Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Detecting Illicit Bitcoin Transactions for Anti-Money-Laundering Monitoring: A Temporal and Interpretable Machine Learning Benchmark on the Elliptic Dataset

Abdikadir Hussein Elmi

xayeeysi77@gmail.com

Gole Research and Consultancy

Abdijalil Abdullahi

SIMAD University

Abdirisak Ahmed Salah

Gole Research and Consultancy

Abdimalik Osman Hassan

Gole Research and Consultancy

Abdirahman Mohamud Abdi

Gole Research and Consultancy

Saed Mohamud Muse

Gole Research and Consultancy

Research Article

Keywords:

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11735399/v1

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

Additional Declarations: No competing interests reported.

Detecting Illicit Bitcoin Transactions for Anti-Money-Laundering Monitoring: A Temporal and Interpretable Machine Learning Benchmark on the Elliptic Dataset

Abdikadir Hussein Elmi[1*], Abdijalil Abdullahi[2], Abdirisak Ahmed Salah[1], Abdimalik Osman Hassan[1], Abdirahman Mohamud Abdi[1], Saed Mohamud Muse[1]

[1*]Gole Research and Consultancy, Mogadishu, Somalia.
[2]Department of Information Technology, Faculty of Computing, SIMAD University, Mogadishu, Somalia.

*Corresponding author(s). E-mail(s): xayeeysi77@gmail.com;

Abstract

Cryptocurrency networks give criminals a low-friction way to move illicit funds, so institutions and regulators need tools that flag suspicious transactions and can justify each alert. This study presents a temporally faithful benchmark of three supervised classifiers (Logistic Regression, Random Forest, and XGBoost) for detecting illicit transactions in the Elliptic Bitcoin dataset for anti-money-laundering (AML) research. Models are trained on earlier time steps and tested on later ones to avoid temporal leakage, and the benchmark adds a quantification of concept drift and paired significance tests. Only 9.76% of the labeled transactions are illicit. Logistic Regression reaches an area under the receiver operating characteristic curve of 0.8777 but a precision of only 0.1762 on the illicit class, so over 80% of its alerts would be false. Random Forest and XGBoost reach illicit-class F1-scores of 0.7562 and 0.7878, with precision of 0.9875 and 0.9442 and precision-recall areas of 0.7595 and 0.7751; XGBoost attains the highest area under the receiver operating characteristic curve (0.9355) at a recall of 0.6759. Performance degrades sharply when the test window lies far from the training window, and a threshold analysis shows that missed illicit transactions are limited by feature separability rather than by the decision threshold. SHapley Additive exPlanations (SHAP) identify the small set of transaction features that drives most predictions, supporting the audit trail that compliance teams require. The study does not

propose a new algorithm; its contribution is a rigorous and interpretable reference benchmark for AML monitoring in digital finance.

Keywords: Anti-money laundering, Financial crime, Cryptocurrency, Illicit transaction detection, Machine learning, Explainable AI

JEL Classification: C53, C55, G28, K42

1 Introduction

The accelerated growth of blockchain technology has revolutionized digital financial systems by allowing peer-to-peer value transfer without the necessity of centralized third-party intermediaries (Nakamoto, 2008). Bitcoin, being the first and most popular cryptocurrency, has enabled cross-border payments, investments, and smart contracts (Buterin, 2014). Although blockchain technology's transparent and distributed ledger architecture may improve transaction traceability, it has also been used for malicious purposes, such as money laundering, terrorist financing, and fraud (Foley et al., 2019). These malicious activities, taking advantage of the transparent and distributed nature of public blockchain systems, have used the pseudonymous nature of public blockchains to conceal the origin and destination of the transferred value (Möser et al., 2018), which poses significant difficulties to regulatory compliance and financial crime prevention. Because a public blockchain has no central intermediary that screens customers or transactions, compliance monitoring has to rely on analysing the transactions that are broadcast and recorded on the ledger, which makes data-driven detection of illicit transactions a central tool for financial-crime compliance in digital finance. Traditional anti-money laundering (AML) systems, which rely on rule-based systems and manual analysis, are inefficient in identifying complex and adaptive illicit patterns in large-scale blockchain transaction networks (Arner et al., 2016).

Traditional AML methods, such as threshold-based flags for large transactions or heuristic rules for known malicious behaviors, tend to have high false positive rates and are ineffective in identifying complex patterns of transactions that span multiple hops in the network (Weber et al., 2019; Alarab et al., 2020b). The distinctive nature of blockchain data, including graph structure, temporal dependencies, and high dimensionality, demands the need for sophisticated analytical tools that can learn from transactional behavior and structural patterns simultaneously (Wu et al., 2021; Zhou et al., 2020). In this regard, machine learning has been identified as a promising tool for improving AML systems, providing the capability to model complex nonlinear relationships and interactions that are hard to manually encode. However, the incorporation of explainable artificial intelligence (XAI) in blockchain AML applications has been underrepresented in the existing literature (Lundberg and Lee, 2017; Arrieta et al., 2020), and there is a lack of studies that provide interpretable insights into the decision-making process of the models, which can be beneficial for operational transparency and regulatory accountability.

Recently, there have been studies examining different machine learning techniques to identify illegal activities in cryptocurrencies. Supervised learning techniques like Support Vector Machines, Random Forest, and Gradient Boosting Decision Trees have been used to distinguish between legal and illegal transactions (Alarab et al., 2020b) based on engineered features extracted from transaction attributes and network properties. More recently, graph neural networks (GNNs) have been proposed to make use of the inherent structural properties of transaction graphs (Weber et al., 2019; Lo et al., 2023). These studies have shown improved detection accuracy over simple statistical models. However, most of these studies either ignore model interpretability or report results on artificially balanced datasets, which do not represent real-world class imbalance settings. Moreover, existing studies have not provided rigorous temporal validation, where the model is trained and tested on data that are separated in time (Carcillo et al., 2021), which is an important requirement to represent real-world settings where past data is used to predict future illegal activities.

To address these gaps, this study does not propose a new learning algorithm; rather, it contributes a rigorous, temporally faithful benchmarking study of three widely used supervised classifiers on the Elliptic dataset, combined with an explicit quantification of concept drift and a substantive interpretability analysis, three elements that are rarely reported jointly in the reviewed literature (Table 1). We design and test three supervised machine learning classifiers: Logistic Regression, Random Forest, and Extreme Gradient Boosting (XGBoost) on a temporally ordered train-test split that maintains the original temporal order of the transactions. This approach ensures that the machine learning models are tested in a way that simulates real-world AML scenario settings (Weber et al., 2019). Beyond the temporal split, we explicitly quantify how model performance degrades as the test period is pushed further from the training period (Section 3.7), providing a direct, dataset-grounded measurement of concept drift rather than only asserting that drift is a limitation. In contrast to previous work that primarily concentrates on the classification accuracy of illicit Bitcoin transactions, our study also includes an explanation analysis of the machine learning models based on SHapley Additive exPlanations (SHAP) to interpret the machine learning model predictions and to determine the most influential features for the classification of illicit Bitcoin transactions. Explainability is an essential component of AML systems, where the underlying reasons for the automated alerts must be well understood by practitioners and regulators to establish trust in the AML system (Lundberg and Lee, 2017). Section 2.7 and Table 2 situate these contributions explicitly against the strongest published results on this dataset, and we do not claim to exceed state-of-the-art predictive accuracy; the contribution of this work is methodological rigor and interpretability rather than a new algorithm.

Our empirical findings show that ensemble learning algorithms, especially XGBoost, outperform linear models in identifying illicit transactions in a naturally imbalanced dataset, where around 9.8% of the transactions are illicit. XGBoost performs better in terms of precision-recall area under the curve (PR-AUC), F1-score for the illicit class, and overall discriminative power as measured by the area under the receiver operating characteristic curve (ROC-AUC), signifying its capability to model complex nonlinear relationships and interactions among the features of a transaction. The

addition of SHAP analysis also shows that a small set of features drives the majority of the predictions, which can be used to provide interpretable results that can help AML analysts validate the predictions of the model. This work adds to the existing body of research on AI-based blockchain analytics and signifies the need for a combination of predictive power and interpretability in AML systems.

In summary, this work contributes to the field of blockchain AML not by introducing a new algorithm, but by providing a rigorous, temporally faithful benchmarking study that quantifies the strengths, weaknesses, and drift sensitivity of established machine learning algorithms under realistic evaluation conditions, coupled with a substantive interpretability analysis. The main contributions of this work include the application of temporally validated supervised models for illicit transaction identification, an empirical comparison of ensemble and linear learning algorithms for the task that is further grounded by paired statistical significance testing (Section 4.2), and the incorporation of interpretable AI models for the development of transparent and actionable AML insights. This work is intended as a rigorously validated benchmark and interpretability reference point that can inform, rather than replace, the development of scalable AML solutions for blockchain networks.

2 Literature Review

2.1 Overview and Positioning

Research on illicit transaction detection in Bitcoin has intensified over the past five years, largely because blockchain data are publicly accessible while laundering activities are deliberately concealed and ground-truth labels are limited and often delayed. The Elliptic Bitcoin transaction dataset (Weber et al., 2019) has emerged as the de facto benchmark, providing transaction-level features, a transaction graph, and expert-assigned labels (licit/illicit/unknown), thereby enabling systematic evaluation of both supervised and graph-based approaches. Nevertheless, the dataset poses well-recognized challenges: pronounced label sparsity (roughly 77% of transactions are unlabeled), a severe class imbalance in the labeled subset (approximately 9.8% illicit), and strong temporal structure that invalidates random train–test splits. These properties mean that models achieving high overall accuracy may still perform poorly on the illicit minority class, particularly under realistic temporal evaluation conditions. This review structures recent research into five themes: (i) classical machine learning baselines and their limitations, (ii) graph-based and temporal modeling approaches, (iii) explainable AI (XAI) for financial crime detection, (iv) datasets and benchmarks extending Elliptic, and (v) critical limitations and open challenges that directly motivate the methodological choices of the present study.

2.2 Machine Learning Baselines for Illicit Transaction Detection

The foundational study by Weber et al. (2019) introduced both the Elliptic dataset and the first systematic benchmark comparing tabular classifiers (Logistic Regression, Random Forest) with a Graph Convolutional Network (GCN). Their key finding was that incorporating graph-structural information, specifically the embedding of

neighboring transaction features, improved the illicit-class F1-score from around 0.67 (Random Forest on local features alone) to approximately 0.82 with the full GCN. Critically, Weber et al. employed a temporal split (the first 34 time-steps for training, the last 15 for testing), acknowledging that randomly shuffled splits would violate the chronological ordering of real-world transaction data. This methodological decision is of central importance and is directly adopted in the present study; yet many subsequent works have overlooked it.

Alarab et al. (2020a) conducted a comparative analysis of supervised classifiers—Logistic Regression, SVM, Random Forest, and a shallow neural network—on the Elliptic dataset without graph information. Their results confirmed that Random Forest outperforms linear models on the illicit class F1 metric, but they also revealed that all tabular models degraded substantially around time-step 43, which corresponds to the closure of a dark marketplace. This event caused a distributional shift that static classifiers could not accommodate, thereby demonstrating a fundamental limitation of non-adaptive, feature-only approaches: they are brittle to emerging criminal behaviors.

