Towards Explainable Detection of Money Laundering in Transaction Networks

Mario Trerotola[1][0009−0000−7052−4588], Davide Calvaresi[2][0000−0001−9816−7439], and Mimmo Parente[3][0000−0002−4935−6003]

[1] Department of Control and Computer Engineering (DAUIN), Politecnico di Torino, 10129 Torino, Italy
mario.trerotola@polito.it
[2] Haute École d'Ingénierie (HEI), School of Engineering, HES-SO Valais-Wallis, CH-1950 Sion, Switzerland
davide.calvaresi@hevs.ch
[3] Dipartimento di Scienze Aziendali, Management & Innovation Systems, Università degli Studi di Salerno, 84084 Fisciano, Italy
parente@unisa.it

Abstract. An estimated 2–5% of global GDP is laundered every year, and transparency and human oversight are increasingly expected in regulated Anti-Money Laundering (AML) settings. Detection is almost always reported through aggregate metrics, yet these can hide the failures that matter most in practice. This study makes three contributions. First, we evaluate detection recall separately for each laundering typology, rather than only in aggregate. Testing four models (Random Forest, LightGBM, XGBoost, and TabNet) on a synthetic AMLSim network, we find that fan-out is systematically missed, around 30 percentage points below the other typologies and consistently across all four models: the gap is a persistent limitation of the evaluated account-level representation, not of any single model. Second, account-level feature engineering does not close this gap: after retuning the baseline at a matched budget, our targeted features leave fan-out recall essentially unchanged, pointing to a structural limitation of the representation rather than missing features; the gap also survives multiple sampling strategies and a pattern-disjoint split. Third, we apply three explainability methods (TreeSHAP, DiCE, and TabNet attention) together with a feature-ablation sensitivity check. Finally, replicating the pipeline on the more realistic synthetic AMLworld HI-Medium benchmark, we confirm that the gap is not specific to AMLSim: fan-out recall remains low (0.62 vs. 0.87 for cycle). On the structured subset, PR-AUC is 0.82 under a random split and 0.58 under a temporal split.

Keywords: Anti-money laundering · Explainable AI · Transaction networks · TabNet · SHAP · Money laundering typologies

1 Introduction

Money laundering is estimated to account for 2–5% of global GDP annually. Financial institutions are required to detect and report suspicious activity, yet the scale of modern transaction networks makes manual surveillance impractical. Machine learning methods have been increasingly applied to Anti-Money Laundering (AML) detection, spanning supervised tabular classification on real banking data and graph-based approaches on cryptocurrency transaction networks. Synthetic benchmarks have enabled controlled experimentation by embedding known laundering patterns within realistic transaction flows: the AML Simulator (AMLSim) labels each transaction with its laundering typology, whereas IBM synthetic AML transaction datasets (AMLworld) model the full standard three-stage laundering process for greater realism. The two are complementary and we use both: AMLSim as the primary dataset (its per-transaction typology labels are a prerequisite for our per-typology analysis) and AMLworld HI-Medium as a second, more realistic synthetic benchmark for replication. Detection performance is almost exclusively reported through aggregate metrics such as AUC and F1, without distinguishing among structurally different laundering typologies. A classifier achieving 0.95 F1 overall may perform well on topologically distinctive patterns while underperforming on patterns whose structural signatures are less separable from legitimate activity, as we show for the fan-out typology. Moreover several recent studies have used AMLSim typologies as classification features or evaluated detection at the subgraph level, but supervised account-level performance is rarely reported per individual typology. Second, although transparency and human oversight are increasingly emphasized for automated decision systems in regulated financial settings (e.g., under the EU AI Act), most AML detection studies either omit explainability entirely or employ a single post-hoc method.

In this scenario, we identify three points that can be explored to give new insights.

