An Explainable Machine-Learning Framework for Detecting fraudulent Bitcoin Addresses with Graph–Temporal Features

Mario Trerotola[1], Davide Calvaresi[2], and Mimmo Parente[3]

[1] Politecnico di Torino, Torino, Italy
mario.trerotola@polito.it
[2] University of Applied Sciences of Western Switzerland, Sierre, Switzerland
davide.calvaresi@hevs.ch
[3] Università degli Studi di Salerno, Fisciano, Italy
parente@unisa.it

Abstract. The pseudonymous nature of blockchain transactions poses a significant challenge for identifying fraudulent activity in decentralized financial systems. This study presents a comprehensive framework for classifying Bitcoin wallets as fraudulent or legitimate by integrating multi-source data, graph-based transaction modeling, and machine learning. Our methodology builds upon publicly available datasets—namely Elliptic++, Chainabuse, and a curated sample of recent transactions—and integrates structural, temporal, and monetary features extracted from the Bitcoin transaction graph. Through systematic experiments across three distinct labeling scenarios, we demonstrate that ensemble methods such as Random Forests offer strong performance even under label noise, achieving F1-scores up to 0.92. Moreover, an explainability framework grounded, in SHAP values, is used to systematically analyze feature contributions and elucidate behavioral patterns associated with financial fraud. Our approach bridges empirical robustness with forensic insight, contributing a scalable, transparent toolset for blockchain compliance and risk analysis.

Keywords: Bitcoin · Blockchain Analytics · Wallet Classification · Financial Fraud · Machine Learning · Explainable AI

1 Introduction

Cryptocurrencies like Bitcoin have redefined financial transactions, enabling decentralized, borderless value transfer. However, their adoption has been accompanied by growing concerns over fraudulent uses including scams, money laundering, and darknet commerce. Identifying fraudulent actors on-chain is a non-trivial task due to the pseudonymous nature of blockchain addresses, the high volume of transactions, and the lack of ground truth labels.

Existing approaches to wallet classification rely on supervised learning, requiring both engineered features and labeled data. Yet, the scarcity of high-quality labels, combined with behavioral variability among actors, complicates

detection efforts. Moreover, heuristic-based clustering approaches have shown promising results in attributing ownership based on transaction patterns.

Our work tackles these challenges through:

- Multi-source data integration and pre-processing strategies for constructing high-quality labeled datasets, inspired by the address and transaction table extraction pipelines used in large-scale studies [**?**].
- A principled set of features capturing structural, temporal, and monetary characteristics of wallets, analogous to transaction patterns such as peel chains, sweep, relay, and self-spending, which have been successfully exploited to improve address clustering [**?**].
- Evaluation of machine learning models under varying data quality scenarios, as similarly conducted through Gini impurity measures in clustering validation [**?**].
- Application of SHAP for model explainability, allowing forensic interpretation of fraud indicators, complementing the heuristic interpretability present in traditional pattern-based techniques.

The structure of this study is organized as follows. Section 2 details our data collection and cleaning pipeline. Section 3 introduces our feature engineering framework. Section 5 describes model training, evaluation results, and comparative analysis and it presents interpretability findings. Section 6 concludes the study by outlining future research directions.

2 Data collection and pre-processing pipeline

This section outlines the data sources, the rationale behind our sampling strategy, and the theoretical foundations of the cleaning operations applied to obtain a high-quality dataset for wallet classification.

Data sources and sampling. Data acquisition proceeded in two main phases. In the first phase, wallet addresses were collected from three primary sources:

- **Elliptic++**[4]: an expanded version of the original Elliptic dataset, comprising 14,267 Bitcoin addresses labeled as fraud and 7,378 labeled as licit, an enriched version of the original Elliptic data set that adds richer entity labels and more recent blocks, aimed at benchmarking anti-money laundering (AML) models[5] [**?**].
- **Chainabuse**[6]: a collection of crowdsourced fraud reports from which 6,910 fraudulent bitcoin addresses were extracted. The entries were gathered

[4] Public release at https://github.com/git-disl/EllipticPlusPlus

[5] Anti-money laundering (AML) models combine human expertise, automated software processes, artificial intelligence (AI) and machine learning (ML) to identify money laundering activities

[6] Chainabuse is an open-source, crowdsourced reporting platform operated by TRM Labs, Binance, and Nansen. It collects addresses associated with scams, ransomware, phishing, and other fraudulent activities. Repository: https://chainabuse.com

by scraping the Chainabuse platform and subsequently filtered to retain only Bitcoin addresses. Corresponding labels were obtained through the `blockchain.info` REST API.
– **Sample of Recent Transactions**: a uniform sample of 10,000 Bitcoin addresses extracted from the most recent 12,000 on-chain transactions, retrieved using https://blockchain.info REST API here again.