More recent work has pushed classical ensemble methods further. Adloori et al. (2024, unpublished preprint) compared Logistic Regression, Random Forest, XGBoost, and multiple GNN variants (a GCN, a graph attention network (GAT), and a proposed GAT-ResNet) on Elliptic, reporting that tree-ensemble models remain competitive baselines but that graph-aware architectures achieve higher recall on the illicit class. Importantly, the study highlights that XGBoost, when combined with appropriate class-weighting for imbalance, achieves near-competitive performance relative to more complex GNN models at a fraction of the computational cost. This observation is directly relevant to the present work, which evaluates XGBoost as the primary ensemble candidate and benchmarks it against simpler baselines (Logistic Regression and Random Forest) under an identical temporal split. Unlike Adloori et al., however, the present study explicitly avoids over-sampling the training set before splitting, thereby preventing data leakage—a methodological flaw documented in prior literature (Carcillo et al., 2021).

A critical gap across these baseline studies is the consistent absence of interpretability analysis. Model outputs are reported as performance metrics only, with no examination of *which features* drive predictions. This matters operationally: AML analysts must justify flagged transactions to regulators, meaning that a high-precision but opaque model has limited practical utility. The present study addresses this gap directly through SHAP-based post-hoc explanation.

2.3 Graph-Based and Temporal Modeling Approaches

The inherently graph-structured nature of Bitcoin transactions, where nodes represent transactions and directed edges encode value flows, motivated a wave of GNN-based approaches following Weber et al. (2019). Alarab et al. (2020b) augmented the GCN architecture by interleaving graph convolutional layers with linear transformation layers and an MLP output head, reporting improved illicit-class F1 compared to the Weber et al. baseline on Elliptic. Their model outperforms the original GCN but still suffers from the same dark-market shutdown event at time-step 43, confirming that spectral GCNs without temporal awareness are poorly equipped to handle concept drift.

Pareja et al. (2020) addressed this limitation with EvolveGCN, which replaces static GCN weights with dynamically updated parameters driven by an RNN, effectively treating the graph as a temporally evolving object. Applied to the Elliptic dataset, EvolveGCN improved robustness around the dark-market event, though even the dynamic model suffered performance degradation, underscoring that *emerging* events that are absent from training data remain fundamentally difficult to detect regardless of model architecture. The broader implication is that temporal robustness cannot be fully solved by architectural complexity alone; evaluation protocols and data collection practices must also account for distribution shift.

Lo et al. (2023) introduced Inspection-L, a self-supervised GNN framework combining Deep Graph Infomax (DGI) with a Graph Isomorphism Network (GIN) encoder, followed by a Random Forest classifier. By leveraging global graph information without relying solely on K-hop neighborhoods, Inspection-L mitigated the over-smoothing and label sparsity problems endemic to supervised GNNs on Elliptic. Their approach achieved state-of-the-art illicit F1 at the time of publication and is notable for its practical insight: when labels are extremely scarce, self-supervised pre-training on unlabeled transactions can transfer useful structural representations.

Alarab and Prakoonwit (2023) further combined a temporal GCN with LSTM units (Graph-LSTM) to capture sequential dependencies between consecutive time-steps in Elliptic. Their model improved performance on the temporal test window, particularly around distributional shifts, but introduced substantial model complexity and required careful hyperparameter tuning. Collectively, these graph and temporal studies establish that while GNN architectures can improve detection accuracy, they do so at the cost of interpretability, scalability, and reproducibility, making them less suitable as the primary tool in regulatory compliance contexts where auditability is required.

The present study deliberately occupies the complementary position: rather than proposing a new GNN architecture, it rigorously evaluates whether classical ensemble methods (Random Forest and XGBoost) combined with SHAP interpretability can provide competitive, auditable, and practically deployable performance under realistic conditions (temporal split, natural class imbalance, no synthetic oversampling of the test set).

2.4 Explainable AI for Financial Crime Detection

Model explainability in AML has received growing attention, driven partly by regulatory requirements and partly by the practical need to reduce false-positive alert fatigue among compliance analysts. Lundberg and Lee (2017) introduced SHAP (SHapley Additive exPlanations), a game-theoretic framework that assigns each feature a contribution value consistent with model outputs. SHAP has since been widely adopted in financial fraud and AML contexts because it satisfies desirable axiomatic properties (local accuracy, missingness, consistency) and can generate both global (feature importance ranking) and local (transaction-level) explanations.

In the context of cryptocurrency anomaly detection, Elmougy and Liu (2023) applied SHAP to the Elliptic++ dataset—an extended version of Elliptic including actor-level graphs—and demonstrated that feature attribution could be semantically linked to known laundering typologies, such as structuring and layering. This represents

an important step beyond performance reporting, as it connects model outputs to investigative workflows. However, their work focuses on the richer Elliptic++ dataset and uses Random Forest without the temporal split discipline of the present study.

Hasan et al. (2025) employed SHAP to interpret a one-dimensional CNN classifier trained on blockchain features, explicitly connecting explainability to the mitigation of class imbalance and the reduction of false positives. Their finding that a small subset of features (fewer than 10 out of 166) dominates predictive decisions is consistent with the results of the present study and has important operational implications: AML dashboards can be simplified to present analysts with only the most actionable feature signals.

Research by Kungu et al. (2026) combined VAE-based anomaly detection with SHAP interpretation in a hybrid deep learning AML framework applied to real-world cross-border transaction data. While this work goes beyond Elliptic into a live banking context, it reinforces a recurring finding: models with high predictive performance but low interpretability face adoption barriers, since compliance officers need to explain flagged cases to auditors. This operational transparency argument directly motivates the SHAP component of the present study.

A critical limitation of most XAI-in-AML studies is that they treat interpretability as an *add-on* analysis performed after model selection, rather than as an integrated criterion during model evaluation. The present study addresses this by treating SHAP feature importance as a substantive output of the research rather than an appendix, enabling a discussion of which transaction features are most diagnostic for illicit activity and how these align with known AML typologies.

2.5 Datasets and Benchmarks

The Elliptic dataset (Weber et al., 2019) remains the most widely used public benchmark for supervised Bitcoin AML research, but its limitations are well-documented. The 77% unlabeled fraction creates a significant semi-supervised challenge. The 49 temporal time-steps cover only a two-week window, limiting long-term generalizability. The 165 anonymized features—while rich—are opaque by design, making it impossible to map results directly to economic transaction attributes.

Lorenz et al. (2020) addressed the label scarcity problem through an active learning framework, demonstrating that a model trained on only 5% of available labels could match the performance of a fully supervised Random Forest baseline. This work is important because it confirms that label scarcity is not merely a nuisance but a fundamental constraint: in real deployments, expert annotation of illicit transactions is costly and slow, so methods that maximize detection performance per labeled example have high practical value. Importantly, their results also show that unsupervised anomaly detection methods fail on Elliptic because illicit transactions are *not* statistical outliers in feature space; they cluster within regions occupied by licit transactions. This negative result justifies the present study's focus on supervised and ensemble methods rather than unsupervised approaches.

Elmougy and Liu (2023) introduced the Elliptic++ dataset, which extends the original Elliptic transaction graph with actor-level entity graphs, allowing models to incorporate account-level behavioral patterns alongside transaction-level features.

Their experiments show that the additional actor graph consistently improves detection performance, suggesting that transaction-only models (including the present study) represent a lower bound on achievable performance. Future work could investigate whether SHAP applied to actor-level features yields similarly interpretable insights.

Schnoering and Vazirgiannis (2025) introduced a large-scale temporally annotated Bitcoin transaction graph with labeled subsets and GNN baselines, providing an alternative benchmark that complements Elliptic. This work underscores the broader problem of evaluation fragmentation in the field: because studies use different datasets, different splits, and different evaluation metrics, direct comparisons across papers are frequently misleading. The present study contributes to comparability by (i) using the standard Elliptic benchmark, (ii) adopting a strict temporal split, and (iii) reporting both ROC-AUC and PR-AUC, the latter being more informative under class imbalance.

2.6 Limitations and Open Challenges

Three recurring limitations across the reviewed literature directly motivate the methodological choices of the present study.

(1) Temporal validation deficits. The majority of classical ML studies on Elliptic either use random splits or do not report their splitting strategy. Random splits allow future transaction features to leak into training data, inflating reported performance metrics. Poursafaei et al. (2021) explicitly demonstrate that models evaluated with temporal splits consistently show lower illicit-class F1 than identical models evaluated with random splits, sometimes by margins exceeding 15 percentage points. This performance gap represents an important reproducibility and deployment-readiness concern. The present study addresses this by employing a strict temporal train–test split in which all training transactions precede all test transactions chronologically.

(2) Class imbalance handling. Many prior studies address class imbalance through synthetic oversampling (e.g., SMOTE) applied to the full dataset before splitting, which introduces data leakage and yields unrealistically optimistic metrics (Carcillo et al., 2021). The present study instead relies on the class-weighting mechanisms built into each classifier and evaluates performance using the Precision-Recall AUC (PR-AUC), which is more sensitive than ROC-AUC to minority-class detection quality in heavily imbalanced settings.

(3) Absence of interpretability. Despite the operational importance of explainability in AML, where analysts must justify alerts to compliance officers and regulators, the majority of studies reviewed report only aggregate performance metrics. Hasan et al. (2024) identify this as a systemic gap in the blockchain AML literature and call for SHAP or LIME-based analyses to be integrated into future benchmark evaluations. The present study directly addresses this call by incorporating a SHAP-based feature attribution analysis as a first-class research output, enabling an assessment of which transaction features are most diagnostic for illicit activity.

2.7 Positioning of the Present Study

Table 1 summarizes the most directly related studies. Relative to this body of work, the present study makes three distinct contributions. First, it applies a rigorous temporal

train–test split that faithfully replicates the conditions under which a deployed AML system must operate—using past data to predict future illicit transactions. Second, it evaluates three supervised classifiers (Logistic Regression, Random Forest, XGBoost) under natural class imbalance, without synthetic data augmentation, reporting PR-AUC, F1, and ROC-AUC to facilitate direct comparison with prior work. Third, it integrates SHAP-based post-hoc explanation as a substantive output, bridging the gap between predictive performance and operational interpretability that has been identified as a critical limitation in the existing literature. Unlike GNN-based approaches, which improve detection accuracy but reduce auditability, the present study prioritizes the combination of predictive power and model transparency that is essential for regulatory trust in AML systems.

Table 1 Summary of closely related studies on machine learning for illicit Bitcoin transaction detection.