C1: A reproducible fan-out feature detection gap. High aggregate scores do not reveal an around 30 pp gap in fan-out recall measure; we attribute this gap to the account-level representation rather than to any single model, as it is consistent across four architectures, stable under different sampling strategies, and present under a pattern-disjoint split.

C2: Detection gains from targeted features. Channel and currency features yield the largest improvement among the interventions we evaluate on AMLworld (+8.0 pp PR-AUC); on AMLSim, by contrast, extending the account-level feature set does not close the fan-out gap, indicating a limitation of the representation rather than missing features.

C3: Multi-method explainability. Across TreeSHAP, DiCE, and TabNet attention, the methods show partial agreement at the feature-family level but weak agreement at the individual-feature level. An ablation confirms that a few features drive detection: masking the top three drops PR-AUC from 0.96 to 0.50.

We use AMLSim and AMLworld since the typology labels (not publicly available for real banking data) are a prerequisite for C1. The high aggregate performance (AUC > 0.99) reflects the regularity of synthetic patterns and should be contextualized accordingly. Later in the paper we quantify this by re-running the pipeline on AMLworld HI-Medium.

2 Related Work

Among the first publicly available labeled benchmarks, the Elliptic dataset [31] provided over 200K labeled Bitcoin transactions, where Random Forest outperformed the baseline GCN; later graph architectures closed and reversed this gap [20,9]. Elmougy and Liu [11] extended the benchmark to the actor level with Elliptic++, Bellei et al. [5] reframed AML as subgraph classification in Elliptic2, showing that laundering activity exhibits characteristic structural shapes, and Johannessen and Jullum [16] applied heterogeneous GNNs to a 5-million-node banking graph.

Self- and semi-supervised strategies. Label scarcity is a central challenge in AML: rule-based systems in production generate false-positive rates estimated above 95%, making manual annotation prohibitively expensive [7]. Self- and semi-supervised graph learning addresses this, from Inspection-L [20] and LaundroGraph [7] to contrastive pre-training in GCPAL [21] and scalable semi-supervised pipelines [18]. These directions are orthogonal to ours: we work in a fully supervised setting and ask what supervised detection misses.

Synthetic data and typological analysis. IBM AMLSim [28] embeds configurable laundering patterns within realistic transaction flows; Altman et al. [1] extended this with AMLworld, modeling the full three-stage laundering process. On AMLSim, Meng and Li [23] proposed an unsupervised structure detection framework with recall between 0.94 and 0.95 at the subgraph level. Phyu and Uttama [25] examined typological features as input, finding that Decision Tree reached 87.6% accuracy when all typology flags were included. Phyu and Uttama [26] combined SMOTE with Random Forest, achieving over 99% accuracy with typological augmentation but without disaggregating performance by pattern. Anaroch et al. [2] compared traditional, deep learning, and hybrid models on the highly imbalanced AMLworld HI-Small benchmark, with XGBoost achieving 76% recall under random undersampling.

To our knowledge, none of these studies disaggregates supervised account-level classification performance across individual patterns (C1), and none combines multiple explainability techniques (C3).

Table 1 compares prior results and their configurations. Results on synthetic AML benchmarks are sensitive to network size, fraud prevalence, typology mix, amount distributions, simulated timesteps, feature construction, and evaluation level. Since the prior studies considered here do not report all configuration details needed to reconstruct an identical experimental setting, their results are not directly comparable to ours. Nevertheless, the comparison highlights an important pattern: very high aggregate accuracy is typically reported on smaller graphs

Table 1. Configuration-aware comparison with prior work on synthetic AML benchmarks. "R" denotes recall; "w-F1" denotes weighted F1; "n/r" = not reported.