3 Graph–Temporal Feature Engineering

To systematically distinguish between fraudulent and legitimate Bitcoin wallets, we commence by extracting all on-chain transactions associated with each wallet of interest. Each wallet is represented as a subgraph of the global Bitcoin transaction network, modeled as a directed multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where:

– $\mathcal{V}$ denotes the set of unique addresses (nodes).
– $\mathcal{E}$ denotes the set of transaction edges, each edge defined as $e = (u, v, \tau, a)$, representing a transfer of amount a from address u to address v at timestamp τ. In the following the components of an edge e will be denoted using dot notation, e.g. the amount a will be denoted $e.a$

From this foundational representation, we derive three complementary categories of features:

Pruning of Low-Activity Addresses. For addresses $u, v \in \mathcal{E}$ let us define the timestamp multisets

$$T_u^{\text{out}} = \{\tau \mid (u, v, \tau, a) \in \mathcal{E}\}, \qquad T_v^{\text{in}} = \{\tau \mid (u, v, \tau, a) \in \mathcal{E}\},$$

with cardinalities n_i^{out} and n_i^{in}. We retain i only if $n_i^{\text{out}} \geq 2$ and $n_i^{\text{in}} \geq 2$, ensuring that

$$\Delta t_i^{\text{out}} = \frac{\max T_i^{\text{out}} - \min T_i^{\text{out}}}{n_i^{\text{out}} - 1}, \quad \Delta t_i^{\text{in}} = \frac{\max T_i^{\text{in}} - \min T_i^{\text{in}}}{n_i^{\text{in}} - 1}$$

are well-defined for both directions. Addresses failing either criterion are discarded from further analysis.

Label Merging and Deduplication. We merge the three strata and resolve conflicts by prioritizing the "fraud" label whenever an address appears in multiple sources. Let $L_i \in \{\text{licit}, \text{fraud}\}$ be the label for address i. The merged label is

$$\widetilde{L}_i = \begin{cases} \text{fraud}, & \exists \text{ source } s : L_i^{(s)} = \text{fraud}, \\ \text{licit}, & \text{otherwise.} \end{cases}$$

Address-Level Metrics. For each address $v \in \mathcal{V}$, we denote the incoming transactions into node v and the outgoing transactions from node v as:

$$\mathcal{I}(v) = \{e \in \mathcal{E} | e = (u, v, \tau, a)\}, \quad \mathcal{O}(u) = \{e \in \mathcal{E} | e = (u, v, \tau, a)\}.$$

We compute the following address-specific metrics:

- **In-degree** $\text{in\_deg}(v) = |\mathcal{I}(v)|$ and **Out-degree** $\text{out\_deg}(v) = |\mathcal{O}(v)|$. The collections $\mathcal{I}(v)$ and $\mathcal{O}(v)$ are formally regarded as *multisets* of incoming and outgoing transactions, respectively, thereby preserving the full transaction multiplicities associated with address v.
- **Unique counterparties**. Let

$$C^{\text{in}}(v) = \{\, u \mid (u, v, \tau, a) \in E \,\}, \qquad C^{\text{out}}(v) = \{\, v \mid (u, v, \tau, a) \in E \,\},$$

  where each element of $C^{\text{in}}(v)$ (resp. $C^{\text{out}}(v)$) appears exactly once, irrespective of transaction multiplicity. We define

$$\text{uniq\_in}(v) = |C^{\text{in}}(v)|, \qquad \text{uniq\_out}(v) = |C^{\text{out}}(v)|,$$

  which quantify the number of *distinct* origin and destination addresses interacting with v, thus capturing the diversity of its transactional counterparties.
- **Average Transaction Amount** $\overline{a}_{\text{in}}(v) = \frac{1}{|\mathcal{I}(v)|} \sum_{e \in \mathcal{I}(v)} e.a$ and likewise for $\overline{a}_{\text{out}}(v)$, reflecting fragmentation or bulk transfer patterns.
- **Mean Inter-transaction Interval** $\overline{\Delta t}_{\text{in}}(v) = \frac{\tau_{\max}(v) - \tau_{\min}(v)}{|\mathcal{I}(v)| - 1}$ (and analogously for outbound edges), indicating temporal regularity or burstiness.
- **Net Balance** $\text{balance}(v) = \sum_{e \in \mathcal{I}(v)} e.a - \sum_{e \in \mathcal{O}(v)} e.a$: measures fund retention versus immediate pass-through.
- **Local Clustering Coefficient** $C(v) = \frac{2\,T(v)}{k(v)(k(v)-1)}$, where $T(v)$ is the count of cycles of length at least 3 containing node v and $k(v) = \text{in\_deg}(v) + \text{out\_deg}(v)$, capturing mixer-like subgraph density.