Paper	Year	Method	Dataset	Temporal Split	XAI	Imbalance Handling
Weber et al. (2019)	2019	RF, LR, GCN, EvolveGCN	Elliptic	Yes	No	None reported
Alarab et al. (2020a)	2020	LR, SVM, RF, MLP	Elliptic	No	No	None
Alarab et al. (2020b)	2020	GCN + linear layers	Elliptic	No	No	None
Lorenz et al. (2020)	2020	Active learning + RF	Elliptic	Partial	No	Label scarcity approach
Pareja et al. (2020)	2020	EvolveGCN (dynamic GCN)	Elliptic	Yes	No	None
Lo et al. (2023)	2023	Self-supervised GNN + RF	Elliptic	No	No	None
Alarab & Prakoonwit (2023)	2023	Graph-LSTM (temporal GCN)	Elliptic	Yes	No	None
Elmougy & Liu (2023)	2023	RF + SHAP	Elliptic++	No	Yes	Feature refinement
Hasan et al. (2025)	2025	1D-CNN + SHAP	Elliptic	No	Yes	Sampling
Adloori et al. (2024, preprint)	2024	LR, RF, XGB, GCN, GAT	Elliptic	No	No	Class weighting
Present study	**2026**	**LR, RF, XGBoost + SHAP**	**Elliptic**	**Yes**	**Yes**	**Class weighting; no leakage**

To situate the present results quantitatively within the published literature, Table 2 reports illicit-class precision, recall, and F1-score for the present study alongside three directly comparable, externally published results on the same Elliptic dataset: the Random Forest and Skip-GCN results originally reported by Weber et al. (2019), and the self-supervised graph representation learning result (Inspection-L) reported by Lo et al. (2023). We restrict this comparison to studies that report precision, recall, and

F1-score on the illicit class using a methodology comparable to ours (supervised or self-supervised classification with a reported train–test protocol); several other studies cited in Table 1 either report only aggregate accuracy, use a different derivative dataset (e.g., Elliptic++), or do not disclose sufficient metric detail for a fair numeric comparison, and are therefore excluded from Table 2 rather than have their numbers approximated.

Table 2 Illicit-class precision, recall, and F1-score compared against externally published results on the Elliptic dataset.

Study	Method	Precision	Recall	F1
Weber et al. (2019)	Random Forest	0.956	0.670	0.788
Weber et al. (2019)	Skip-GCN	0.812	0.623	0.705
Lo et al. (2023)	Inspection-L (self-supervised GNN + RF)	0.972	0.721	0.828
Present study	Logistic Regression	0.176	0.791	0.288
Present study	Random Forest	0.988	0.613	0.756
Present study	**XGBoost**	**0.944**	**0.676**	**0.788**

The present study's XGBoost F1-score (0.788) is essentially on par with Weber et al.'s original Random Forest baseline (0.788) and below the self-supervised, graph-augmented Inspection-L result (0.828), while exceeding the graph-only Skip-GCN result (0.705). This positions the present contribution honestly: it does not surpass the strongest published graph-based methods on raw F1, and we do not claim otherwise. Its distinguishing value lies elsewhere, in the combination of (i) a temporally faithful evaluation protocol applied consistently across all three classifiers, (ii) an explicit quantification of performance sensitivity to the train–test split point and to hyperparameter choices (Sections 3.7–3.8), (iii) paired statistical significance testing of model differences (Section 4.2), and (iv) SHAP-based interpretability integrated as a primary output rather than an afterthought, none of which are jointly reported in the three comparison studies above. We consider this combination of rigor and interpretability, rather than a claim of state-of-the-art predictive accuracy, to be the primary contribution of this work.

3 Methodology

This study adopts a structured, end-to-end machine learning pipeline for illicit transaction detection using the Elliptic dataset. The methodology integrates data preprocessing, exploratory analysis, supervised classification, imbalance-aware evaluation, and model explainability to ensure both predictive performance and interpretability.

3.1 Methodological Framework

Figure 1 illustrates the end-to-end workflow employed in this study. The framework consists of six sequential stages, starting from data acquisition and ending with model explainability, ensuring a comprehensive and interpretable analysis pipeline.

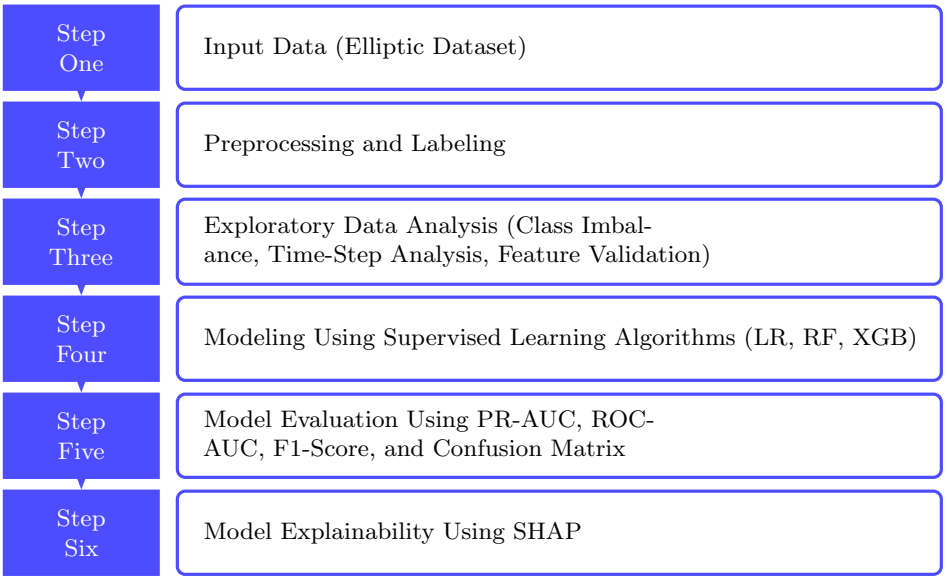


Fig. 1 End-to-end methodological framework for illicit transaction detection using the Elliptic dataset

3.2 Dataset Description

This study utilizes the publicly available Bitcoin transaction dataset released by Elliptic, which has been extensively used in prior research on blockchain analytics and anti-money-laundering (AML). The dataset is specifically designed to support supervised learning for illicit transaction detection and includes labeled transaction records along with temporal information and feature-based representations derived from the Bitcoin blockchain. Owing to its real-world provenance and substantial scale, it is particularly well suited for evaluating machine learning models under realistic AML conditions.

The dataset comprises three main components: a transaction feature file, a class label file, and a transaction edge list. The transaction feature file contains numerical attributes for each Bitcoin transaction, including a discrete time-step indicator and

a set of engineered features that capture transactional behavior and local network characteristics. Each transaction is represented as a node associated with a unique transaction identifier. These features are precomputed by the dataset provider and consist of aggregated statistics derived from the transaction graph, enabling learning algorithms to model behavioral patterns without direct access to raw blockchain scripts or wallet addresses.

All 165 features are anonymized by the dataset provider due to intellectual property restrictions (Weber et al., 2019), so no raw economic labels (e.g., transaction fee, output volume) are attached to individual feature names. Nevertheless, the dataset documentation establishes a clear structural distinction between two feature groups (Weber et al., 2019): features `feat_1` through `feat_94` encode *local transaction-level information*, comprising the transaction time step, the number of inputs and outputs, transaction fees, output volume, and aggregated statistics (e.g., average BTC received or spent by inputs/outputs, average number of incoming and outgoing transactions associated with those inputs/outputs). Features `feat_95` through `feat_165` encode *aggregated neighbourhood information*, derived by computing the maximum, minimum, standard deviation, and correlation coefficients of the same local attributes across transactions one hop backward and one hop forward from the focal transaction in the graph. This structural knowledge is used later in Section 4.5 to interpret the SHAP results, even in the absence of precise semantic feature labels.

The class label file assigns each transaction to one of three categories: licit, illicit, or unknown. Licit transactions represent legitimate economic activity, whereas illicit transactions are linked to known malicious entities, such as scams, darknet markets, or other criminal services identified through forensic investigation. Transactions labeled as unknown correspond to instances where ground-truth classification is unavailable. In the present study, only transactions with confirmed licit or illicit labels are retained to establish a binary supervised learning setting. Transactions labeled as unknown are excluded to prevent label ambiguity and to ensure reliable model evaluation.

It is important to distinguish this dataset's notion of "illicit" from the broader regulatory notion of a "suspicious transaction." In this dataset, the illicit label denotes transactions forensically linked, after the fact, to a known illicit entity or service (e.g., a darknet market, a Ponzi scheme, a scam) through external investigation, whereas a regulatory suspicious transaction report (STR), as used in AML practice, reflects a financial institution's contemporaneous, risk-based judgment that a transaction warrants further scrutiny, without requiring forensic confirmation of wrongdoing (FATF, 2021). A model trained on this dataset therefore learns to reproduce forensically confirmed illicit-activity patterns, which is a narrower and more precisely defined target than the population of transactions a compliance analyst would flag as suspicious in practice; this distinction should be kept in mind when interpreting the practical AML relevance of the reported precision and recall values.

The transaction edge list defines directed relationships between transactions, forming a temporal transaction graph that represents the flow of funds within the Bitcoin network. A directed edge from one transaction to another indicates that an output from the former is spent as an input in the latter. Although this study primarily emphasizes

feature-based machine learning models, the availability of transaction graph information enables future extensions involving graph-based or hybrid modeling approaches.

Following preprocessing and label filtering, the final dataset used in this research comprises 46,564 labeled transactions, including 42,019 (90.24%) licit and 4,545 (9.76%) illicit instances, reflecting a naturally imbalanced class distribution. This imbalance closely mirrors real-world AML contexts, where illicit activity constitutes only a small fraction of total transactions. Each transaction is associated with a discrete time step, allowing the data to be chronologically ordered. This temporal structure supports the implementation of time-based train–test splits, which more accurately simulate real deployment conditions in which models are trained on historical data and applied to future transactions.

Overall, the Elliptic dataset serves as a realistic, large-scale, and well-structured benchmark for assessing machine learning models for illicit Bitcoin transaction detection. Its combination of labeled data, temporal sequencing, and rich feature representations makes it particularly appropriate for examining both predictive performance and explainability within blockchain-based AML systems.

3.3 Dataset Distribution Analysis

Figure 2 presents the class distribution of the labeled transactions analyzed in this study, revealing a pronounced imbalance between licit and illicit categories. Licit transactions constitute approximately 90.24% of the dataset, whereas illicit transactions account for only about 9.76%. This distribution mirrors real-world blockchain transaction networks, in which illicit activity represents a relatively small share of total financial transactions.

Such class imbalance carries significant implications for machine learning–based anti-money-laundering (AML) systems. In particular, it makes overall accuracy a misleading performance metric, since a trivial classifier could attain high accuracy by consistently predicting the majority class. Accordingly, this study prioritizes evaluation metrics better suited to imbalanced classification settings, including precision, recall, F1-score, and precision–recall area under the curve (PR-AUC), with particular emphasis on performance for the illicit class. The observed distribution also supports the adoption of cost-sensitive learning approaches and ensemble methods that can effectively identify minority-class patterns while limiting excessive false positives.

Overall, the analysis of the dataset's class distribution demonstrates that the classification problem addressed in this research is both realistic and challenging, thereby underscoring the importance of advanced machine learning techniques for detecting illicit Bitcoin transactions.

