-score by 0.56 percentage points. Although this contribution is comparatively weaker, the full descriptor set still achieves the best results across all reported illicit class metrics, confirming that the three descriptor groups are complementary rather than redundant.

4.6. Explainability and Feature Relevance

Because CatBoost was the strongest classifier in the forensic-augmented setting, particularly in terms of the area under the precision-recall curve for the illicit class, the F1-score for the illicit class, and the recall for the illicit class, it was selected as the reference model for post hoc explainability. Accordingly, Shapley Additive explanations (SHAP) were computed only for CatBoost. The results reported below should therefore be interpreted as an explanation of the CatBoost decision function rather than of all evaluated classifiers. This is consistent with the evaluation protocol, in which overall accuracy was treated as supplementary, whereas minority class recovery and ranking quality formed the principal basis for model selection.

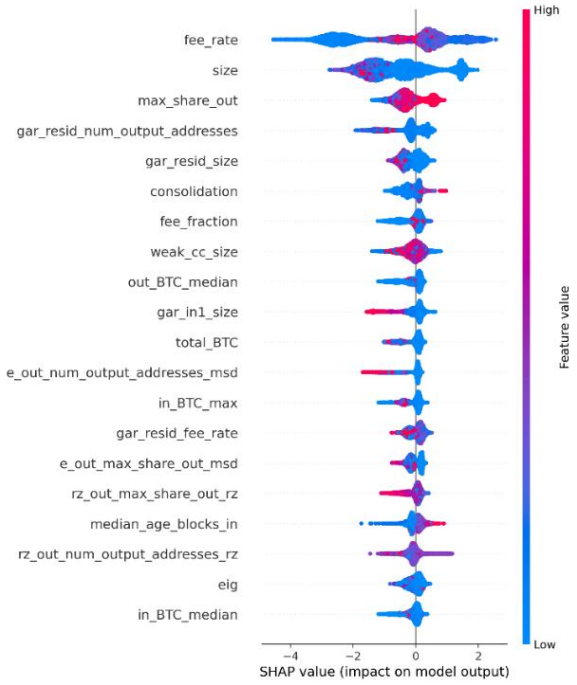

Fig. 2. SHAP values: Top 20 attributes.

The aggregated explainability analysis revealed a stable and internally coherent relevance structure across repeated timestep splits. Across 10 runs, mean absolute SHAP values identified fee_rate and size as the dominant predictors, followed by max_share_out, gar_resid_num_output_addresses, gar_resid_size, and weak_cc_size. A closely aligned ranking was obtained from CatBoost's model-native PredictionValuesChange importance measure, with a strong agreement between the two formalisms (Spearman rank correlation $\approx$ 0.94 across 87 summarized features). This indicates that the reported hierarchy is not an artifact of a single interpretability method.

At the feature family level, the three operator-derived blocks graph autoregressive lineage descriptors, neighborhood forensic features, and outgoing temporal dispersion features jointly accounted for 45.8% of the total mean absolute SHAP relevance and 46.4% of the total PredictionValuesChange importance. Their contribution was driven mainly by graph autoregressive lineage features (22.6% SHAP, 21.0% PredictionValuesChange) and neighborhood forensic features (22.0% SHAP, 24.2% PredictionValuesChange), whereas outgoing temporal dispersion features provided a smaller but complementary contribution (approximately 1.2% under both measures). The most important features, according to SHAP values, are presented in Fig. 2.

4.7. Error Analysis of Misclassified Transactions

Also, we conducted an illustrative case-level inspection of 30 misclassified test transactions produced by the CatBoost model. The subset included 15 false negatives, corresponding to illicit transactions predicted as licit, and 15 false positives, corresponding to licit transactions predicted as illicit. The full list of selected transactions is provided in Appendix B. Because the benchmark labels do not provide fine-grained illicit-activity typologies, missed illicit transactions were characterized through observable graph, temporal, lineage, and transaction descriptor profiles. In the selected false-negative subset, overlapping scenario tags showed that 14 out of 15 cases were associated with sparse local graph context, and 5 of these cases also appeared structurally ordinary despite being illicit. Several missed illicit cases also showed weak outgoing temporal, upstream lineage (GAR), or neighborhood abnormality signals. These findings suggest that illicit transactions are more difficult to detect when they are locally sparse, structurally camouflaged, or weakly expressed through the proposed forensic descriptors.