Wallet-Level Aggregates. Aggregating the address-level metrics across all addresses $v \in \mathcal{V}$ belonging to a given wallet $\mathcal{V}$, we define:

$$D_{\text{in}}(\mathcal{V}) = \sum_{v \in \mathcal{V}} \text{in\_deg}(v), \qquad D_{\text{out}}(\mathcal{V}) = \sum_{v \in \mathcal{V}} \text{out\_deg}(v),$$

$$R_{\text{recv}}(\mathcal{V}) = \sum_{v \in \mathcal{V}} \sum_{e \in \mathcal{I}(v)} e.a, \qquad R_{\text{sent}}(\mathcal{V}) = \sum_{v \in \mathcal{V}} \sum_{e \in \mathcal{O}(v)} e.a,$$

$$\text{net\_bal}(\mathcal{V}) = R_{\text{recv}}(\mathcal{V}) - R_{\text{sent}}(\mathcal{V}), \qquad \overline{d}(\mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} (\tau_{\max}(v) - \tau_{\min}(v)).$$

These aggregates quantify total transaction counts, financial throughput, net fund retention, and mean address lifetime within each wallet.

Composite Ratios and Indices. To normalize across wallets of varying scale and to highlight disproportionate behaviors, we introduce:

$$\text{in\_out\_ratio} = \frac{D_{\text{out}} + \alpha}{D_{\text{in}} + \beta} \qquad \text{volume\_ratio} = \frac{R_{\text{sent}} + \alpha}{R_{\text{recv}} + \beta}$$

$$\text{activity\_idx} = \frac{D_{\text{in}} + D_{\text{out}}}{\overline{d}}$$

where $\alpha, \beta > 0$ are small smoothing constants. An elevated activity_idx signals condensed bursts of transactional activity, while extreme flow ratios flag unbalanced input-output dynamics.

This feature taxonomy not only synthesizes key structural, monetary, and temporal dimensions of on-chain behavior, but also integrates dispersion-based indicators which have emerged in more recent forensic analyses. To contextualize our contributions with respect to the state of the art, Table 1 compares our set of 20 engineered features against those adopted in leading studies on Bitcoin fraud detection and wallet profiling [?,?,?,?,?,?,?]. A checkmark (✓) indicates that a given feature or an equivalent formulation is employed in the cited paper, while a dash denotes its absence. As shown, our framework comprehensively spans the major feature families and incorporates recent heuristics such as transaction dispersion ratios and micro-transaction indicators, which are underrepresented in earlier work.

4 Benchmark Dataset Assembly

This section first outlines three distinct scenarios, obtained using the datasets introduced in Section 2, either combined or standalone. Then it presents two complementary data-quality assessment methods: first, a Confident Learning–based technique for estimating and removing label noise in the Chainabuse dataset (Section 4.1); and second, an unsupervised anomaly-filtering approach driven by an Isolation Forest applied to the most recent sample (Section 4.2).

To assemble our benchmark suite, we first extract on-chain transaction histories for each wallet under investigation. Wallets are grouped via multi-input clustering heuristics applied to the global Bitcoin Unspent Transaction Output (UTXO) graph. From these raw transactions, the full set of address-level and wallet-level features described in Section 3 is computed. Based on this feature matrix, three distinct classification datasets are then constructed.

1. **Chainabuse and recent transactions addresses:** fraudulent labels from the Chainabuse community-reported dataset are paired with unlabeled addresses uniformly sampled from the most recent 10 000 on-chain transactions. This scenario tests robustness against noisy benign samples.
2. **Chainabuse and Elliptic++ licit addresses:** fraudulent addresses from Chainabuse are contrasted against high-quality licit labels drawn from the Elliptic++ academic dataset. This setting simulates a realistic deployment with reliable ground truth on both ends.
3. **Elliptic++**: both fraudulent and legitimate addresses originate from Elliptic++, controlling for source-domain bias by using identical provenance for each class.

Table 1: Mapping of the features published in the literature with respect to the set of 20 variables adopted in this work.

Feature / Famiglia	*Chang 18[?]*	*Toyoda 18[?]*	*Weber 19[?]*	*Nerurkar 21[?]*	*Chen 21[?]*	*Monamo 16[?]*	*Işcan 23[?]*	**Our Paper**
Count / Structural								
In-degree (addr)	✓	✓	✓	✓	✓	✓	–	✓
Out-degree (addr)	✓	✓	✓	✓	✓	✓	–	✓
Unique counterparties (in/out)	–	✓	✓	✓	–	–	–	✓
Local clustering coeff.	✓	–	–	–	–	–	–	✓
Monetary								
Avg. tx amount (in)	–	✓	✓	✓	✓	✓	✓	✓
Avg. tx amount (out)	–	✓	✓	✓	✓	✓	✓	✓
Net balance (address)	–	–	✓	✓	✓	–	–	✓
Temporal								
Mean inter-tx interval (in)	–	✓	✓	✓	✓	✓	–	✓
Mean inter-tx interval (out)	–	✓	✓	✓	✓	✓	–	✓
Address lifetime	–	✓	–	✓	✓	✓	–	✓
Wallet / Aggregate								
$\sum$ in-degree (D_{in})	–	–	–	✓	✓	–	–	✓
$\sum$ out-degree (D_{out})	–	–	–	✓	✓	–	–	✓
Total received (R_{recv})	–	–	–	✓	✓	–	✓	✓
Total sent (R_{sent})	–	–	–	✓	✓	–	✓	✓
Wallet net balance	–	–	–	✓	✓	–	✓	✓
Mean addr duration ($\bar{d}$)	–	–	–	✓	–	–	–	✓
Composite / Ratios								
In/Out ratio (tx count)	–	✓	✓	✓	✓	✓	–	✓
Volume ratio (BTC sent/recv)	–	✓	✓	✓	✓	✓	✓	✓
Activity index (tx / lifetime)	–	✓	–	–	✓	–	✓	✓
Dispersion / Pattern-specific								
Unique-out ratio	✓	–	–	–	–	–	–	✓
Time-interval ratio	–	–	–	–	✓	–	–	✓
Weighted avg tx (micro-tx)	–	–	–	–	✓	–	✓	✓