3.4 Temporal Structure of Transactions

Figure 3 illustrates the distribution of transactions across discrete time steps, highlighting the temporal dynamics of Bitcoin transaction activity within the dataset. Transactions occur over a sequence of time steps with varying frequencies, demonstrating that both licit and illicit activities are distributed throughout the observation period rather than concentrated in specific intervals. This temporal variation indicates

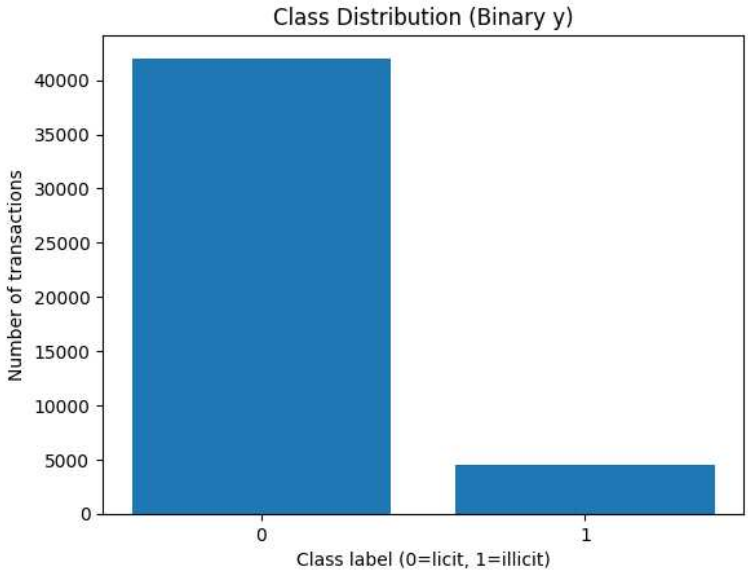


Fig. 2 Class distribution of labeled transactions in the Elliptic dataset

that transaction behavior changes over time, potentially in response to external influences such as market conditions, regulatory developments, or shifts in illicit operational strategies.

The presence of a well-defined temporal structure carries important methodological implications for model evaluation. Random train–test splits risk introducing information leakage by allowing patterns from future transactions to inform model training. To mitigate this issue, the present study employs a time-based train–test split, whereby models are trained on transactions from earlier time steps and evaluated on those from later time steps. This strategy more accurately reflects real-world AML deployment scenarios, in which historical transaction data are used to detect emerging illicit behavior.

By explicitly incorporating temporal ordering into the experimental design, this study strengthens the robustness and credibility of its findings. The temporal analysis underscores that effective illicit transaction detection systems must address not only class imbalance but also the evolving nature of transaction patterns.

An important consequence of the temporal split is that the class balance is not identical across the training and test partitions. At the cutoff used throughout Sections 4, 3.7, and 3.8 (time-step 35), the training set (time-steps 1–35, $n = 31{,}235$) contains 3,644 illicit transactions, an illicit rate of approximately 11.67%, whereas the test set (time-steps 36–49, $n = 15{,}329$) contains 901 illicit transactions, a lower illicit rate of approximately 5.88%. This drop in illicit prevalence between training and test partitions is a direct consequence of the temporal, rather than random, splitting strategy, and it has a specific implication for the PR-AUC values reported in this study: because the PR-AUC of a random (no-skill) classifier is bounded below by the prevalence of the positive class, a PR-AUC computed on the test set is not

directly comparable in absolute terms to one computed on a partition with a different illicit rate, nor to PR-AUC values reported by other studies that use different splits or prevalence levels. All PR-AUC comparisons in this paper are therefore made *within* the same test partition across models, where this effect is held constant, rather than across partitions or across studies. This shrinking positive-class prevalence at later time-steps also partially explains the sharp metric degradation observed at cutoff 43 in Section 3.7: beyond that cutoff the test set's illicit rate falls further, to approximately 2.73%, compounding the effect of the underlying distributional shift discussed below.

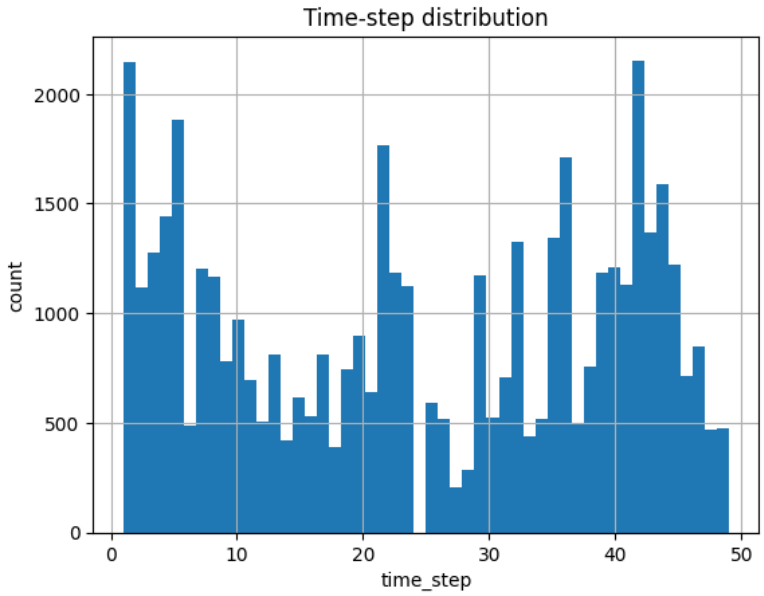


Fig. 3 Temporal structure of transactions across time steps

3.5 Data Preprocessing

Robust data preprocessing is essential to ensure the reliability and validity of machine learning–based illicit transaction detection. Prior to model training, several preprocessing procedures are applied to the Elliptic dataset to enhance data quality, clearly define the learning objective, and mitigate the risk of information leakage. These steps are designed to reflect realistic anti-money-laundering (AML) deployment settings while preserving reproducibility and interpretability.

First, transaction-level feature data are merged with class labels using a unique transaction identifier. This integration ensures that each transaction is represented by a consistent set of numerical features along with its corresponding ground-truth label. Transactions labeled as unknown are excluded to eliminate ambiguity in supervised learning, resulting in a binary classification problem comprising only licit and illicit

classes. This choice facilitates clearer performance evaluation and is consistent with prior studies focusing on confirmed illicit activity.

Second, class labels are encoded in binary form, with licit transactions assigned a value of 0 and illicit transactions assigned a value of 1. This encoding ensures compatibility with standard classification algorithms and supports the use of evaluation metrics tailored to imbalanced datasets. Following label encoding, the dataset is examined for missing values and duplicate records. Because the Elliptic dataset provides pre-engineered features, no significant missing values are detected and no imputation is required. Additionally, no duplicate transactions remain after the merging process.

Feature scaling is applied selectively depending on model requirements. For linear models such as Logistic Regression, numerical features are standardized to have zero mean and unit variance to promote stable optimization and prevent features with larger scales from disproportionately influencing the learning process. In contrast, tree-based ensemble models, including Random Forest and XGBoost, are trained using the original feature scales, as these models are inherently robust to monotonic feature transformations.

Given the naturally imbalanced class distribution—where illicit transactions form a minority—imbalance mitigation strategies are incorporated during model training rather than through data resampling. Specifically, cost-sensitive learning is implemented by assigning greater misclassification penalties to the illicit class, encouraging the models to better identify minority instances. This strategy preserves the original data distribution and avoids introducing synthetic patterns that could distort authentic transaction behavior.

Finally, to ensure realistic evaluation and prevent temporal leakage, the dataset is partitioned using a time-based splitting strategy. Transactions from earlier time steps are used for training, while those from later time steps are reserved for testing. This temporal separation mirrors operational AML contexts, in which historical transaction data are used to detect emerging illicit activities. The resulting preprocessed dataset is applied consistently across all evaluated models to ensure fair comparison and reproducibility.

3.6 Machine Learning Models

This study assesses a set of supervised machine learning models with progressively greater representational capacity for detecting illicit Bitcoin transactions. The selected models comprise a linear baseline and two ensemble-based methods, enabling a structured comparison between simpler and more expressive learning approaches under realistic anti-money-laundering (AML) conditions. All models are trained on the same preprocessed dataset and evaluated using an identical temporal split to ensure a fair and consistent comparison.

Logistic Regression is employed as a baseline classifier due to its simplicity, interpretability, and established role in financial risk modeling (Hosmer et al., 2013). As a linear model, it estimates the probability of illicit activity as a weighted combination of input features through the logistic (sigmoid) function, which maps the linear combination of predictors to a probability in the interval $(0, 1)$. Although its capacity to capture nonlinear relationships is limited, it serves as a meaningful benchmark for

evaluating the incremental benefits of more complex models (Hosmer et al., 2013). To address class imbalance, class weights are adjusted during training, assigning a higher penalty to the misclassification of illicit transactions.

Random Forest (Breiman, 2001) is adopted as a nonlinear ensemble method that constructs multiple decision trees using bootstrap sampling and random feature selection. By aggregating predictions across trees through majority voting, Random Forest reduces variance and enhances generalization performance relative to individual decision trees (Breiman, 2001). The model is well suited to tabular blockchain data, as it can learn intricate feature interactions without extensive feature engineering. Class imbalance is handled through cost-sensitive learning by applying balanced class weights during training.

Extreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016) is selected as the primary advanced model due to its strong performance on structured data and its ability to capture nonlinear and higher-order feature interactions. XGBoost sequentially builds an ensemble of decision trees, where each subsequent tree corrects the errors of preceding ones through second-order gradient-based optimization (Chen and Guestrin, 2016). To accommodate the imbalanced class distribution, the `scale_pos_weight` parameter is configured to impose a greater penalty on misclassifying illicit transactions. Regularization (L_1 and L_2) and subsampling techniques are incorporated to reduce overfitting and improve generalization. The superior predictive performance of XGBoost observed in this study supports its selection for subsequent explainability analysis.

Collectively, these models provide a comprehensive evaluation of linear, bagging-based, and boosting-based approaches to illicit transaction detection. This comparison offers insight into the trade-offs among simplicity, predictive performance, and interpretability in blockchain-based AML systems.

3.7 Performance Sensitivity Analysis of Split Ratio

The temporal cutoff used throughout this study (time-step 35, corresponding to an approximately 67%/33% train–test split by transaction count) is not an arbitrarily chosen ratio but a chronological boundary: unlike a random split, it cannot be freely resized without altering which time-steps the models are evaluated on. To assess how sensitive the reported results are to the specific choice of cutoff, and to verify that the models are not being evaluated on an unrepresentative "lucky" split, the full pipeline described above is repeated at five temporal cutoffs spanning the observation window: time-steps 25, 30, 35, 40, and 43, corresponding to train proportions of approximately 52%, 58%, 67%, 79%, and 89% of labeled transactions, respectively. Each cutoff is evaluated independently, with all preprocessing, class-weighting, and model configurations held identical to those described in Sections 3.5 and 3.6.