Study	Experimental setting	Method	Reported performance
Meng & Li 2023 [23]	AMLSim structure detection; mixed laundering structures; ∼10 k accounts	SIE	R = 0.947; F1=0.882; Acc.=0.983
Phyu & Uttama 2023 [25]	AMLSim (IBM); 1,176 transactions; typologies used as features	DT	Acc.=0.876; R/F1/AUC n/r
Phyu & Uttama 2024 [26]	AMLSim (IBM); 1,048,575 transactions; 585 laundering cases	RF + SMOTE	$R >$ 0.64; Acc.$>$ 0.91; F1/AUC n/r
Anaroch et al. 2025 [2]	AMLworld HI-Small; 5.1 M transactions; highly imbalanced transaction-level test set	XGBoost	R = 0.772; AUC=0.900; w-F1=0.958
This work	**AMLSim account-level detection; 433 k accounts; 1.38% fraudulent accounts**	**LightGBM + TabNet**	R = 0.931**; F1=0.953; AUC=0.994**

or settings with higher fraud prevalence [26,25], while larger and more imbalanced configurations are more challenging. We report recall whenever available because, in AML detection, false negatives correspond to undetected laundering activity and are therefore operationally critical. The closest setting in scale and imbalance is Anaroch et al. [2], albeit on AMLworld HI-Small rather than AMLSim; they report aggregate AUC but no per-typology evaluation.

3 Dataset and Laundering Typologies

The detection framework is evaluated on a synthetic transaction network generated with IBM AMLSim [28], a multi-agent simulator that models the layering phase of money laundering by injecting configurable illicit patterns within legitimate transactions. AMLSim is used because it is among the few publicly available data sources that label each transaction with its laundering typology, a prerequisite for C1.

The simulation populates 600,000 bank accounts and runs for 500 time steps, corresponding to approximately 16 months of banking activity (seed = 42).[1] Legitimate transactions range from $100 to $1,000, while illicit transfers are kept between $100 and $200. At each intermediate hop a 10% margin is retained by the *money mule*, an individual or account that transfers illicit funds on behalf of others, knowingly or not, typically to obfuscate their origin. Eight hundred distinct criminal networks are injected, distributed uniformly across four laundering typologies (200 patterns each), with each network involving 5 to 10 accomplice accounts.

[1] We used the publicly available AMLSim release from [28]; only the `TRANSFER` transaction type was enabled.

The resulting dataset comprises 15,141,450 transactions among 433,455 active accounts. Of these, 5,980 accounts (1.38%) participate in laundering activity, producing 6,217 illicit transactions (0.041% of the total). The data is represented as a directed, weighted, temporal multigraph $G = (V, E)$: each node $v \in V$ corresponds to a bank account, and each edge $e = (v_i, v_j, a, t) \in E$ represents a transfer from v_i to v_j of amount a executed at time t, where t is the transaction timestamp. The two datasets differ in temporal resolution: AMLSim records transaction timestamps at second-level granularity, whereas AMLworld HI-Medium (Section 5.7) provides them at minute-level granularity.

The first four are present in both datasets (Figure 1, top row):

Fan-in. Multiple accounts send micro-payments to a single destination aggregation account.

Fan-out. A single source account disperses funds across many recipients, resembling legitimate distribution (e.g., payroll).

Cycle: Funds circulate through a closed loop of k intermediary accounts, with each hop reducing the amount by the margin ratio.

Scatter-gather. A two-phase pattern: funds are first dispersed (fan-out), then re-aggregated (fan-in) into a different destination.

The remaining three are specific to AMLworld HI-Medium (Figure 1, bottom row; Section 5.7):

Random. An arbitrary directed subgraph with no fixed topology: the involved accounts are linked by an irregular set of transfers that matches no canonical template.

Bipartite. Funds flow many-to-many from a set of source accounts to a disjoint set of destinations, structurally close to legitimate marketplace- or payroll-like settlement between two account populations.

Stacked bipartite. A layered extension of the bipartite pattern that inserts an intermediary tier (source $\rightarrow$ intermediary $\rightarrow$ destination), so that funds traverse two consecutive many-to-many hops.