4.1 ChainAbuse and Potential Label Bias.

Chainabuse is a community-driven registry of cryptocurrency addresses reported for scams, ransomware, phishing and other illicit schemes. Since reports are crowdsourced and not uniformly vetted, the dataset may suffer from two main biases: (i) *reporting bias*, whereby high-profile scams are overrepresented and low-value frauds remain unreported; and (ii) *confirmation bias*, arising if multiple users submit the same address without independent verification. Consequently, Chainabuse's "fraud" labels may include false positives or miss certain illicit addresses.

The reporting bias and confirmation bias inevitably introduce label noise into the ChainAbuse registry—in the form of *false positives* (benign addresses incorrectly flagged) and *false negatives* (illicit addresses that go unreported). To rigorously quantify and mitigate the noise generated by *false positives* and improve downstream model reliability, the Confident Learning framework [**?**], as implemented in the Python module *cleanlab*, is adopted. This approach sys-

tematically identifies examples whose observed labels conflict with a classifier's high-confidence predictions, providing a principled estimate of mislabeling rate.

A combined dataset of 6,910 ChainAbuse "fraud" addresses and 7,378 Elliptic "licit" addresses was used to train a balanced Random Forest (200 trees, max_depth=20) using five-fold, out-of-fold probability estimates. Leveraging the Confident Learning framework (via the `cleanlab` module[7]), we automatically flagged 294 addresses (≈ 2.2 %) whose original labels conflicted with high-confidence predictions. After removing these suspected mislabels and retraining the model, the F1 score on the same held-out test split improved from 0.912 to 0.936 ($\Delta = +0.024$), demonstrating both the presence of label noise and the effectiveness of this methodology.

4.2 Anomaly Filtering in the Recent Sample.

The *Recent Transactions* wallets, drawn uniformly from on-chain activity and lacking explicit labels, may still harbor latent fraudulent behavior. To mitigate the inclusion of potentially malicious wallets in our benign sample, we employ an unsupervised, two-stage anomaly filtering pipeline on the feature representations:

1. **2D PCA Projection.**
   - Compute the covariance matrix
   $$\Sigma = \frac{1}{m-1} X^{\mathsf{T}} X,$$
   where $X \in \mathbb{R}^{m \times d}$ is the feature matrix of the m wallets across d attributes.
   - Solve the eigenproblem $\Sigma\, \mathbf{u}_k = \lambda_k\, \mathbf{u}_k$ and select the top two eigenvectors $\mathbf{u}_1, \mathbf{u}_2$ corresponding to the largest eigenvalues λ_1, λ_2.
   - Project each feature vector x_i into the 2D subspace:
   $$z_i = \left[\mathbf{u}_1^{\mathsf{T}} x_i,\ \mathbf{u}_2^{\mathsf{T}} x_i\right]^{\mathsf{T}},$$
   preserving the cumulative variance fraction $(\lambda_1 + \lambda_2)/\sum_{k=1}^{d} \lambda_k$.
2. **Outlier Removal with Isolation Forest ($\alpha = 0.05$).**
   - Fit an Isolation Forest on the set of projections $\{z_i\}_{i=1}^{m}$, specifying a contamination level $\alpha = 0.05$ to excise the top 5% most extreme points.
   - The anomaly score for a point z is computed as
   $$s(z) = 2^{-\mathbb{E}[h(z)]/c(n)},$$
   where $\mathbb{E}[h(z)]$ is the average path length for z in the forest and
   $$c(n) = 2\,H(n-1) - \frac{2(n-1)}{n}, \quad H(k) \approx \ln k + \gamma$$
   is the normalizing constant for tree size n.

[7] https://github.com/cleanlab/cleanlab

– We adopt $\alpha = 0.05$ to perform robust trimming of heavy tails and extreme observations—not only true frauds, but any anomalous behavior that could distort mean and covariance estimates—in line with classical robust statistical practices [?,?].
– Wallets for which $s(z_i)$ exceeds the threshold defined by α are removed, yielding a purified "Recent" sample free from potential outliers.