Table 3 summarizes the results. Two patterns are evident. First, model rankings are stable between cutoffs 25 and 35: the ensemble methods (Random Forest and XGBoost) outperform Logistic Regression on precision, F1-score, and PR-AUC at every cutoff in this range, confirming that the comparative conclusions of this study are not an artifact of the specific 67%/33% split reported in Section 4. This ranking, however, does not hold at cutoff 43: while XGBoost continues to achieve the highest F1-score (0.1437) at this cutoff, Random Forest's F1-score (0.0395) falls below that of Logistic Regression

(0.1104), reversing the ordering observed at every earlier cutoff. This reversal is driven by Random Forest's near-total loss of recall at cutoff 43 (0.0207, compared with Logistic Regression's 0.3793), rather than by any improvement in Logistic Regression's underlying discriminative ability; Logistic Regression's comparatively higher recall, which was identified in Section 4 as coming at the cost of very low precision, incidentally yields a higher F1-score than Random Forest once Random Forest's recall collapses almost entirely under the extreme drift and reduced illicit prevalence present at this cutoff. The ranking between cutoffs 25 and 35 is therefore stable, but this stability fails at cutoff 43 as a direct consequence of the distributional shift discussed below, and the two ensemble methods should not be assumed to retain a fixed relative ordering under severe drift. Second, performance is not uniform across cutoffs: F1-scores for the ensemble models peak around cutoffs 25–30 and decline as the cutoff moves later, reaching their lowest values at cutoff 43. This degradation coincides with the known closure of a dark marketplace at time-step 43, previously reported to induce a distributional shift in this dataset (Alarab et al., 2020a). The present sensitivity analysis reproduces that effect directly within the pipeline used in this study: at cutoff 43, the test set consists almost entirely of transactions occurring shortly after this shift, and every model's recall for the illicit class collapses (e.g., Random Forest recall falls from 0.61 at cutoff 35 to 0.02 at cutoff 43), since the classifiers, trained on transaction patterns from before the closure, encounter behavior for which they have seen very few comparable illicit examples. As discussed in Section 3.4, this cutoff also coincides with a further drop in test-set illicit prevalence (to approximately 2.73%), which independently lowers the achievable PR-AUC and partially compounds the apparent severity of the metric collapse. This confirms that the moderate recall reported for the illicit class in Section 4 reflects a genuinely difficult and temporally evolving detection problem, rather than a weakness specific to the particular cutoff chosen, and it reinforces the discussion of concept drift in Section 5.

3.8 Hyperparameter Tuning

The hyperparameter values reported for Random Forest and XGBoost in Section 3.6 were manually configured based on common practice for tabular, imbalanced classification tasks, rather than selected through a formal search. To verify that the reported performance is not contingent on a fortunate manual choice, and following common recommendations for reporting tuned baselines prior to deployment, a randomized hyperparameter search is conducted for both ensemble models. Because the evaluation protocol of this study is explicitly temporal, cross-validation is likewise restricted to respect chronological order: the official training period (time-steps 1–35) is internally divided into a tuning sub-training set (time-steps 1–28) and a held-out tuning validation set (time-steps 29–35), and hyperparameter configurations are scored only on this validation set, never on the official test set (time-steps 36–49). This avoids the temporal leakage that a conventional shuffled k-fold cross-validation would introduce. For each model, 8–10 candidate configurations are sampled from a search space covering the number of estimators, tree depth, learning rate (for XGBoost), subsampling ratios, and regularization strength, and the configuration achieving the highest illicit-class

Table 3 Performance of the three models across five temporal train–test cutoffs. Train% is the proportion of labeled transactions used for training at each cutoff.

Cutoff	Train%	Model	Precision	Recall	F1-Score	PR-AUC	ROC-AUC
25	52.0	LR	0.1948	0.9683	0.3243	0.2203	0.8114
		RF	0.9893	0.7563	0.8573	0.8907	0.9540
		XGBoost	0.9082	0.8197	0.8617	0.8928	0.9593
30	57.8	LR	0.1799	0.9554	0.3028	0.2448	0.8522
		RF	0.9902	0.7612	0.8607	0.8640	0.9528
		XGBoost	0.8685	0.8133	0.8400	0.8626	0.9431
35 (used in Sec. 4)	67.1	LR	0.1762	0.7913	0.2883	0.3142	0.8777
		RF	0.9875	0.6127	0.7562	0.7595	0.9288
		XGBoost	0.9442	0.6759	0.7878	0.7751	0.9355
40	78.6	LR	0.1832	0.7481	0.2943	0.4099	0.8871
		RF	0.9803	0.5687	0.7198	0.7456	0.9561
		XGBoost	0.7981	0.6412	0.7111	0.7775	0.9605
43	88.6	LR	0.0646	0.3793	0.1104	0.0722	0.7999
		RF	0.4286	0.0207	0.0395	0.1800	0.9113
		XGBoost	0.5455	0.0828	0.1437	0.4033	0.9581

F1-score on the tuning validation set is selected and refit on the full official training period before being evaluated once on the official test set.

Table 4 compares the manually configured (fixed) and tuned configurations on the official test set. For Random Forest, tuning yields a meaningful improvement in illicit-class F1-score (from 0.7562 to 0.7974), indicating that the manually chosen configuration reported in Section 4 was reasonable but not optimal. For XGBoost, in contrast, the tuned configuration achieves essentially the same F1-score as the manually configured model (0.7771 versus 0.7878), with a marginally lower precision offset by a comparable recall. This indicates that the manually configured XGBoost model used throughout this study was already close to its achievable performance ceiling on this split, and that its reported superiority over Random Forest and Logistic Regression in Section 4 is not an artifact of favorable default settings.

Table 4 Illicit-class test performance before and after hyperparameter tuning, using a temporally-respecting validation split for the search.

Model	Configuration	Precision	Recall	F1-Score	PR-AUC	ROC-AUC
Random Forest	Fixed (Sec. 4)	0.9875	0.6127	0.7562	0.7595	0.9288
	Tuned	0.9837	0.6704	0.7974	0.7553	0.9254
XGBoost	Fixed (Sec. 4)	0.9442	0.6759	0.7878	0.7751	0.9355
	Tuned	0.9080	0.6792	0.7771	0.7684	0.9352

3.9 Reproducibility Details

To support independent verification and reuse of the reported results, this subsection consolidates the implementation details needed to reproduce the experimental pipeline. The dataset is the publicly available Elliptic Bitcoin transaction dataset (Section 3.2); after filtering to labeled transactions, the temporal train–test split uses time-step cutoff 35 (Section 3.4). All three models are implemented in Python 3 using `scikit-learn` (Logistic Regression with `StandardScaler` preprocessing and `class_weight="balanced"`, `max_iter=2000`; Random Forest with `n_estimators=300` and `class_weight="balanced"`) and `XGBoost` (`n_estimators=500`, `max_depth=6`, `learning_rate=0.05`, `subsample=0.8`, `colsample_bytree=0.8`, `reg_lambda=1.0`, with `scale_pos_weight` set to the training-set licit-to-illicit ratio). All models use a fixed random seed (`random_state=42`) for the results reported in Section 4; Section 4.2 additionally reports results under two further seeds (7 and 123) to characterize seed-to-seed variability. The randomized hyperparameter search described above uses a temporally-ordered internal validation split (time-steps 1–28 for sub-training, 29–35 for validation) and 8–10 sampled configurations per model, drawn from search ranges of {200–600} estimators, {4–10} maximum tree depth, {0.01–0.2} learning rate (XGBoost only), {0.6–1.0} subsampling and column-sampling ratios, and {0.1–10} L2 regularization strength. SHAP values (Section 3.10) are computed using `TreeExplainer` on the fitted XGBoost model over the test-set transactions. Given the space constraints of the manuscript format, the full pipeline code and analysis scripts are not reproduced verbatim here; the authors will make the code, together with a requirements file pinning exact package versions, available upon reasonable request to interested researchers, consistent with the Elliptic dataset's own terms of use.

Use of AI-assisted tools. Generative AI tools (Claude, Anthropic) were used to format the manuscript to the journal's requirements, to make corrections and wording edits at the authors' direction, and to assist in preparing and running the code for the additional robustness analyses (bootstrap confidence intervals, paired significance tests, the threshold analysis, and the sensitivity checks). The authors reviewed the code, the outputs, and the text, and take full responsibility for the final text.

3.10 Model Explainability

Predictive performance alone is insufficient for AML applications, where compliance officers and regulators require transparent justifications for automated alerts. Accordingly, this study integrates post-hoc model explainability using SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017), a game-theoretic framework grounded in cooperative game theory that assigns each input feature a contribution value consistent with the model's output for a given prediction.

Formally, the SHAP value ϕ_i for feature i is defined as the weighted average of the marginal contributions of feature i across all possible feature coalitions (Lundberg and Lee, 2017):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{1}$$

where F is the set of all features, S is a subset of features excluding feature i, and $f_S(x_S)$ denotes the model output using only the features in S. SHAP satisfies three desirable axiomatic properties: *local accuracy* (the sum of SHAP values equals the model output), *missingness* (features absent from the input receive zero contribution), and *consistency* (a feature's SHAP value does not decrease if its marginal contribution increases) (Lundberg and Lee, 2017).

SHAP is applied to the best-performing model (XGBoost) identified in the classification experiments. Two levels of analysis are conducted:

1. **Global explainability**: SHAP summary plots are used to rank features by their mean absolute SHAP value across all test-set transactions, identifying the most influential features for the illicit class overall.
2. **Local explainability**: SHAP waterfall plots are used to decompose individual transaction predictions, illustrating how specific feature values push the predicted probability toward or away from the illicit class for a single transaction instance.

This dual-level analysis bridges the gap between aggregate model performance and transaction-level audit trails, directly supporting the operational requirements of AML analysts who must justify flagged transactions to compliance officers and regulators (Lundberg and Lee, 2017). The explainability subsection is consistent with and directly aligned with Stage 6 of the end-to-end pipeline illustrated in Figure 1.

3.11 Evaluation Metrics

Model performance is evaluated using a set of metrics specifically selected to address the challenges of imbalanced classification and the operational demands of anti-money-laundering (AML) systems. Because illicit transactions constitute only a small fraction of the dataset, relying solely on overall accuracy can produce misleading conclusions. Accordingly, this study prioritizes class-specific and imbalance-aware metrics, with particular emphasis on the illicit class.

Precision, recall, and F1-score are calculated for the illicit transaction class. Precision represents the proportion of transactions predicted as illicit that are truly illicit, thereby reflecting the model's capacity to limit false positives and reduce unnecessary investigative costs. Recall measures the proportion of actual illicit transactions correctly identified by the model and is especially critical in AML contexts, where failing to detect illicit activity poses substantial risk. The F1-score provides a balanced summary metric that jointly accounts for precision and recall.

In addition to these class-specific measures, the receiver operating characteristic area under the curve (ROC-AUC) is used to evaluate the overall discriminative ability of each model across varying decision thresholds. However, in the presence of class imbalance, ROC-AUC may present an overly optimistic view of performance due to the dominance of the majority class. For this reason, greater weight is given to the precision–recall area under the curve (PR-AUC), which offers a more informative assessment of performance for rare illicit events by concentrating on the positive class.