The four typologies are balanced in terms of criminal networks (200 each) and account involvement (1,453–1,529 accounts per type, $\approx$7.5 per pattern). Scatter-gather produces the most illicit transactions (2,197) due to its two-phase structure, followed by cycle (1,494), fan-in (1,273), and fan-out (1,253).

4 Account-Level Feature Extraction

From the transaction graph defined in Section 3, we extract 50 account-level features organized in six categories. The first four (37 features) combine indicators standard in prior AML work; the last two (13 features) are novel contributions (C2): targeted features for fan-out dispersal and for the neighborhood context. All features are computed per account over the entire simulation window, with a smoothing constant $\alpha = \beta = 10^{-6}$ in all ratio-based features.[2]

[2] Our feature set shares foundational concepts with the graph-temporal descriptor proposed by Trerotola et al. [29] for crypto-asset scam detection, but differs in scope:

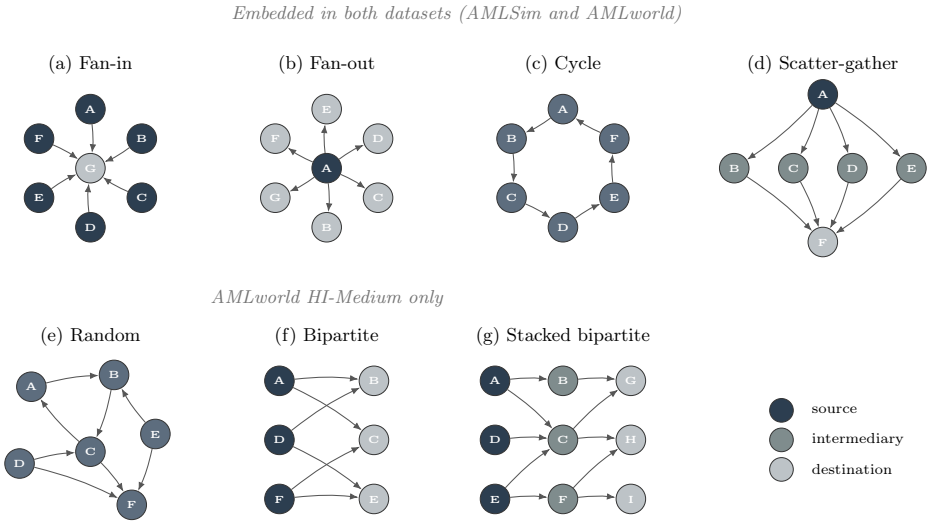

Fig. 1. The seven laundering typologies as directed subgraphs (node shade: source dark, intermediary medium, destination light). Top row: the four embedded in our AMLSim configuration (edge weight decays by 10% per hop); bottom row: the additional typologies in AMLworld HI-Medium (Section 5.7).

4.1 Base Features

The 37 base features are organized into four groups: *Base transaction aggregates* (10) capture in- and out-degree, transaction amount statistics (mean, std), total volume, and unique counterparty counts [31,20]. *Temporal patterns* (10) encode inter-transaction intervals (mean, std, separately for in/out), Shannon entropy of the hourly distribution, and ratios for off-hours (22:00–06:00) and weekend activity [17,15]. *Graph-structural measures* (2) consist of the local clustering coefficient [30] and ego size. *Composite ratios* (15) combine raw indicators into AML-relevant signals: net balance, degree and volume ratios, unique-out and unique-in ratios, temporal interval ratio, activity index, burstiness coefficients [4], and normalized amount ranges.

4.2 Novel Features for Fan-Out and Neighborhood Context

These features address two challenges: separating fan-out laundering from legitimate distribution (features 38–44), and capturing neighborhood context without ground-truth labels (features 45–50).

their work operates at the token level on small graphs, whereas the present work operates at the account level within a 433K-node network where global centrality computation is prohibitive.

Table 2. Summary of the 50 account-level features.