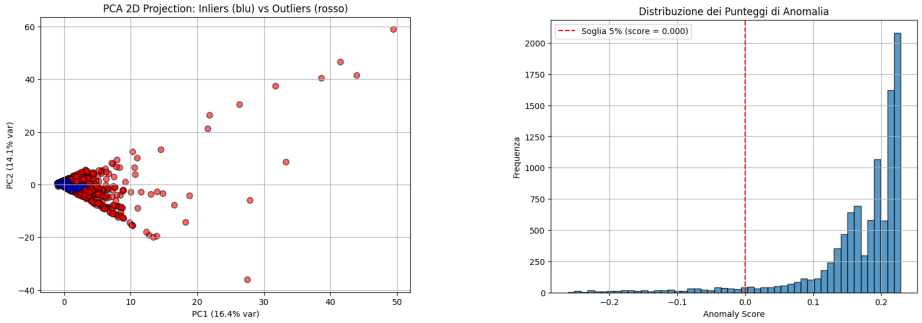


Fig. 1: **Left:** 2D PCA projection of the Recent Transactions wallets. Red points denote outliers identified by the Isolation Forest ($\alpha = 0.05$). **Right:** Distribution of anomaly scores $s(z_i)$; the dashed line marks the 95th percentile threshold.

5 Experiments and Discussion

This section presents the experimental framework used to train and evaluate our classification models, followed by a discussion of the results. In Section 5.1, we describe the preprocessing pipeline, justify our choice of algorithms, outline the hyperparameter optimization strategy, and provide a comparative evaluation of three classification approaches: Random Forest, Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM). In Section 5.2, we interpret the findings, highlight key performance trends, and discuss their implications.

5.1 Experimental Results

To assess the effectiveness of our classification framework, we designed a controlled experimental pipeline encompassing standardized preprocessing, model selection, hyperparameter optimization, and performance evaluation. This section outlines the methodological foundations underpinning our experiments, with an emphasis on reproducibility and comparability across different model families. All experiments were conducted on the three benchmark datasets introduced in Section 2, using the full suite of engineered features described in Section 3.

Preprocessing. To ensure consistency and reproducibility, all datasets underwent an identical preprocessing pipeline. First, the feature matrix was separated from the target class labels. A type-inference stage distinguished between numerical and categorical variables. Numerical attributes were subjected to median imputation followed by z-score normalization to reduce scale-related biases and the impact of outliers. Categorical features, where present, were imputed with a constant placeholder and transformed through one-hot encoding, allowing for compatibility with downstream models.

The resulting processed dataset was encoded entirely in numerical form and subsequently partitioned into training and test sets using a stratified 80:20 split to preserve class balance.

Random Forest Classifier. We trained a Random Forest classifier with class weights set to 'balanced' to compensate for minor class imbalances. A grid search was conducted over a range of hyperparameters using 5-fold cross-validation with accuracy as the optimization criterion. The optimal configuration consisted of 200 estimators, a maximum depth of 20, and the square root of the number of features as the maximum feature subset size.

The results are summarized in Table 2. Notably, in Scenario 2—where Chainabuse-provided fraudulent labels are paired with licit samples from Elliptic++—the Random Forest achieved an F1-score of 0.92 for both classes, indicating excellent discriminative performance.

Table 2: Random Forest Classification Results

Scenario	Class	Precision	Recall	F1-score
1	Fraud	0.78	0.79	0.79
	Licit	0.85	0.84	0.85
2	Fraud	0.93	0.91	0.92
	Licit	0.92	0.93	0.93
3	Fraud	0.92	0.92	0.92
	Licit	0.87	0.87	0.87

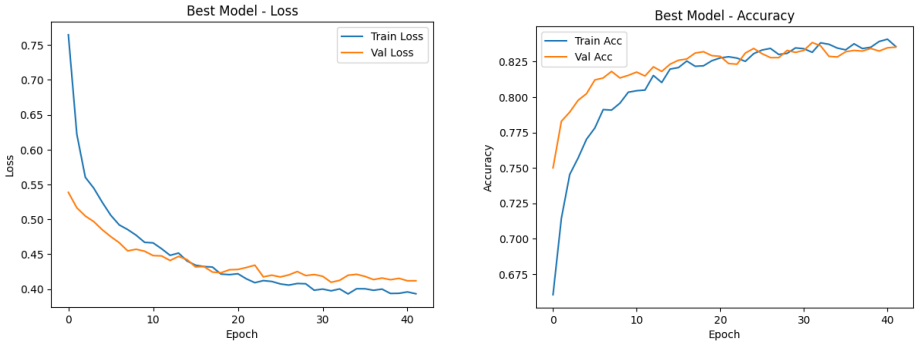
Feed-Forward Neural Network. We implemented a fully–connected *multilayer perceptron* and performed *hyperparameter tuning* using a random search strategy. The search space and the best configuration discovered are reported in Table 3. Unless stated otherwise, all layers use the ReLU activation.