4 Results

This section presents and analyzes the experimental results obtained from applying the machine learning pipeline to the Elliptic dataset. Model performance is evaluated using imbalance-aware metrics, and the findings are examined with respect to detection effectiveness, robustness, and explainability.

4.1 Performance Comparison

Table 5 presents a comparative evaluation of the machine learning models, with particular emphasis on performance for the illicit transaction class, which constitutes the central focus of anti-money-laundering (AML) analysis. Owing to the pronounced class imbalance in the dataset, model performance is assessed primarily using precision, recall, F1-score, PR-AUC, and ROC-AUC, rather than overall accuracy. As noted in Section 3.4, the illicit rate differs between the training partition (approximately 11.67%) and the test partition on which the results below are computed (approximately 5.88%); this lower test-set prevalence lowers the achievable PR-AUC baseline relative to the training partition and should be kept in mind when interpreting the PR-AUC values reported in this section.

Logistic Regression achieves a high recall for illicit transactions (0.7913), indicating that it successfully detects a substantial proportion of illicit activity. However, this sensitivity is accompanied by extremely low precision (0.1762) and a modest F1-score (0.2883), implying that the majority of transactions flagged as illicit are false positives. This outcome reflects the limitations of linear models in distinguishing complex illicit transaction patterns and renders Logistic Regression unsuitable for operational AML applications despite its strong recall.

Random Forest attains the highest precision (0.9875), demonstrating that nearly all transactions classified as illicit are correctly identified. Such effective control of false positives is particularly valuable in compliance contexts where investigative resources are constrained. Nevertheless, the model exhibits lower recall (0.6127), indicating that a meaningful proportion of illicit transactions remain undetected. The corresponding F1-score (0.7562) and PR-AUC (0.7595) characterize the model as adopting a strong yet relatively conservative detection strategy.

XGBoost provides the most balanced trade-off between precision and recall, achieving a precision of 0.9442 and a recall of 0.6759, which together yield the highest F1-score for the illicit class (0.7878). Furthermore, XGBoost records the highest PR-AUC (0.7751) and ROC-AUC (0.9355), demonstrating superior discriminative performance under class imbalance. These findings indicate that gradient boosting is particularly effective in modeling nonlinear and higher-order feature interactions in blockchain transaction data, positioning XGBoost as the most appropriate model for illicit Bitcoin transaction detection in this study.

4.2 Statistical Significance of Model Differences

To assess whether the performance differences reported in Table 5 reflect genuine differences between models rather than sampling variation on a single test partition, we performed two complementary analyses. First, we computed 95% confidence intervals for

Table 5 Performance comparison of machine learning models on the illicit class.

Model	Precision (Illicit)	Recall (Illicit)	F1-Score (Illicit)	PR-AUC	ROC-AUC
Logistic Regression	0.1762	0.7913	0.2883	0.3142	0.8777
Random Forest	0.9875	0.6127	0.7562	0.7595	0.9288
XGBoost	0.9442	0.6759	0.7878	0.7751	0.9355

each metric via percentile bootstrap resampling of the fixed test set (1,000 resamples). The resulting intervals are: Logistic Regression F1 $\in [0.273, 0.307]$, Random Forest F1 $\in [0.733, 0.780]$, and XGBoost F1 $\in [0.766, 0.808]$. The Random Forest and XGBoost intervals overlap slightly (approximately 0.766–0.780), which under a naive interval-overlap heuristic might be read as an inconclusive difference between the two ensemble methods.

However, interval overlap is not by itself a valid significance test when the two models are evaluated on the *same* test instances, because their predictions are paired and correlated; an overlap-based judgment ignores this pairing and is conservative to the point of being misleading. We therefore additionally applied a continuity-corrected McNemar's test, which is the standard paired test for comparing two classifiers on identical test data. The results confirm that all three pairwise differences are statistically significant: Logistic Regression versus Random Forest ($\chi^2 = 2756.14$, $p < 0.001$), Logistic Regression versus XGBoost ($\chi^2 = 2850.12$, $p < 0.001$), and, notably, Random Forest versus XGBoost ($\chi^2 = 6.33$, $p = 0.012$), despite the latter pair's overlapping bootstrap intervals. This illustrates a well-known statistical nuance: bootstrap CI overlap on a shared test set is a weak, overly conservative heuristic for paired classifier comparison, and a dedicated paired test such as McNemar's is required to draw a valid significance conclusion. On this basis, we conclude that XGBoost's improvement over Random Forest, while numerically modest (F1 0.7878 versus 0.7562), is statistically robust rather than attributable to test-set sampling noise.

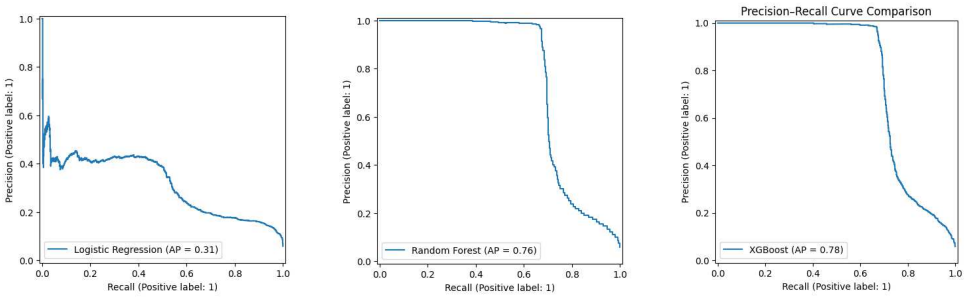
As a secondary robustness check, we also refit each model under three random seeds (42, 7, 123) with the train–test partition held fixed. Random Forest and XGBoost show low across-seed variability (F1 standard deviation of 0.0033 and 0.0017, respectively), while Logistic Regression is essentially deterministic under `liblinear`-style convergence (F1 standard deviation < 0.0001), confirming that the reported metrics are stable with respect to model-internal randomness and are not an artifact of a favorably chosen random seed.

4.3 Precision–Recall Analysis

Figure 4 displays the precision–recall (PR) curves for the evaluated models, offering a detailed assessment of performance under class imbalance. Because illicit transactions constitute the minority class, PR analysis is particularly appropriate, as it highlights the trade-off between accurately identifying illicit activity (recall) and minimizing false positives (precision). Accordingly, the area under the precision–recall curve (PR-AUC) is used as a principal measure of model effectiveness in this setting.

Logistic Regression demonstrates relatively weak PR performance, with an average precision (AP) of approximately 0.31. The curve exhibits a steep decline in precision as recall increases, indicating that efforts to detect a larger share of illicit transactions lead to a substantial increase in false positives. This pattern underscores the limitations of linear decision boundaries in capturing complex transactional structures and confirms the limited suitability of logistic regression for AML tasks involving rare illicit events.

By contrast, the ensemble-based models exhibit significantly stronger PR characteristics. Random Forest achieves a considerably higher average precision of approximately 0.76, maintaining elevated precision across a wide recall range before gradually declining at higher recall levels. XGBoost further enhances this performance, attaining the highest average precision of approximately 0.78 and consistently demonstrating a more favorable precision–recall trade-off across most operating thresholds. These results suggest that gradient boosting more effectively detects illicit transactions while controlling false alarms, making XGBoost particularly well suited for real-world AML applications that require both high detection capability and operational efficiency.



(a) Logistic Regression (AP = 0.31) (b) Random Forest (AP = 0.76) (c) XGBoost (AP = 0.78)

Fig. 4 Precision–Recall curve comparison across models on the illicit class: (a) Logistic Regression, (b) Random Forest, (c) XGBoost

4.4 Confusion Matrix Analysis

Figure 5 presents the confusion matrix for the best-performing model, XGBoost, offering a detailed summary of classification outcomes for both licit and illicit Bitcoin transactions. The confusion matrix provides an intuitive representation of model behavior by explicitly distinguishing between correct classifications and misclassifications, which is particularly valuable for assessing operational trade-offs in anti-money-laundering (AML) systems.

The results show that XGBoost correctly classifies 14,392 licit transactions as licit and 609 illicit transactions as illicit, indicating strong predictive performance across both classes. Only 36 licit transactions are incorrectly labeled as illicit, corresponding to a very low false positive rate. This characteristic is especially desirable in AML settings, where excessive false positives can impose significant investigative burdens and increase compliance costs.

For the illicit class, the model incorrectly classifies 292 illicit transactions as licit, representing false negatives. Although false negatives are of greater concern in AML applications, since they correspond to undetected illicit activity, the observed number remains relatively small in relation to the total number of illicit transactions. Overall, the confusion matrix demonstrates that XGBoost achieves a well-balanced trade-off between accurately detecting illicit transactions and minimizing false alarms, reinforcing its suitability for practical deployment in blockchain-based AML monitoring systems.

Expressed as class-conditional (row-normalized) percentages, of the 14,428 actual licit transactions in the test set, 99.75% are correctly classified as licit and 0.25% are misclassified as illicit; of the 901 actual illicit transactions, 67.59% are correctly classified as illicit and 32.41% are misclassified as licit. These percentages are, by construction, identical to the precision-denominator and recall values already reported in Table 5 (recall = 0.6759), since recall is exactly the true-positive percentage within the actual-illicit row of the confusion matrix; we report both forms here because the raw counts in Figure 5 are most informative for judging absolute investigative workload, whereas the percentages are more directly comparable across studies that use test sets of different sizes. We also note, for clarity, that the average precision (AP) values annotated in Figure 4 (0.31, 0.76, and 0.78 for Logistic Regression, Random Forest, and XGBoost, respectively) are the same quantity as the PR-AUC column reported in Table 5 (0.3142, 0.7595, and 0.7751), rounded to two decimal places for figure legibility; the two are not independent measurements and do not represent a discrepancy.

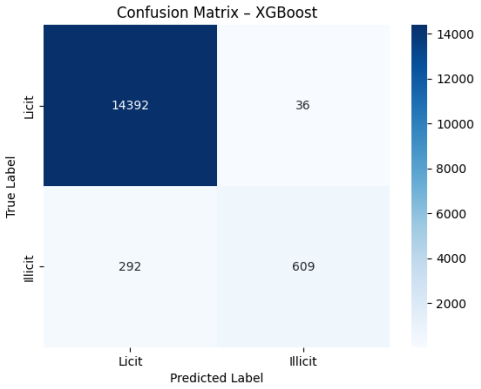


Fig. 5 Confusion matrix of the XGBoost model on the temporal test set

Is the Observed Recall Operationally Acceptable, and Is It Threshold-Limited?

XGBoost's recall of 0.6759 means that roughly one in three illicit transactions in the test partition is missed at the default 0.5 decision threshold. Because AML deployments can typically tolerate a controlled investigative workload rather than a fixed classification threshold, we examined whether lowering the decision threshold recovers materially

more recall at an acceptable precision cost. Table 6 reports XGBoost's precision, recall, and F1-score at five thresholds from 0.5 down to 0.1.

Table 6 XGBoost operating points at decreasing decision thresholds on the temporal test set.