Category	Count	Novel	Representative features
Base aggregates	10	–	degree, amounts, unique counterparties
Temporal patterns	10	–	burstiness, hourly entropy, off-hours ratio
Graph-structural	2	–	clustering coefficient, ego size
Composite ratios	15	–	unique-out ratio, volume ratio, activity index
Fan-out targeted	7	C2	Gini coeff., fresh recipient, relay ratio
Neighborhood	6	C2	nbr degree mean/std, degree-vs-nbr
Total	**50**	**13**	

Fan-out targeted features (7). *Gini coefficients* of outgoing and incoming amounts measure distribution inequality; we expect low Gini values to be more characteristic of automated dispersal than of legitimate payments. The *fresh recipient ratio* captures the fraction of first-time recipients. *Dispersion speed* measures how quickly new recipients are reached. *Outgoing transaction frequency* captures outbound intensity. The *relay ratio* measures the fraction of recipients forwarding funds within ± 1 hour, detecting pass-through behavior. The *round amount ratio* captures multiples of 50 or 100. These features are designed to capture behavioral differences that we expect to distinguish illicit from legitimate dispersal: laundering fan-out is more likely to involve uniform amounts, fresh counterparties, high-speed activity, and downstream relay, whereas legitimate distribution typically involves heterogeneous amounts, stable recipients, and no immediate forwarding.

Neighborhood-context features (6). These capture the 1-hop neighborhood without ground-truth labels: mean and standard deviation of neighbors' degree, mean lifetime and clustering coefficient of neighbors, and two relative measures (*degree-vs-nbr*, *lifetime-vs-nbr*). They are computable in $O(d)$ per node, making them feasible on large networks. We hypothesize that accounts involved in laundering patterns tend to be surrounded by others with similarly anomalous activity profiles; these features encode this form of guilt by association without label leakage, and the hypothesis is tested in the multi-method explainability analysis and the feature-ablation study of Section 5.6 and Section 5.3. While neighborhood aggregation is a standard primitive in graph representation learning [20,9], its application as an explicit, hand-crafted tabular feature set remains uncommon in the AML literature. This design choice is motivated by two considerations: (i) the resulting features retain direct semantic interpretability, which supports the transparency and human-oversight documentation relevant to regulated AML settings (e.g., the EU AI Act [12]); and (ii) 1-hop statistics are computable in $O(d)$ per node, avoiding the memory and training costs of full GNN pipelines on 400K+ node networks. Table 2 summarizes all 50 features by category.

5 Results

This section first reports aggregate and per-typology detection performance across four model architectures (Sections 5.1–5.2), then applies three complementary explainability methods across two model families to investigate the sources of the observed detection gap (Section 5.6), and finally tests whether the findings generalize to the more realistic synthetic AMLworld HI-Medium benchmark (Section 5.7).

5.1 Experimental Setup

The AMLSim dataset is split into training (72%), validation (8%), and test (20%) sets using stratified sampling. The validation set is used for threshold optimization and for Platt scaling of TabNet outputs. Platt scaling is applied to TabNet only: the tree ensembles are already well-calibrated under this training regime, whereas TabNet's sparsemax outputs are systematically over-confident, so calibration is used only where probabilities are interpreted (the TabNet counterfactual and attention analyses). Given the extreme imbalance (1.38% fraud), SMOTE [8] is applied on the training set ($k = 5$), producing 615,562 balanced samples. No oversampling is applied to the validation or test sets. During the Optuna search, SMOTE is applied inside each cross-validation fold as an `imblearn` pipeline step, so no synthetic samples enter the held-out fold; the final model is refit on the SMOTE-balanced full training set. Section 5.5 additionally re-evaluates the same pipeline under a pattern-disjoint split as a robustness check on the neighborhood-context features of Section 4.2.