Figure 2 shows that the validation curves closely follow the training ones: the loss stabilises around 0.40, while accuracy converges near 0.85 without any noticeable divergence, indicating good generalisation.

Overall, the model attains an **accuracy of 0.85** on the test split, with balanced F1-scores across the fraud and licit classes (Table 4). These results confirm the effectiveness of the selected architecture and regularisation strategy for the task.

Table 3: Hyper-parameter search space explored.

Hyper-parameter	**Search space**	**Best value**
Number of hidden layers	$n_{\text{layers}} \in \{2, 3\}$	3
Units per layer	$u_i \in \{32, 64, 96, 128\}$	[96, 96, 32]
Dropout rate	$p_i \in \{0.2, 0.3, 0.4\}$	0.30
Learning rate (Adam)	$\alpha \in \{10^{-2}, 10^{-3}, 5 \times 10^{-4}\}$	10^{-3}
Batch size	$B \in \{32, 64, 128\}$	64
L2 weight decay	λ (fixed)	10^{-4}



(a) Training and validation loss.

(b) Training and validation accuracy.

Fig. 2: Learning curves of the optimised MLP over 40 epochs.

Support Vector Machine. To further validate the generality of our feature set, we trained a Support Vector Machine (SVM) classifier. The model was embedded in the same preprocessing pipeline adopted for the other learners, and class imbalance was mitigated via the `balanced` option. Hyperparameters were tuned through a 5-fold grid search over $C \in \{0.1, 1, 10\}$, `kernel`$\in \{\text{linear}, \text{rbf}\}$, and `gamma`$\in \{\text{scale}, \text{auto}\}$. The optimal configuration—shared across scenarios—was obtained with $C = 10$, an RBF kernel, and `gamma=scale`.

The SVM attains performance comparable to the MLP, particularly in Scenario 2 where both precision and recall exceed 0.80 for each class. Nevertheless, its sensitivity to noisy or weakly-labeled data (Scenario 1) remains evident, reinforcing the advantages of ensemble models in heterogeneous operational settings.

5.2 Discussion.

Our experiments demonstrate that ensemble models, particularly Random Forests, offer a favorable performance. Scenario 2 yielded the best results, confirming the value of combining crowd-sourced and expert-labeled datasets. In contrast, the neural network showed reduced resilience to label noise and distributional shifts.

To benchmark our methodology, we compared classification accuracy with previous results in Table 6. Our Random Forest model matches or surpasses

Table 4: MLP — Classification Results

Scenario	Class	Precision	Recall	F1-score
1	Fraud	0.66	0.78	0.71
	Licit	0.82	0.71	0.76
2	Fraud	0.89	0.78	0.83
	Licit	0.82	0.91	0.87
3	Fraud	0.81	0.85	0.83
	Licit	0.73	0.67	0.70

Table 5: Support Vector Machine — Classification Results

Scenario	Class	Precision	Recall	F1-score
1	Fraud	0.65	0.76	0.70
	Licit	0.80	0.71	0.76
2	Fraud	0.83	0.80	0.81
	Licit	0.83	0.85	0.84
3	Fraud	0.82	0.87	0.85
	Licit	0.77	0.69	0.73

state-of-the-art approaches, validating the effectiveness of our feature set and classification pipeline.

Table 6: Accuracy reported in related works compared to our best-performing model (Random Forest, Scenario 2).

Authors (Year)	Model	Dataset	Accuracy
Chang et al. (2018)[?]	Random Forest / SVM	Bitcoin tx patterns	89%
Toyoda et al. (2018)[?]	k-NN / Decision Tree	Blockchain service wallets	85%
Nerurkar et al. (2021)[?]	CNN (Deep Learning)	Transaction flows	91%
Işcan et al. (2023)[?]	LightGBM	Wallet-based behaviors	93%
Our Results (2025)	Random Forest	Chainabuse + Elliptic++	**92%***

*Inferred from F1-scores of 0.92 for both fraud and licit classes.

Explainability via SHAP. To interpret the predictions of our Random Forest classifier, we compute SHAP values [?] on the binary fraud/licit model (fraud samples from Chainabuse; licit from Elliptic++). SHAP values provide both global feature importance and the direction in which each feature drives the model's output toward "fraud."

Figure 3 shows the SHAP summary plot: horizontal spread indicates each feature's contribution to output variance, and color denotes effect direction. Several key patterns emerge:

– **Dispersion-oriented features dominate.** Both `unique_out_ratio` and `total_unique_out` appear at the top of the ranking, underscoring that wallets which fan out funds to a large number of distinct counterparts are

markedly more likely to be classified as fraudulent. This accords with classical forensic typologies—such as peel chains and mixer operations—where rapid splitting hinders traceability.