The selected false-positive cases show the opposite pattern: licit transactions were incorrectly flagged when they exhibited descriptor patterns resembling suspicious behavior. In the selected false-positive subset, all 15 cases were associated with sparse local graph context. In addition, overlapping scenario tags showed anomalous upstream lineage residuals in 6 cases, abnormal neighborhood patterns in 4 cases, graph structural anomalies in 4 cases, and bursty outgoing behavior in 2 cases. This indicates that the proposed descriptors are sensitive to forensic abnormality, but such abnormality is not always equivalent to illicit activity. Therefore, the model output should be interpreted as a risk-triage signal rather than as standalone forensic proof.

Overall, the analysis suggests that future improvements should focus on richer wallet level context, longer multi-hop lineage windows, improved temporal aggregation, uncertainty-aware triage, and external compliance or address-clustering signals where available. In practical AML deployment, sparse context or near boundary transactions should be treated as lower-evidence cases and routed to additional review rather than interpreted as final automated decisions.

4.8. Hybrid Framework for Interpretable Bitcoin AML Analysis

To clarify how the proposed interpretable descriptors can support practical AML investigation, this study formulates a transaction-centered hybrid framework for Bitcoin transaction review. The entry point is a seed transaction identifier obtained from an alert, compliance queue, exchange report, or ongoing investigation. The framework does not replace expert judgment, rather, it combines model-based risk scoring, descriptor-based explanation, and address-level forensic context into an auditable review record [40].

The framework consists of four coordinated layers: ingestion and context construction, descriptor-extended machine learning, address-level forensic interpretation, and case-level risk fusion. In the ingestion layer, the seed transaction is normalized and linked to reconstructed transaction-level attributes and observable blockchain metadata. A bounded graph context is then constructed to compute structural, neighborhood, outgoing temporal, and lineage-based descriptors, allowing the transaction to be assessed within its local environment rather than as an isolated ledger record.

The machine learning branch converts the reconstructed attributes and forensic descriptors into a fixed length representation for classification. The classifier estimates whether the transaction should be prioritized as illicit or licit, while post hoc feature attribution shows which descriptors had the strongest influence on the prediction. This links the model output to interpretable forensic signals, including local graph abnormality, downstream redistribution, and inconsistency with upstream transaction lineage.

The address forensics branch adds case-level context when address information is available. It extracts input and output addresses and screens them against validated high-risk repositories, sanctions related datasets, curated abuse-reporting sources, and selected open-source intelligence. It may also generate rule-based cues such as short-lived address behavior, rapid onward forwarding, relay like patterns, fan-in or fan-out concentration, and balance-sweep behavior. These cues are not treated as proof of illicit activity, since similar patterns may also occur in benign exchange, custodial, treasury management, or wallet migration workflows.

The final layer integrates the machine learning and address forensics outputs into a case-level record containing the predicted class or risk score, influential descriptors, address screening results, local neighborhood observations, and a recommended review priority. A strong model signal supported by abnormal descriptor values and address-level evidence can justify escalation, while a high model score without corroborating context remains inconclusive. Conversely, a direct match against validated high-risk intelligence may increase priority even when the model score is moderate.

In practical use, the descriptors guide investigation by indicating which aspect of the transaction requires attention. Neighborhood and discrepancy descriptors point to unusual local graph context, outgoing temporal descriptors indicate rapid redistribution, delayed forwarding, or burst-like downstream movement, and lineage-based descriptors motivate provenance checks when the transaction deviates from upstream behavior.

For example, when a transaction receives a high illicit-risk score because of strong neighborhood deviation and abnormal downstream temporal redistribution, the analyst first inspects the most influential descriptors, then traces the downstream outputs to identify rapid forwarding or splitting behavior and finally checks whether the involved addresses overlap with validated high-risk repositories or sanctions related intelligence. If the model signal, descriptor evidence,

and address-level context are mutually consistent, the alert is escalated for enhanced review, otherwise, it remains documented as a lower-priority or inconclusive case. Fig. 3 summarizes the proposed framework.