Threshold	Precision	Recall	F1-Score	Confusion Matrix (TN, FP; FN, TP)
0.5	0.9385	0.6770	0.7866	14388, 40; 291, 610
0.4	0.9123	0.6815	0.7802	14369, 59; 287, 614
0.3	0.8813	0.6837	0.7700	14345, 83; 285, 616
0.2	0.8141	0.6903	0.7471	14286, 142; 279, 622
0.1	0.7099	0.6981	0.7040	14171, 257; 272, 629

Even at the most aggressive threshold examined (0.1, an 80% reduction from the default), recall improves only marginally, from 0.677 to 0.698, while precision collapses from 0.939 to 0.710 and the false-positive count rises more than six-fold (40 to 257). This asymmetric response indicates that the residual 30% of undetected illicit transactions are not simply sitting just below the default decision boundary waiting to be recovered by a threshold adjustment; rather, they are cases for which the model's learned feature representation places them close to, or on the wrong side of, the licit distribution, i.e., a genuine separability limitation of the feature set and model rather than a threshold-tuning artifact. Consequently, closing this recall gap further would likely require additional information sources (e.g., graph-structural features, richer temporal context, or multi-hop neighbourhood aggregation beyond one hop) rather than post-hoc threshold recalibration alone.

From an operational standpoint, this analysis clarifies the trade-off available to a compliance team: a lower threshold (e.g., 0.3) offers a modest recall gain (610 to 616 detected illicit transactions, six additional detections) at the cost of more than double the false-positive investigative burden (40 to 83 flagged licit transactions), a trade-off that most AML operations would not consider favorable given typical investigative capacity constraints. We therefore retain the default 0.5 threshold as the primary operating point reported in Table 5, while noting that threshold selection in a live deployment should ultimately be calibrated against the specific investigative capacity and risk appetite of the institution, rather than fixed a priori by the model developer.

Operational Implications for AML Compliance

Translating these results into an operational reading: at the default threshold, deploying this pipeline over a transaction population with the same illicit prevalence as the test

partition (901 illicit out of 15,329 transactions, approximately 5.9%) would generate 650 alerts (610 true positives plus 40 false positives), of which 93.8% would correspond to genuinely illicit activity, a substantially lower investigative burden than a naive rule-based screen with comparable recall would typically produce, since precision this high is unusual for AML alerting systems in practice. The 291 missed illicit transactions (false negatives), however, represent the compliance-relevant residual risk that this model, in isolation, does not address; consistent with the risk-based approach that the Financial Action Task Force (FATF) recommends for virtual-asset service providers (FATF, 2021), we emphasize that a model of this kind should be positioned as one layer within a broader, defense-in-depth AML monitoring program, e.g., combined with network-based or rule-based screens targeting known typologies, rather than as a sole detection mechanism. This framing is consistent with FATF guidance that risk-based AML controls should be proportionate and layered rather than reliant on a single automated system (FATF, 2021).

4.5 Explainability Analysis Using SHAP

Figure 6 displays the SHAP (SHapley Additive exPlanations) beeswarm plot for the XGBoost model, offering a global perspective on how individual features contribute to illicit transaction predictions. The beeswarm plot ranks features by their mean absolute SHAP value and simultaneously depicts the magnitude, direction, and feature-value dependence of each contribution. Each point represents a single transaction; the color gradient encodes the feature value from low (blue) to high (red), while the horizontal position on the x-axis indicates whether that feature value pushes the prediction toward the illicit class (positive SHAP) or the licit class (negative SHAP).

Feature Categorization

As noted in Section 3.2, the 165 features fall into two structural groups: `feat_1`–`feat_94` capture local transaction-level attributes, while `feat_95`–`feat_165` capture aggregated one-hop neighbourhood statistics. This grouping is used below to interpret the SHAP results, even in the absence of precise semantic labels for individual features.

Feature Collinearity and Redundancy

Because the neighbourhood features (`feat_95`–`feat_165`) are constructed by aggregating the same local attributes (`feat_1`–`feat_94`) over one-hop transaction neighbours, it is reasonable to ask whether the two blocks are largely redundant with one another, which would in turn undermine the independence of the SHAP attributions discussed above. To address this, we computed the full 165×165 Pearson correlation matrix over all engineered features on the complete labeled dataset. Of the 13,530 unique feature pairs, only 126 (0.93%) exhibit $|r| > 0.9$ and 191 (1.41%) exhibit $|r| > 0.8$; the mean absolute pairwise correlation is low (0.086), indicating that severe multicollinearity is confined to a small minority of feature pairs rather than being a global property of the feature space. The most strongly correlated pairs (several with $|r| \approx 1.0$, e.g., `feat_23`/`feat_29`, `feat_58`/`feat_64`) are concentrated within the same structural

block (local–local or neighbourhood–neighbourhood), consistent with these being different temporal or directional aggregates (e.g., maximum versus minimum) of a single underlying quantity.

Critically, cross-block correlation between the local features and the neighbourhood features is weak: the mean absolute cross-block correlation is only 0.049, and only 1 of the 94×71 cross-block pairs exceeds $|r| > 0.9$ (maximum cross-block $|r| = 0.973$). This indicates that, despite being derived from the local attributes, the neighbourhood-aggregated features capture substantially different information than their local counterparts, most likely because aggregation over a transaction's one-hop neighbours reflects the behavior of *other* transactions rather than the focal transaction itself. This finding supports treating the two feature groups as largely complementary, rather than redundant, in the SHAP interpretation above.

A principal component analysis (PCA) on standardized features further quantifies the overall redundancy: 48 of the 165 components are needed to explain 90% of the total variance, 59 components for 95%, and 74 components for 99%, with the first component alone explaining only 11.9% of variance and the first ten components jointly explaining 47.0%. This indicates a moderate, but not extreme, degree of redundancy—consistent with the pairwise correlation results—and confirms that no single feature or small feature cluster dominates the variance structure of the dataset. We note that the tree-based models used in this study (Random Forest, XGBoost) are inherently robust to correlated features, since correlated predictors simply compete for split selection rather than destabilizing coefficient estimates as they could in Logistic Regression; the moderate collinearity identified here is therefore unlikely to materially affect the comparative model rankings reported in Section 4.

Interpretation of Top Features

A caveat applies to the entire discussion below. As stated in Section 3.2, the Elliptic dataset provider anonymizes all 165 features and does not release a mapping from feature index to a specific economic quantity (Weber et al., 2019); only the coarse local-versus-neighbourhood structural grouping is documented. Consequently, the semantic labels proposed below for individual top-ranked features (e.g., associating `feat_53` with output volume, or `feat_90` with a fee-to-volume ratio) are plausibility-motivated interpretive hypotheses consistent with each feature's SHAP behavior and its position within the documented local or neighbourhood block, not confirmed semantic identifications, since no ground-truth feature dictionary is available to verify them against. We present them because they generate testable hypotheses for practitioners with access to the underlying transaction data, and we flag this limitation explicitly rather than presenting the interpretations as established fact.

The SHAP analysis reveals that a small subset of features accounts for the majority of the model's predictive behavior, with the top 20 features shown in Figure 6.

feat_53 is the single most influential feature, exhibiting the widest SHAP value range (approximately −3 to +2) among all features. Its index places it firmly within the local feature group (`feat_1`–`feat_94`), consistent with a transaction-level attribute such as output volume or an aggregated spending pattern associated with the transaction's inputs or outputs. The plot shows that high values of `feat_53` are predominantly

associated with strongly positive SHAP values, meaning transactions with elevated values of this feature are disproportionately flagged as illicit. This pattern is consistent with the known behavior of illicit actors who consolidate or distribute large volumes of Bitcoin across multiple outputs to layer proceeds, "layering" being the second of the three classical stages of money laundering (placement, layering, integration) formalized in FATF typology guidance (FATF, 2021), in which illicit funds are moved through successive transactions to obscure their origin.

feat_90 is the second most influential feature and likewise belongs to the local feature group. The beeswarm pattern for `feat_90` shows a bimodal distribution, with both high and low feature values contributing positively to the illicit prediction in different subsets of transactions. This non-monotonic SHAP pattern suggests that `feat_90` may capture a ratio or normalized metric (such as average BTC per output or a fee-to-volume ratio) where both extreme values, unusually high or unusually low, signal atypical transaction behavior associated with illicit activity, such as either dust attacks, a technique in which attackers send tiny, uneconomical amounts of Bitcoin to many addresses to de-anonymize wallet clusters, a pattern documented in blockchain forensics and AML typology literature (FATF, 2021), or large-value single-output transfers.

feat_59 and **feat_3**, ranked third and fourth respectively, are also local features. Their SHAP distributions display a consistent directional pattern: high feature values are associated with positive (illicit-indicating) SHAP contributions, while low values are associated with near-zero or negative contributions. Given their index positions, these features likely encode aspects of transaction throughput or connectivity (e.g., number of inputs, number of outputs, or related aggregated counts), which are known discriminators between normal peer-to-peer transactions and those generated by automated mixing services or tumbling protocols that artificially fragment transactions to obscure the flow of funds (Foley et al., 2019).

feat_125 and **feat_2**, ranked fifth and sixth, straddle the boundary between the local and aggregated feature groups. `feat_2` (local) likely encodes a basic transaction attribute such as the raw transaction fee or a count variable. `feat_125` (aggregated neighbourhood feature, index > 94) captures statistical properties of neighbouring transactions—for example, the standard deviation or maximum of a local attribute across one-hop neighbours. Its presence among the top features confirms that the transaction's *network context*, not just its own attributes, contributes meaningfully to illicit classification. This is consistent with the guilt-by-association principle documented in prior blockchain AML research (Weber et al., 2019): illicit transactions tend to cluster with other illicit transactions in the payment graph, so neighbourhood-level features carry discriminative signal even in a non-graph model.

Features ranked 7 through 20 (`feat_101`, `feat_16`, `feat_58`, `feat_46`, `feat_163`, `feat_55`, `feat_161`, `feat_65`, `feat_138`, `feat_52`, `feat_106`, `feat_89`, `feat_29`, `feat_41`) constitute a secondary tier of contributors. Several of these (indices > 94: `feat_101`, `feat_138`, `feat_161`, `feat_163`) belong to the aggregated neighbourhood group, reinforcing the finding that graph-structural context adds predictive value even within a tabular model. Their narrower SHAP value distributions relative to the top features indicate more moderate but consistent contributions across transactions.

Global Patterns and AML Implications

Three broader patterns emerge from the SHAP analysis. First, the dominance of local features among the top-ranked predictors confirms that transaction-level attributes, likely related to output structure, fee behavior, and volume, are the primary discriminators of illicit activity in this dataset. Second, the inclusion of aggregated neighborhood features in the top 20 demonstrates that even a non-graph model implicitly benefits from contextual signals, suggesting that the transaction's position in the payment network is encoded, at least partially, in the feature representation. Third, the considerable dispersion of SHAP values across transactions for the top features, particularly `feat_53` and `feat_90`, reflects heterogeneous behavioral patterns within the illicit class, consistent with the diverse typologies (ransomware, darknet markets, scams, Ponzi schemes) present in the Elliptic dataset (Foley et al., 2019).