Four models are trained: Random Forest, LightGBM, XGBoost, and TabNet [3]. Random Forest, LightGBM, and XGBoost are optimized using Optuna (30 trials, 5-fold stratified CV, F1 objective); the selected hyperparameters are reported in the footnote.[3][4]

We adopt a two-model design with explicit roles. *LightGBM* is the primary detector: highest aggregate F1 (Table 3), with exact TreeSHAP attribution [22]. *TabNet* is retained as an interpretability companion: its sequential attention masks expose the *order* in which features are consulted, which no post-hoc method on a tree model can natively capture. Tree-based ensembles remain state-of-the-art on medium-sized tabular data [13]; because the two models learn in very different ways, agreement between their explanations is unlikely to be model-specific (Section 5.6).

We use PR-AUC as the primary ranking metric and recall at a fixed 1% false-positive rate as the operationally meaningful operating point; ROC-AUC,

[3] RF: 300 trees, depth 30. LightGBM: 550 trees, depth 13, 82 leaves, lr 0.145, subsample 0.723, colsample 0.909. XGBoost: 200 trees, depth 14, lr 0.084, subsample 0.857, colsample 0.682. CV F1: 0.946 / 0.955 / 0.951.

[4] TabNet uses n_d=n_a=64, 5 sparsemax steps, γ=1.5, λ_{sparse}=10^{-3}, lr = 0.02, batch 4,096; 50 unsupervised pre-training epochs (masking 0.8) plus up to 150 supervised epochs with CosineAnnealingWarmRestarts, early stopping on val. AUC, then Platt scaling.

Table 3. Aggregate classification performance on the test set (fraud class). Best values in bold.

Model	PR-AUC	ROC-AUC	F1	Precision	Recall	Thr
Random Forest	0.961	0.9951	0.9455	0.955	**0.936**	0.450
LightGBM	**0.963**	0.9944	**0.9529**	**0.976**	0.931	0.233
XGBoost	0.961	**0.9949**	0.9527	0.972	0.934	0.521
TabNet	0.947	0.9917	0.9314	0.956	0.908	0.643

Table 4. Per-typology recall on the test set. With $n \approx 280$–330 illicit accounts per typology, 95% Wilson half-widths range from $\approx \pm 0.01$ (recall $\geq$0.99) to $\approx \pm 0.05$ (fan-out); within-typology differences between models are not statistically significant.

Model	Cycle	Fan-in	Fan-out	Scatter-gather
Random Forest	0.993	**0.991**	**0.761**	0.996
LightGBM	**0.997**	0.976	0.751	0.996
XGBoost	0.993	0.988	0.754	0.996
TabNet	0.990	0.963	0.693	0.982

F1, precision, and recall at F1-optimized thresholds (grid search on the validation set) are reported as supplementary. Per-typology metrics are computed by filtering to accounts belonging to each pattern; each fraudulent account carries a single typology label, so the per-typology recalls partition the fraud population with no double-counting.

Table 3 reports test-set performance (86,691 accounts, 1,196 fraudulent). All models achieve ROC-AUC above 0.991 and F1 above 0.93, with LightGBM reaching the highest F1 (0.953) and TabNet the lowest (0.931). PR-AUC is uniformly high (0.95–0.96), and at a 1% false-positive rate recall is 0.94–0.96 across models.

TabNet's optimized threshold (0.643) is substantially higher than those of the tree-based models: its probability outputs are shifted upward, making raw scores harder to read as confidence estimates. This observation motivates the multi-perspective explainability analysis in Section 5.6.

5.2 Per-Typology Detection Gap

Table 4 disaggregates the recall measure across the four typologies, revealing a detection disparity, invisible to previous aggregate metrics.

Cycle and scatter-gather are detected with recall above 0.98 by all models. Fan-out is the hardest pattern, with recall dropping to 0.693–0.761: between one in four and one in three dispersal-based schemes remain undetected. This is consistent with the measured overlap between illicit fan-out dispersal and legitimate distribution (quantified below). The recall gap of around 30 percentage points is consistent across all four architectures, pointing to a persistent limitation of the evaluated account-level representation rather than a model-specific weakness. Section 5.3 quantifies the contribution of the C2 features through a 37-vs-50 ablation, and Section 5.4 verifies that this gap is robust across the resampling strategies.