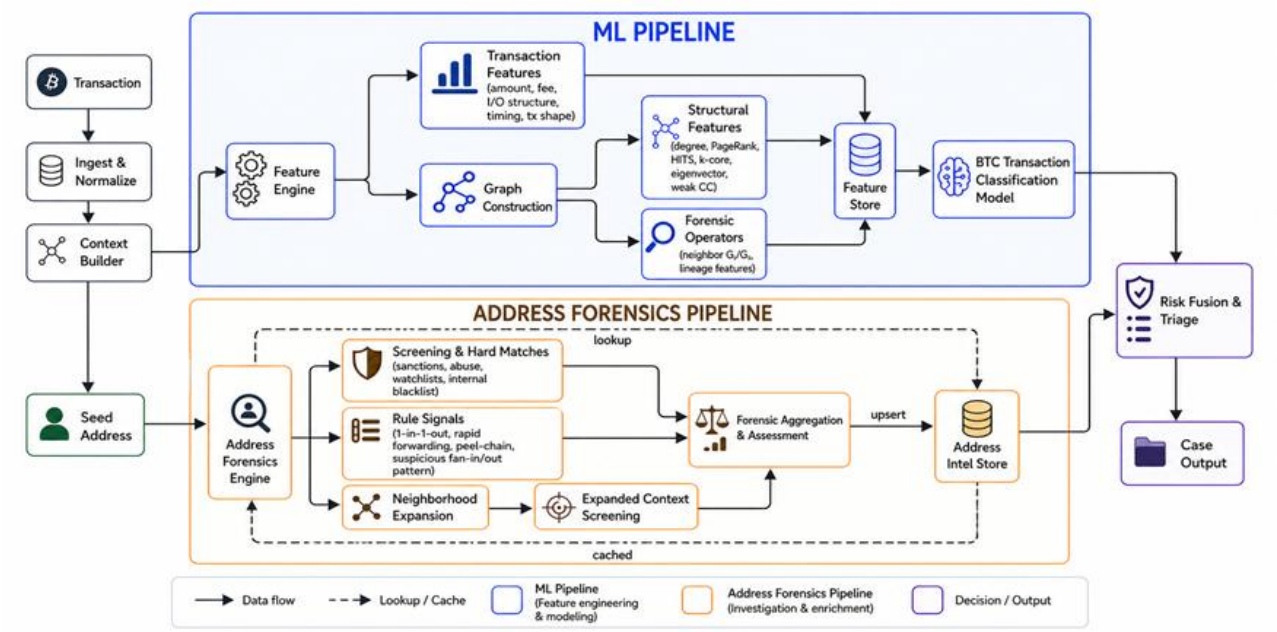

Fig. 3. Overall architecture of the proposed hybrid framework, showing the machine learning pipeline and address forensics pipeline.

5. Discussion

The results show that a descriptor-extended semantically explicit representation improves illicit transaction detection under pronounced class imbalance. Across all eight evaluated classifiers, appending the added forensic descriptors to the reconstructed base feature set increased illicit class PR-AUC, precision, recall, and F1, indicating that these descriptors capture discriminatory structure not fully represented by transaction-level on-chain attributes and derived metadata alone. The contribution of the proposed framework is therefore twofold: it replaces the original anonymized Elliptic variables with a non-anonymized and semantically explicit base representation, and it demonstrates that explicit neighborhood, temporal, and lineage descriptors add measurable predictive value on top of that reconstructed base.

The gains were most pronounced for boosting-based models, with CatBoost emerging as the strongest overall classifier under the whole-snapshot train-test protocol. This pattern suggests that the descriptor-extended representation contains heterogeneous nonlinear patterns whose predictive value is realized most effectively by models capable of learning interactions across transaction-level, neighborhood-level, temporal, and lineage-derived features. LightGBM remained a strong alternative, but with a somewhat more conservative operating profile: compared with CatBoost, it retained comparatively stronger precision while trailing in PR-AUC, recall, and F1. The comparison therefore points to different operational trade-offs rather than a purely ordinal ranking.

The explainability analysis supports this interpretation. SHAP and CatBoost's PredictionValuesChange importances produced closely aligned feature rankings, indicating a stable relevance structure for the best-performing model. More importantly, descriptor-derived feature groups accounted for about 46% of the total explanatory mass under both measures, showing that the predictive gains are not driven solely by reconstructed transaction-level attributes. Within this group, graph autoregressive lineage features and neighborhood descriptors contributed most, whereas outgoing temporal dispersion provided a smaller but complementary contribution. The predictive benefit of the descriptor-extended representation therefore appears to stem not from a simple increase in feature volume, but from the addition of behaviorally meaningful graph forensic context.

The study also extends label coverage by resolving Elliptic node identifiers to public transaction hashes and assigning labels to previously unknown transactions only when explicit licit/illicit designations are available from an external blockchain analytics source, while preserving the original Elliptic annotations and leaving ambiguous cases unlabeled. This design shifts the benchmark from an opaque feature-based classification task toward a representation setting in which model inputs can be examined, audited, and related to recognizable on-chain behavior.

The proposed descriptors further operationalize investigative concepts that are not explicitly encoded in the original Elliptic feature space, including local transaction abnormality, outgoing value redistribution behavior, temporal dispersion, and deviations from upstream transaction history. Their purpose is to make the model's behavioral signals more meaningful for AML analysis and post hoc interpretation. In addition, global and local feature attribution analysis

helps explain how reconstructed features and forensic descriptors contribute to model decisions. The proposed hybrid analytical framework further connects supervised transaction classification with address-level forensic interpretation, positioning the model output as an entry point for structured AML investigation rather than as an isolated classification result.