- **Temporal compactness signals illicit behaviour.** Features that capture *burstiness* (`time_interval_ratio`, `avg_out_time_interval`) and *ephemeral lifetimes* (`wallet_lifetime_sec`) exert a strong positive influence on the fraud score, reflecting the operational practice of using short-lived "burner" addresses for one-off laundering rounds.
- **Micro-transactions and rapid depletion are red flags.** Low-value transfers (`weighted_avg_tx`, `avg_out_transaction`) coupled with near-zero `net_balance` drive the model towards the fraud class, consistent with "smurfing" or "dusting" strategies designed to fragment illicit proceeds below exchange reporting thresholds.

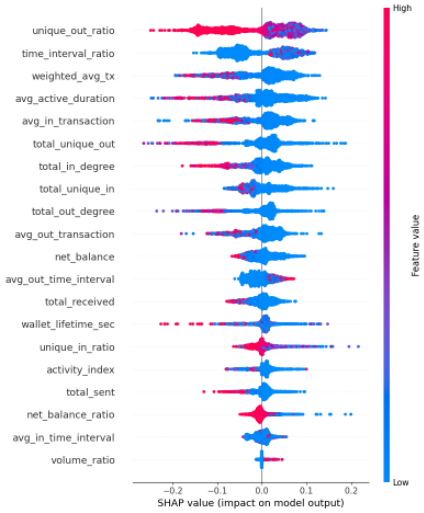


Fig. 3: SHAP summary plot: points show feature SHAP values; features are ordered by mean absolute impact.

Table 7 distils these insights, reporting for each feature its tier ranking, the polarity of its dominant contribution, and a succinct forensic interpretation.

The SHAP analysis shows that rapid-dispersion (e.g. `unique_out_ratio`, `time_interval_ratio`) and micro-slicing (e.g. `weighted_avg_tx`, `avg_out_transaction`) features most strongly indicate fraud, offering concrete compliance and forensic insights.

Ablation Study. Building upon the optimal Random Forest configuration described, we conducted a comprehensive ablation analysis by retraining the model omitting each individual feature. The baseline macro-averaged F1 score with the

Table 7: Global Feature Importance and Direction via SHAP

Rank	Feature	↑ Fraud	Rationale
1	unique_out_ratio	high (↑)	Many distinct recipients – typical of fund dispersion in scams or mixers.
2	time_interval_ratio	high (↑)	Denser outflows than inflows – rapid unloading of funds.
3	weighted_avg_tx	low (↓)	Numerous micro-transactions suggest smurfing or dusting.
4	avg_active_duration	low (↓)	Short-lived wallets often serve as "burner" addresses.
5	avg_in_transaction	low (↓)	Small inbound amounts frequently precede fraudulent aggregation.
6	total_unique_out	high (↑)	Large number of unique outgoing addresses reinforces dispersion.
7	total_in_degree	low (↓)	Few incoming transactions relative to outflows.
8	total_unique_in	low (↓)	Concentrated inbound sources indicate planned layering.
9	total_out_degree	high (↑)	High total count of outgoing transactions.
10	avg_out_transaction	low (↓)	Many small outgoing slices – indicative of splitting funds.
11	net_balance	low (↓)	Near-zero or negative balance from rapid depletion.
12	avg_out_time_interval	low (↓)	Tight timing between successive outflows.
13	total_received	low (↓)	Few receipts but many outgoing payments is anomalous.
14	wallet_lifetime_sec	low (↓)	Short operational lifetime increases suspicion.
15	unique_in_ratio	low (↓)	Low diversity of inbound sources – concentrated deposits.
16	activity_index	high (↑)	Very high daily transaction rate – hyperactivity.
17	total_sent	high (↑)	Large absolute outgoing volume.
18	net_balance_ratio	low (↓)	Re-sends nearly all received funds.
19	avg_in_time_interval	low (↓)	Closely spaced inbound deposits.
20	volume_ratio	high (↑)	Sends much more than receives (out/in elevated).

Table 8: Single-feature ablation – top 10 ΔF1 (top 3 in bold)

Feature removed	$\mathbf{F1}_{\text{macro}}$	$\boldsymbol{\Delta}$F1
weighted_avg_tx	0.9167	**−0.0088**
wallet_lifetime_sec	0.9186	**−0.0069**
avg_out_transaction	0.9192	**−0.0063**
total_unique_in	0.9193	−0.0062
avg_active_duration	0.9196	−0.0059
avg_in_time_interval	0.9196	−0.0059
total_sent	0.9203	−0.0052
total_in_degree	0.9204	−0.0051
unique_out_ratio	0.9204	−0.0051
total_out_degree	0.9211	−0.0044

full feature set is **0.9255**. Table 8 reports the ten individual features whose removal produces the largest declines in macro-averaged F1; ΔF1 denotes the absolute drop relative to the baseline.