Table 5. Missed fan-out dispersers vs. legitimate high out-degree accounts: per-feature median and Cohen's $|d|$. The medians nearly coincide.

Feature	Missed fan-out	Legit high out-degree	$\|d\|$
Out-degree	72	72	0.37
Unique recipients	3	3	0.02
Total sent	39,501	41,001	0.34
Out-tx per hour	6.0	6.0	0.38
Fresh-recipient ratio	0.50	0.50	0.35

Table 6. Feature ablation (LightGBM, XGBoost). For LightGBM we add the per-group rows (+fan-out, +neighborhood, fixed HP) and a matched-budget retune of the 37-feature baseline. Bold marks the near-identical fan-out recall of `full_50` and the retuned baseline.

Model	Set	#feat	F1	AUC	cycle	fan_in	fan_out	scatter
LGB	baseline_37	37	0.9385	0.9924	0.966	0.939	0.7201	0.971
LGB	+ fan-out (7)	44	0.9417	0.9923	0.990	0.927	0.7031	0.971
LGB	+ neighborhood (6)	43	0.9557	0.9946	0.993	0.973	0.7474	0.996
LGB	full_50	50	0.9529	0.9944	0.997	0.976	**0.7509**	0.996
LGB	baseline_37 (retuned)	37	0.9362	0.9923	0.973	0.957	**0.7543**	0.971
XGB	baseline_37	37	0.9366	0.9933	0.966	0.927	0.7065	0.957
XGB	full_50	50	0.9527	0.9949	0.993	0.988	0.7543	0.996

Why fan-out is missed. Partitioning illicit fan-out accounts by out-degree localizes the gap in the high out-degree *dispersers* (recall 0.63), not their leaf recipients (recall 0.81). In Table 5, the missed dispersers, against legitimate high-out-degree accounts (top licit out-degree quartile, $\geq$71), are nearly indistinguishable on outgoing behavior (medians nearly coincide mean $|d| = 0.30$): illicit dispersal overlaps with legitimate distribution at the account level.

5.3 Feature Ablation

To isolate the contribution of the fan-out-targeted and neighborhood-context features (C2), we retrained LightGBM and XGBoost with `baseline_37` (all features except the 13 in C2) and `full_50` (all features), holding hyperparameters and SMOTE 1:1 constant and re-tuning the threshold per variant. We then added two checks: a per-group decomposition (baseline plus only the seven fan-out features, or only the six neighborhood features), and a fairness control in which the 37-feature baseline is itself retuned with the same Optuna budget (30 trials, 5-fold CV) used for the 50-feature model.

Under fixed hyperparameters the 13 features raise fan-out recall by +3.07 pp (LightGBM), which the per-group rows attribute to the neighborhood group (+2.7 pp), not the fan-out-specific group (−1.7 pp). This gain is largely a tuning artifact: a matched-budget retune of the 37-feature baseline reaches 0.7543 on fan-out, matching `full_50` (0.7509) within seed noise (±0.6 pp). What survives matched tuning is an aggregate gain (F1 +1.7 pp, PR-AUC from 0.950 to 0.963), not a fan-out one. The gap is thus not reduced by account-level feature engineering, reinforcing C1 and motivating the subgraph reformulation of Section 6.

Table 7. Sampling robustness on LightGBM (top) and XGBoost (bottom). All variants use the same Optuna-tuned hyperparameters and validation-tuned threshold. "cw" = class_weight balanced.