Several limitations should be acknowledged. First, the study remains bounded by its Elliptic-based Bitcoin AML setting and by the availability of a mapping between benchmark node identifiers and public Bitcoin transaction hashes. Therefore, the findings should be interpreted as evidence of effectiveness within an expanded Elliptic-based Bitcoin setting rather than as proof of generalizability across other blockchains, labeling sources, or institutional AML datasets. This is important because reliable illicit/licit labels in blockchain forensics remain scarce and often depend on proprietary or law-enforcement sources. Second, the proposed descriptors are more computationally demanding than simple transaction-level variables because they capture graph-forensic behavior, including neighborhood deviation, temporal dispersion, and upstream lineage dynamics. As a result, the present evidence supports the approach mainly as an offline or batch-oriented analytical method, while production deployment would require optimized incremental graph updates, caching, or approximate descriptor computation. Third, the study is limited in terms of operational and investigative validation. Although the descriptors and attribution results improve post hoc interpretability, they do not replace full forensic investigation, entity attribution, legal assessment, or human analyst judgment. Since the evaluation is benchmark-based, further operational testing is needed in real AML workflows involving alert triage, analyst feedback, and adversarial adaptation.

Future research could develop in three main directions. First, evaluating the approach on more recent Bitcoin transaction samples and alternative labeled AML datasets would help test temporal robustness and external validity. Second, extending the descriptor set beyond transaction-level features to address, cluster, and entity-level representations would allow models to capture repeated laundering patterns and broader behavioral structures. Third, testing the approach in batch and streaming monitoring architectures, including human-in-the-loop settings with AML analysts or blockchain investigators, would help assess feature extraction cost, inference latency, operational feasibility, and the practical value of its interpretability gains.

6. Conclusion

This study examined whether semantically explicit and forensically interpretable graph representations can improve illicit transaction retrieval in Bitcoin AML classification while supporting post hoc investigative interpretation. The results suggest that such representations can improve illicit transaction retrieval, although this conclusion remains bounded by the empirical setting of the Elliptic benchmark. By using Elliptic dataset as the transaction graph, temporal partitioning scheme, and source of original labels, but replacing its anonymized feature space with a reconstructed transaction-level representation derived from public on-chain data and metadata, the study showed that illicit transaction detection can benefit from more transparent and semantically meaningful graph representations.

The results show that the descriptor-extended representation yields consistent improvements in illicit class PR-AUC, precision, recall, and F1 across eight machine learning models, with CatBoost achieving the strongest overall performance. Feature attribution analysis of the best-performing model further shows that predictive behavior depends not only on reconstructed transaction-level variables, but also on a stable multiscale relevance structure in which neighborhood and lineage-derived descriptors play a substantial role.

Overall, by replacing anonymized features with reconstructed on-chain features and augmenting them with interpretable graph-forensic descriptors, the study provides a route toward AML models that are empirically effective, auditable, and better aligned with the explanatory requirements of financial crime investigation.

All the Declarations and Statements

Author Contributions Statement

Medet Shaizat – Conceptualization, Methodology, Formal Analysis, Statistical Analysis, Writing – Original Draft, Writing – Review and Editing, Data Curation, Visualization, Software Implementation, Model Training, Validation, Performance Evaluation, Supervision.
Shynar Mussiraliyeva, Ihor Tereikovskyi – Conceptualization, Methodology, Supervision, Writing – Review and Editing, Project Management.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP25793719).

Data Availability Statement

This study analyzed two datasets. The Elliptic dataset is publicly available at: https://www.kaggle.com/datasets/ellipticco/elliptic-data-sets, accessed on 15 November 2025. The second dataset is available from the corresponding author upon reasonable request.

Ethical Declarations

This article does not contain any studies with human participants or animals performed by the author. Therefore, ethical approval was not required for this research.

Acknowledgments

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

The author confirms that no generative AI or AI-assisted technologies were used in the preparation of this manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

AML - Anti-money laundering
BTC - Bitcoin
FO - Forensic operators
GAR - Graph autoregressive lineage
ML - Machine learning
PR-AUC - Area under the precision-recall curve
SHAP - SHapley Additive exPlanations
UTXO - Unspent transaction output
XAI - Explainable artificial intelligence

Appendix

Appendix A

Table 7. Full feature inventory and formal definitions.