These results corroborate the SHAP-based interpretability analysis: omitting `weighted_avg_tx`, which captures micro-transaction patterns, causes the single largest drop (0.88 pp); elimination of `wallet_lifetime_sec` and `avg_out_transaction` similarly degrades performance by over 0.6 pp, underscoring the importance of bursty temporal behavior.

Scalability and Deployment. While the present study concentrates on the *offline* evaluation of our wallet-classification framework, operating at blockchain scale and low-latency processing introduces further engineering challenges. In an extended version of this work we plan to: (i) design a streaming architec-

ture capable of ingesting live Bitcoin transactions; (ii) demonstrate incremental maintenance of the graph–temporal features introduced in Section 3; and (iii) empirically assess latency, throughput, and cost trade-offs under realistic workloads. Preliminary sketches suggest that combining lightweight stream processing for rapid event filtering with scheduled micro-batch jobs for feature aggregation could offer an effective balance between responsiveness and resource utilisation, but a rigorous validation is left for future work.

6 Conclusions and Future Work

In this study, we have developed and evaluated a comprehensive framework for classifying Bitcoin wallets as illicit or legitimate. By integrating three disparate data sources—Elliptic++, ChainAbuse and a uniformly sampled set of recent on-chain wallets—we produced a balanced corpus of over 31 000 labeled entities, resolving conflicts in favor of the fraud label to maximize recall (cf. [**?**]). Our feature engineering pipeline extracted twenty metrics spanning structural counts, monetary aggregates, temporal patterns and composite ratios; these draw upon and extend prior heuristics [**?**,**?**] while incorporating dispersion indicators recently highlighted in live-forensics research [**?**]. In comparative experiments, Random Forest consistently achieved F1-scores above 0.92 under high-quality labeling, whereas LightGBM was notably effective at halving the false-alarm rate in an industrial setting [**?**]. Further, SHAP analyses elucidated that unique-out-ratio, time-interval-ratio and micro-transaction averages are the strongest predictors of fraud.

Nonetheless, crowdsourced labels and anomaly filtering may still misclassify borderline cases, and concentrating solely on on-chain metrics can overlook crucial off-chain evidence (e.g., exchange KYC).The UTXO-centric feature set requires adaptation for account-based or privacy-focused blockchains; and fourth, our evaluation on static snapshots must give way to a streaming analytics architecture for real-time risk scoring.

Moving forward, we will leverage self-supervised and contrastive learning to exploit large volumes of unlabeled transactions for richer graph embeddings [**?**], integrate heterogeneous GNNs (e.g., GAT, HGT) to model complex laundering networks beyond pairwise links [**?**], and extend our pipeline to additional chains (e.g., Ethereum, BSC) while fusing on-chain analytics with off-chain data such as exchange records and darknet market reports.

References

1. Agarwal, U., Rishiwal, V., Tanwar, S., Yadav, M.: Blockchain and crypto forensics: Investigating crypto frauds. International Journal of Network Management **34**(2), e2255 (2024)
2. Van den Broeck, G., Lykov, A., Schleich, M., Suciu, D.: On the tractability of shap explanations. Journal of Artificial Intelligence Research **74**, 851–886 (2022)

3. Chang, T.H., Svetinovic, D.: Improving bitcoin ownership identification using transaction patterns analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems **50**(1), 9–20 (2018)
4. Chen, B., Wei, F., Gu, C.: Bitcoin theft detection based on supervised machine learning algorithms. Security and Communication Networks **2021**(1), 6643763 (2021)
5. Elmougy, Y., Liu, L.: 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 (2023)
6. Iscan, C., Kumas, O., Akbulut, F.P., Akbulut, A.: Wallet-based transaction fraud prevention through lightgbm with the focus on minimizing false alarms. IEEE Access **11**, 131465–131474 (2023)
7. Lin, C., Liao, H., Tsai, F.: A systematic review of detecting illicit bitcoin transactions. In: Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022). Procedia Computer Science, vol. 207, pp. 3211–3219. Elsevier (2022)
8. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
9. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **6**(1), 1–39 (2012)
10. Monamo, P., Marivate, V., Twala, B.: Unsupervised learning for robust bitcoin fraud detection. In: 2016 Information Security for South Africa (ISSA). pp. 129–134. IEEE (2016)
11. Nerurkar, P., Bhirud, S., Patel, D., Ludinard, R., Busnel, Y., Kumari, S.: Supervised learning model for identifying illegal activities in bitcoin. Applied Intelligence **51**, 3824–3843 (2021)
12. Northcutt, C., Jiang, L., Chuang, I.: Confident learning: Estimating uncertainty in dataset labels. Journal of Artificial Intelligence Research **70**, 1373–1411 (2021)
13. Toyoda, K., Ohtsuki, T., Mathiopoulos, P.T.: Multi-class bitcoin-enabled service identification based on transaction history summarization. In: 2018 IEEE Int. Conf. Internet of Things (iThings), GreenCom, CPSCom & SmartData. pp. 1153–1160. IEEE (2018)
14. Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591 (2019)