These results have direct operational implications for AML analysts. The concentration of predictive signal in a small number of local and neighbourhood features suggests that AML dashboards can be streamlined to present a compact set of the most diagnostically relevant transaction attributes, reducing cognitive load during case review. Moreover, the ability to associate specific SHAP contributions with feature-group semantics, even without precise feature labels, provides a principled basis for escalating flagged transactions for human review, satisfying the interpretability requirements central to regulatory compliance frameworks (Lundberg and Lee, 2017).

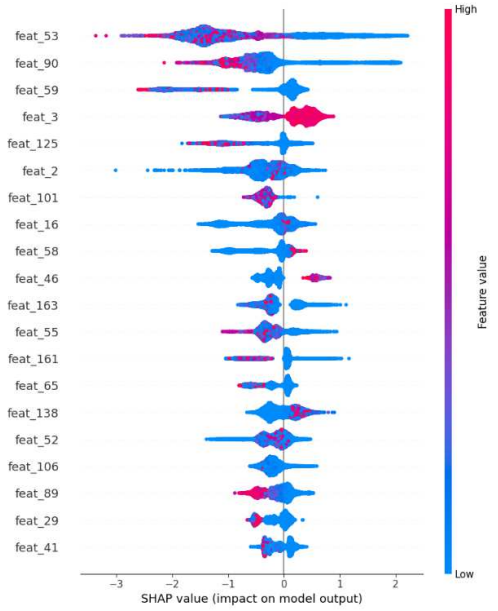


Fig. 6 SHAP beeswarm plot of the top 20 features for the XGBoost model

5 Conclusions and Future Work

This study examined the effectiveness of machine learning methods for detecting illicit Bitcoin transactions using the Elliptic dataset, with particular attention to class imbalance and model interpretability. By comparing a linear baseline with ensemble-based approaches within a temporally realistic evaluation framework, the research offers a comprehensive analysis of how different modeling strategies perform in blockchain-based anti-money-laundering (AML) settings.

The empirical results indicate that ensemble methods substantially outperform linear models in identifying illicit transactions. Although Logistic Regression achieves high recall, its low precision results in an impractically large number of false positives. Random Forest markedly improves precision but does so at the expense of reduced recall. XGBoost provides the most balanced performance, attaining the highest illicit-class F1-score and PR-AUC, as well as the strongest overall discriminative ability measured by ROC-AUC. These findings demonstrate that gradient boosting is particularly effective in capturing the complex and nonlinear transaction patterns characteristic of cryptocurrency networks.

Beyond predictive accuracy, the study highlights the importance of model transparency through SHAP-based explainability. The analysis shows that a relatively small subset of transaction features accounts for the majority of illicit predictions, offering meaningful insights into behavioral patterns associated with financial crime on the Bitcoin blockchain. Such interpretability is crucial in practical AML applications, where regulatory requirements and human oversight demand clear and defensible explanations for automated decisions.

From an applied perspective, the evaluated pipeline illustrates how machine learning can support scalable and interpretable AML monitoring in cryptocurrency environments. The integration of temporal validation, imbalance-aware evaluation, and explainable AI strengthens the robustness and practical relevance of the approach for deployment by financial institutions, blockchain analytics firms, and regulatory authorities.

For a compliance team, the confusion matrix of the XGBoost model (Section 4.4) translates into a concrete workload: of the 645 alerts raised on the test period, 609 were genuine illicit transactions and 36 were false alarms, while 292 illicit transactions went undetected. The Logistic Regression baseline, whose precision is 0.1762, would instead raise about 4.7 false alerts for every genuine one. These counts show that the practical trade-off for an AML analyst is not between accuracy levels but between alert volume and the share of illicit activity that is missed.

Despite these contributions, several limitations remain. The analysis is confined to a single labeled dataset and excludes transactions with unknown labels, which may constrain generalization to novel or emerging forms of illicit activity. Furthermore, the study relies on feature-based models and does not explicitly incorporate graph neural network architectures that could further exploit transaction connectivity. The models are also static in the sense that they are trained once on historical time-steps and evaluated on later ones without any mechanism for continued adaptation; the split-ratio sensitivity analysis in Section 3.7 confirms directly within this pipeline that performance on the illicit class deteriorates sharply once the test period is dominated by transactions following the dark-marketplace closure at time-step 43, consistent

with prior reports of distributional shift on this dataset (Alarab et al., 2020a), and the present study does not attempt to correct for this. In addition, the scope of this work is limited to Bitcoin. Extending the proposed pipeline to account-based ledgers such as Ethereum is not a straightforward transfer, since the underlying transaction model differs substantially from Bitcoin's unspent transaction output (UTXO) structure, and the engineered features used here would need to be redefined accordingly. Finally, this study deliberately adopts a fully supervised approach and does not draw on the unlabeled majority of the dataset, in contrast to self-supervised or actor-level approaches such as Elliptic++ (Elmougy and Liu, 2023) and Inspection-L (Lo et al., 2023); this choice reflects the emphasis placed here on model auditability and interpretability, which is more directly supported by supervised tree-based ensembles than by representation-learning approaches.

Future research may extend this work in several directions. Incorporating graph-based learning techniques could enhance detection performance by leveraging network structure more directly. Given the label scarcity inherent to the Elliptic dataset, exploring self-supervised pre-training or active-learning strategies over the unlabeled transactions, in the spirit of Elliptic++ and Inspection-L, could complement the interpretability-focused approach adopted here. Addressing concept drift explicitly, for instance through incremental or online learning schemes that periodically update the model as new time-steps become available, would also be a natural extension for deployment settings where transaction behavior continues to evolve after training. Finally, adapting the framework to account-based blockchains such as Ethereum, and evaluating it across multiple cryptocurrency networks, would help establish the broader applicability of the proposed approach beyond Bitcoin.

Statements and Declarations

Funding

No funding was received for conducting this study.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval

The study was conducted under Gole Research and Consultancy. It used only the publicly available, anonymized Elliptic Bitcoin transaction dataset and involved no human participants, no animals, and no personally identifiable information. Formal ethics approval was therefore not required.

Consent to participate

Not applicable. The study did not involve human participants; it analysed a publicly available, anonymized dataset of Bitcoin transactions.

Consent to publish

Not applicable. The manuscript does not contain any individual person's data, images, or other identifying details.

Author contributions

Abdikadir Hussein Elmi conceived the study, designed the methodology, carried out the experiments, and wrote the original draft of the manuscript. Abdijalil Abdullahi contributed to the literature review and to the interpretation of the SHAP results. Abdirisak Ahmed Salah contributed to data curation and to the design of the temporal evaluation protocol. Abdimalik Osman Hassan contributed to the statistical validation and to the preparation of the figures and tables. Abdirahman Mohamud Abdi contributed to the interpretation of the results from the anti-money-laundering and compliance perspective and to the review of the manuscript. Saed Mohamud Muse contributed to the validation of the results and to the review and editing of the manuscript. All authors revised the manuscript critically, read and approved the final version, and agree to be accountable for the work.

Data availability

The dataset analysed during the current study is the publicly available Elliptic Bitcoin transaction dataset (Weber et al., 2019), accessible at `https://www.kaggle.com/datasets/ellipticco/elliptic-data-set`. No new datasets were generated during the current study.

Code availability

The analysis code and a requirements file listing the package versions used are available from the corresponding author upon reasonable request.

References

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020a). Comparative analysis using supervised learning methods for anti-money laundering in Bitcoin. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies* (pp. 11–17). ACM. https://doi.org/10.1145/3409073.3409078

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020b). Competence of graph convolutional networks for anti-money laundering in Bitcoin blockchain. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies* (pp. 23–27). ACM. https://doi.org/10.1145/3409073.3409080

Alarab, I., & Prakoonwit, S. (2023). Graph-based LSTM for anti-money laundering: Experimenting temporal graph convolutional network with Bitcoin data. *Neural Processing Letters, 55*(1), 689–707. https://doi.org/10.1007/s11063-022-10904-8

Arner, D. W., Barberis, J., & Buckey, R. P. (2016). FinTech, RegTech, and the reconceptualization of financial regulation. *Northwestern Journal of International Law & Business, 37*, 371.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform* [White paper].

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences, 557*, 317–331. https://doi.org/10.1016/j.ins.2019.05.042

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Elmougy, Y., & Liu, L. (2023). Demystifying fraudulent transactions and illicit nodes in the Bitcoin network for financial forensics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3979–3990). ACM. https://doi.org/10.1145/3580305.3599803

Financial Action Task Force. (2021). *Updated guidance for a risk-based approach to virtual assets and virtual asset service providers*. FATF. https://www.fatf-gafi.org/en/publications/Fatfrecommendations/Guidance-rba-virtual-assets-2021.html

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

Hasan, M., Rahman, M. S., Chowdhury, M. J. M., & Sarker, I. H. (2025). CNN based deep learning modeling with explainability analysis for detecting fraudulent blockchain transactions. *Cyber Security and Applications, 3*, 100101. https://doi.org/10.1016/j.csa.2025.100101

Hasan, M., Rahman, M. S., Janicke, H., & Sarker, I. H. (2024). Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. *Blockchain: Research and Applications, 5*(3), 100207.

Hosmer, D. W., Jr., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). John Wiley & Sons. https://doi.org/10.1002/9781118548387

Kungu, C. O., Senagi, K., & Omondi, E. (2026). Hybrid deep learning for anti-money laundering: Unsupervised detection of emerging schemes via feature fusion and explainable artificial intelligence. *Machine Learning with Applications, 23*, 100856. https://doi.org/10.1016/j.mlwa.2026.100856

Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., & Portmann, M. (2023). Inspection-L: Self-supervised GNN node embeddings for money laundering detection in Bitcoin. *Applied Intelligence, 53*(16), 19406–19417.

Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T., & Bizarro, P. (2020). Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. In *Proceedings of the First ACM International Conference on AI in Finance* (pp. 1–8). ACM. https://doi.org/10.1145/3383455.3422549

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (Vol. 30).

Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Hogan, K., Hennessey, J., Miller, A., Narayanan, A., & Christin, N. (2018). An empirical analysis of traceability in the Monero blockchain. *Proceedings on Privacy Enhancing Technologies, 2018*(3), 143–163. https://doi.org/10.1515/popets-2018-0025

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., & Leiserson, C. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(4), 5363–5370. https://doi.org/10.1609/aaai.v34i04.5984

Poursafaei, F., Rabbany, R., & Zilic, Z. (2021). SigTran: Signature vectors for detecting illicit activities in blockchain transaction networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 27–39). Springer. https://doi.org/10.1007/978-3-030-75762-5_3

Schnoering, H., & Vazirgiannis, M. (2025). Bitcoin research with a transaction graph dataset. *Scientific Data, 12*(1), 404. https://doi.org/10.1038/s41597-025-04684-8

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. In *KDD '19 Workshop on Anomaly Detection in Finance*. ACM. https://doi.org/10.48550/arXiv.1908.02591

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

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open, 1*, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001