Feature	Category	Description
total_BTC	Transaction-level attributes	Total BTC value carried by the transaction.
num_input_addresses	Transaction-level attributes	Number of input addresses referenced by the transaction.
num_output_addresses	Transaction-level attributes	Number of output addresses created by the transaction.
size	Transaction-level attributes	Serialized transaction size in bytes.
in_BTC_max	Transaction-level attributes	Maximum BTC amount among the transaction inputs.
in_BTC_median	Transaction-level attributes	Median BTC amount across the transaction inputs.
in_BTC_total	Transaction-level attributes	Total BTC amount across the transaction inputs.
out_BTC_max	Transaction-level attributes	Maximum BTC amount among the transaction outputs.
out_BTC_median	Transaction-level attributes	Median BTC amount across the transaction outputs.
out_BTC_total	Transaction-level attributes	Total BTC amount across the transaction outputs.
consolidation	Transaction-level attributes	UTXO consolidation indicator capturing whether many inputs are merged into fewer outputs.
max_share_in	Transaction-level attributes	Largest share of total input value contributed by a single input.
max_share_out	Transaction-level attributes	Largest share of total output value assigned to a single output.
fee_rate	Transaction-level attributes	Transaction fee rate per unit of transaction size.
fee_fraction	Transaction-level attributes	Transaction fee expressed as a fraction of transaction value/volume.
median_age_blocks_in	Transaction-level attributes	Median age in blocks of the input UTXOs consumed by the transaction.
median_age_blocks_out	Transaction-level attributes	Median block-age statistic associated with the transaction outputs.
in_degree	Structural topology features	Number of incoming edges to the node in G_t
out_degree	Structural topology features	Number of outgoing edges from the node in G_t.
pagerank	Structural topology features	PageRank centrality score of the node in G_t.
authority	Structural topology features	HITS authority score of the node in G_t.
hub	Structural topology features	HITS hub score of the node in G_t.
avg_neighbor_degree	Structural topology features	Average degree of neighboring nodes in G_t.

weak_cc_size	Structural topology features	Size of the weakly connected component containing the node.
k_core	Structural topology features	k-core number of the node in G_t.
eig	Structural topology features	Eigenvector centrality of the node in G_t.
rz_in_log_total_BTC_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for log(total_BTC).
rz_in_num_input_addresses_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for number of input addresses.
rz_in_num_output_addresses_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for number of output addresses.
rz_in_size_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for transaction size.
rz_in_fee_rate_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for fee rate.
rz_in_max_share_in_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for maximum input share.
rz_in_max_share_out_rz	Neighborhood forensic features	Incoming neighborhood robust z-score for maximum output share.
rz_out_log_total_BTC_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for log(total_BTC).
rz_out_num_input_addresses_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for number of input addresses.
rz_out_num_output_addresses_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for number of output addresses.
rz_out_size_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for transaction size.
rz_out_fee_rate_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for fee rate.
rz_out_max_share_in_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for maximum input share.
rz_out_max_share_out_rz	Neighborhood forensic features	Outgoing neighborhood robust z-score for maximum output share.
e_in_log_total_BTC_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for log(total_BTC).
e_in_num_input_addresses_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for number of input addresses.
e_in_num_output_addresses_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for number of output addresses.
e_in_size_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for transaction size.
e_in_fee_rate_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for fee rate.
e_in_max_share_in_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for maximum input share.
e_in_max_share_out_msd	Neighborhood forensic features	Incoming neighborhood energy (MSD-based) for maximum output share.
e_out_log_total_BTC_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for log(total_BTC).
e_out_num_input_addresses_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for number of input addresses.
e_out_num_output_addresses_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for number of output addresses.
e_out_size_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for transaction size.
e_out_fee_rate_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for fee rate.
e_out_max_share_in_msd	Neighborhood forensic features	Outgoing neighborhood energy (MSD-based) for maximum input share.
e_out_max_share_out_msd	Neighborhood forensic features (robust z / energy)	Outgoing neighborhood energy (MSD-based) for maximum output share.
out_k_bh	Outgoing temporal dispersion features	Outgoing block height count statistic k used in temporal dispersion modeling.
out_log1p_tau_rho	Outgoing temporal dispersion features	log1p transformed outgoing τρ statistic based on block-height lags.
out_lag_median	Outgoing temporal dispersion features	Median outgoing lag measured in block heights.
out_lag_std	Outgoing temporal dispersion features	Standard deviation of outgoing lags measured in block heights.
out_lag_min	Outgoing temporal dispersion features	Minimum outgoing lag measured in block heights.
gar_in1_log_total_BTC	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for log(total_BTC).
gar_in2_log_total_BTC	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for log(total_BTC).
gar_pred_log_total_BTC	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of log(total_BTC).
gar_resid_log_total_BTC	Graph autoregressive lineage (GAR) features and residuals	GAR residual for log(total_BTC) (observed minus GAR prediction).
gar_in1_num_input_addresses	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for number of input addresses.
gar_in2_num_input_addresses	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for number of input addresses.
gar_pred_num_input_addresses	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of number of input addresses.
gar_resid_num_input_addresses	Graph autoregressive lineage (GAR) features and residuals	GAR residual for number of input addresses (observed minus GAR prediction).
gar_in1_num_output_addresses	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for number of output addresses.
gar_in2_num_output_addresses	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for number of output addresses.
gar_pred_num_output_addresses	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of number of output addresses.
gar_resid_num_output_addresses	Graph autoregressive lineage (GAR)	GAR residual for number of output addresses (observed minus

	features and residuals	GAR prediction).
gar_in1_size	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for transaction size.
gar_in2_size	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for transaction size.
gar_pred_size	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of transaction size.
gar_resid_size	Graph autoregressive lineage (GAR) features and residuals	GAR residual for transaction size (observed minus GAR prediction).
gar_in1_fee_rate	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for fee rate.
gar_in2_fee_rate	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for fee rate.
gar_pred_fee_rate	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of fee rate.
gar_resid_fee_rate	Graph autoregressive lineage (GAR) features and residuals	GAR residual for fee rate (observed minus GAR prediction).
gar_in1_max_share_in	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for maximum input share.
gar_in2_max_share_in	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for maximum input share.
gar_pred_max_share_in	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of maximum input share.
gar_resid_max_share_in	Graph autoregressive lineage (GAR) features and residuals	GAR residual for maximum input share (observed minus GAR prediction).
gar_in1_max_share_out	Graph autoregressive lineage (GAR) features and residuals	First order GAR lineage term for maximum output share.
gar_in2_max_share_out	Graph autoregressive lineage (GAR) features and residuals	Second order GAR lineage term for maximum output share.
gar_pred_max_share_out	Graph autoregressive lineage (GAR) features and residuals	GAR predicted value of maximum output share.
gar_resid_max_share_out	Graph autoregressive lineage (GAR) features and residuals	GAR residual for maximum output share (observed minus GAR prediction).

Appendix B

Table 8. Selected misclassified transactions used in the error analysis.

ID	Error type	Selection type	Failure scenario tags
1	FN	High-confidence	high-confidence wrong prediction
2	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
3	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
4	FN	High-confidence	high-confidence wrong prediction, sparse local graph context, structurally ordinary illicit transaction
5	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
6	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
7	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
8	FN	High-confidence	high-confidence wrong prediction, sparse local graph context
9	FN	Near-boundary	near decision boundary, sparse local graph context
10	FN	Near-boundary	near decision boundary, sparse local graph context structurally ordinary illicit transaction
11	FN	Near-boundary	near decision boundary, sparse local graph context
12	FN	Near-boundary	near decision boundary, sparse local graph context, weak neighborhood abnormality, weak upstream lineage signal, structurally ordinary illicit transaction
13	FN	Near-boundary	near decision boundary, sparse local graph context, weak outgoing temporal features
14	FN	Near-boundary	near decision boundary, sparse local graph context, weak outgoing temporal features
15	FN	Near-boundary	near decision boundary, sparse local graph context, weak outgoing temporal features
16	FP	High-confidence	high-confidence wrong prediction, sparse local graph context
17	FP	High-confidence	high-confidence wrong prediction, sparse local graph context
18	FP	High-confidence	high-confidence wrong prediction, sparse local graph context, abnormal neighborhood pattern
19	FP	High-confidence	high-confidence wrong prediction, sparse local graph context
20	FP	High-confidence	high-confidence wrong prediction, sparse local graph context, graph structural anomaly
21	FP	High-confidence	high-confidence wrong prediction, sparse local graph context, abnormal neighborhood pattern
22	FP	High-confidence	high-confidence wrong prediction, sparse local graph context
23	FP	High-confidence	high-confidence wrong prediction, sparse local graph context, abnormal neighborhood pattern, anomalous upstream-lineage residuals, graph structural anomaly
24	FP	Near-boundary	near decision boundary, sparse local graph context
25	FP	Near-boundary	near decision boundary, sparse local graph context, abnormal neighborhood pattern, graph structural anomaly
26	FP	Near-boundary	near decision boundary, sparse local graph context, anomalous upstream lineage residuals
27	FP	Near-boundary	near decision boundary, sparse local graph context, anomalous upstream lineage residuals
28	FP	Near-boundary	near decision boundary, sparse local graph context, bursty outgoing pattern
29	FP	Near-boundary	near decision boundary, sparse local graph context, anomalous upstream lineage residuals
30	FP	Near-boundary	near decision boundary, sparse local graph context, bursty outgoing pattern, anomalous upstream-lineage

			residuals

References

[1] Chainalysis Team, “Crypto Crime Reaches Record High in 2025 as Nation-State Sanctions Evasion Moves On-Chain at Scale,” Chainalysis, Jan. 8, 2026. Available at: https://www.chainalysis.com/blog/2026-crypto-crime-report-introduction/ (accessed Mar. 05, 2026).

[2] Financial Action Task Force (FATF), *Virtual Assets: Targeted Update on Implementation of the FATF Standards on VAs and VASPs*. Paris: FATF/OECD, 2025. Available at: https://www.fatf-gafi.org/content/dam/fatf-gafi/recommendations/2025-Targeted-Upate-VA-VASPs.pdf (accessed Mar. 05, 2026).

[3] Chainalysis Team, “Chinese Language Money Laundering Networks Emerge as Major Facilitators of the Illicit Crypto Economy, Now Driving 20% of Laundering Activity,” Chainalysis, Jan. 27, 2026. Available at: https://www.chainalysis.com/blog/2026-crypto-money-laundering/ (accessed Mar. 05, 2026).

[4] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, “Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics,” arXiv:1908.02591, 2019, doi:10.48550/arXiv.1908.02591.

[5] Blockchair, “Blockchair.com API v.2.0.80 Documentation.” Available at: https://blockchair.com/api/docs (accessed Mar. 01, 2026).

[6] Crystal Intelligence, “Accurate data for blockchain investigation and compliance.” Available at: https://crystalintelligence.com/blockchain-analytics-tool/ (accessed Mar. 01, 2026).

[7] A. B. Turner, S. McCombie, and A. J. Uhlmann, “Analysis Techniques for Illicit Bitcoin Transactions,” *Frontiers in Computer Science*, vol. 2, Art. no. 600596, pp. 1–12, 2020, doi:10.3389/fcomp.2020.600596.

[8] Y. Elmougy and L. Liu, “Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics,” in *Proc. 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’23)*, Long Beach, CA, USA, 2023, pp. 3979–3990, doi:10.1145/3580305.3599803.

[9] J. Song and Y. Gu, “HBTBD: A Heterogeneous Bitcoin Transaction Behavior Dataset for Anti-Money Laundering,” *Applied Sciences*, vol. 13, no. 15, Art. no. 8766, 2023, doi:10.3390/app13158766.

[10] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, “Machine Learning Methods to Detect Money Laundering in the Bitcoin Blockchain in the Presence of Label Scarcity,” in *Proc. 1st ACM International Conference on AI in Finance (ICAIF ’20)*, New York, NY, USA: ACM, 2020, Art. no. 23, pp. 1–8, doi:10.1145/3383455.3422549.

[11] I. Alarab, S. Prakoonwit, and M. I. Nacer, “Comparative Analysis Using Supervised Learning Methods for Anti-Money Laundering in Bitcoin,” in *Proc. 2020 5th International Conference on Machine Learning Technologies (ICMLT ’20)*, Beijing, China, 2020, pp. 11–17, doi:10.1145/3409073.3409078.

[12] I. Alarab and S. Prakoonwit, “Graph-Based LSTM for Anti-Money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data,” *Neural Processing Letters*, vol. 55, no. 1, pp. 689–707, 2023, doi:10.1007/s11063-022-10904-8.

[13] W. W. Lo, G. K. Kulatilleke, M. Sarhan, S. Layeghy, and M. Portmann, “Inspection-L: Self-Supervised GNN Node Embeddings for Money Laundering Detection in Bitcoin,” *Applied Intelligence*, vol. 53, no. 16, pp. 19406–19417, 2023, doi:10.1007/s10489-023-04504-9.

[14] A. Mohan, Karthika P. V., P. Sankar, K. Maya Manohar, and A. Peter, “Improving Anti-Money Laundering in Bitcoin Using Evolving Graph Convolutions and Deep Neural Decision Forest,” *Data Technologies and Applications*, vol. 57, no. 3, pp. 313–329, 2023, doi:10.1108/DTA-06-2021-0167.

[15] J. Nicholls, A. Kuppa, and N.-A. Le-Khac, “FraudLens: Graph Structural Learning for Bitcoin Illicit Activity Identification,” in *Proc. 39th Annual Computer Security Applications Conference (ACSAC ’23)*, Austin, TX, USA, 2023, pp. 324–336, doi:10.1145/3627106.3627200.

[16] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *Proc. International Conference on Learning Representations (ICLR)*, 2017, arXiv:1609.02907, doi:10.48550/arXiv.1609.02907.

[17] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 1024–1034, doi:10.48550/arXiv.1706.02216.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’ Explaining the Predictions of Any Classifier,” in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16)*, San Francisco, CA, USA, 2016, pp. 1135–1144, doi:10.1145/2939672.2939778.

[19] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774, doi:10.5555/3295222.3295230.

[20] D. V. Kute, B. Pradhan, N. Shukla, and A. Alamri, “Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering - A Critical Review,” *IEEE Access*, vol. 9, pp. 82300–82317, 2021, doi:10.1109/ACCESS.2021.3086230.

[21] J. Černevičienė and A. Kabašinskas, “Explainable Artificial Intelligence (XAI) in Finance: A Systematic Literature Review,” *Artificial Intelligence Review*, vol. 57, Art. no. 216, pp. 1–45, 2024, doi:10.1007/s10462-024-10854-8.

[22] M. Shaizat and S. Mussiraliyeva, “Enhanced Identification of Illicit Bitcoin Transactions Through Genetic Algorithm-Based Feature Selection,” *Eastern-European Journal of Enterprise Technologies*, vol. 4, no. 9(136), pp. 34–42, 2025, doi:10.15587/1729-4061.2025.335630.

[23] A. Benzik, “Deanonymized 99.5 pct of Elliptic Transactions,” *Kaggle Datasets*, 2019. Available at: https://www.kaggle.com/datasets/alexbenzik/deanonymized-995-pct-of-elliptic-transactions (accessed Mar. 01, 2026).

[24] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. Available at: https://bitcoin.org/bitcoin.pdf (accessed Mar. 01, 2026).

[25] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names," in *Proc. 2013 Internet Measurement Conference*, 2013, pp. 127–140, doi:10.1145/2504730.2504747.
[26] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab Technical Report, no. 1999-66, 1999. Available at: https://ilpubs.stanford.edu/422/1/1999-66.pdf (accessed Mar. 10, 2026).
[27] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999, doi:10.1145/324133.324140.
[28] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003, doi:10.1137/S003614450342480.
[29] S. B. Seidman, "Network Structure and Minimum Degree," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983, doi:10.1016/0378-8733(83)90028-X.
[30] [30] P. Bonacich, "Factoring and Weighting Approaches to Status Scores and Clique Identification," *The Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972, doi:10.1080/0022250X.1972.9989806.
[31] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013, doi:10.1016/j.jesp.2013.03.013.
[32] P. J. Rousseeuw and C. Croux, "Alternatives to the Median Absolute Deviation," *Journal of the American Statistical Association*, vol. 88, no. 424, pp. 1273–1283, 1993, doi:10.1080/01621459.1993.10476408.
[33] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013, doi:10.1109/MSP.2012.2235192.
[34] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013, doi:10.1109/TSP.2013.2238935.
[35] P. Holme and J. Saramäki, "Temporal Networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012, doi:10.1016/j.physrep.2012.03.001.
[36] A.-L. Barabási, "The Origin of Bursts and Heavy Tails in Human Dynamics," *Nature*, vol. 435, pp. 207–211, 2005, doi:10.1038/nature03459.
[37] K.-I. Goh and A.-L. Barabási, "Burstiness and Memory in Complex Systems," *Europhysics Letters*, vol. 81, no. 4, Art. no. 48002, 2008, doi:10.1209/0295-5075/81/48002.
[38] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, doi:10.1080/00401706.1970.10488634.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004, doi:10.1017/CBO9780511804441.
[40] Financial Action Task Force (FATF), *Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers*. Paris: FATF/OECD, 2021. Available at: https://www.fatf-gafi.org/content/dam/fatf-gafi/guidance/Updated-Guidance-VA-VASP.pdf.coredownload.inline.pdf (accessed Mar. 15, 2026).

Authors' Profiles

Medet Shaizat: Senior Lecturer of the Department of Cybersecurity and Cryptology at Al-Farabi Kazakh National University, Kazakhstan. Areas of scientific interests: machine learning, blockchain technologies, big data analytics, digital forensics, and cybersecurity.

Shynar Mussiraliyeva: Candidate of physical and mathematical sciences, Full Professor, she is currently a Professor and Head of Cybersecurity and Cryptology Research Institute, al-Farabi Kazakh National University, Almaty, Kazakhstan. Her research interests include information security and cryptography.

Ihor Tereikovskyi: Doctor of Sciences (Engineering), Full Professor, a Professor of the Faculty of Applied Mathematics, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine. Areas of scientific interests: cybersecurity, neural networks.

How to cite this paper:

Medet Shaizat, Shynar Mussiraliyeva, Ihor Tereikovskyi, "Forensically Interpretable Graph Descriptors for Improved Illicit Bitcoin Transaction Detection", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.3, pp. 63-84, 2026. DOI:10.5815/ijcnis.2026.03.04