Model	Strategy	F1	AUC	cycle	fan_in	fan_out	FP % licit
LGB	SMOTE 1:1 (paper baseline)	0.953	0.994	0.997	0.976	**0.7509**	0.032
LGB	SMOTE+Tomek	0.955	0.994	0.993	0.957	0.7270	0.008
LGB	RandomUnderSampler 1:1	0.924	0.995	0.990	0.951	0.7167	0.085
LGB	no resample + cw	**0.960**	0.994	0.993	0.988	**0.7509**	0.015
XGB	SMOTE 1:1	0.953	0.995	0.993	0.988	0.7543	0.037
XGB	SMOTE+Tomek	0.951	0.995	0.993	0.991	0.7543	0.043
XGB	RandomUnderSampler 1:1	0.934	0.996	0.993	0.954	0.7133	0.058
XGB	no resample + cw	0.958	0.995	0.993	0.985	0.7543	0.020

5.4 Sampling Robustness

The training distribution is highly imbalanced (licit:fraud $\approx$ 71:1). Our paper baseline applies SMOTE 1:1, which raises the concern that synthetic oversampling could inflate fan-out recall by interpolating between rare positive examples. To distinguish the structural difficulty of fan-out from a possible SMOTE artifact, we retrain LightGBM and XGBoost under four resampling strategies, holding hyperparameters and threshold tuning constant.

The key finding is that fan-out recall is identical (0.7509 LGB / 0.7543 XGB) under SMOTE 1:1 and under cost-sensitive learning with no resampling. If SMOTE were the cause of the observed fan-out recall, removing it should reduce it; it does not. SMOTE+Tomek and RandomUnderSampler both *decrease* fan-out recall, consistent with information loss at the licit/fraud boundary. On this dataset, the fan-out recall gap is therefore robust across the evaluated sampling strategies rather than an artifact of the resampling choice; over five LightGBM seeds it is also stable (fan-out recall 0.743 ± 0.006, aggregate F1 0.954 ± 0.001). The no-resample + class_weight variant additionally achieves the highest F1 (0.960 LGB) with halved licit-side false-positive rate (0.015% vs. 0.032%); this is reported as a robustness finding and discussed in Section 6.

5.5 Robustness to Pattern Leakage

The neighborhood-context features of Section 4.2 are computed on the global transaction graph, so an account-level stratified split may let accounts of the same injected pattern appear in train and test simultaneously, leaking pattern-level structure across folds. We re-evaluate LightGBM and XGBoost under a *pattern-disjoint* split: each of the 800 alert patterns ($\sim$7.5 accounts each) is assigned in its entirety to one fold, stratified by typology; licit accounts use a random 72/8/20 partition. Hyperparameters, SMOTE 1:1, and threshold tuning are held constant.[5] Under the stratified split, all 638 patterns contributing to

[5] The stratified baseline was re-trained independently for this experiment; the marginal deviations from Table 3 (e.g., LightGBM F1 0.9551 vs. 0.9529, fan-out recall 0.761 vs. 0.751) reflect run-to-run training variance, not a protocol change.

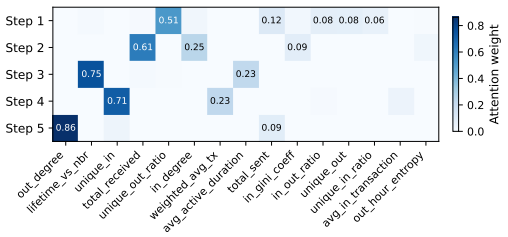

Fig. 2. TabNet attention weights across five decision steps, aggregated over fraud-classified test accounts. Darker cells indicate higher attention.

test also appear in train; under the pattern-disjoint split this overlap is zero by construction. Because the neighborhood features are computed on the global transaction graph, a test account's feature values still incorporate its own and its neighbors' transactions, some of which belong to training accounts. The split removes pattern-level leakage, as no injected pattern spans train and test, but it does not remove all graph-level information sharing: feature values are still computed over the full graph, test accounts included [27].

Aggregate F1 drops by only 0.29 pp for LightGBM (0.9551 to 0.9523) and 0.41 pp for XGBoost (0.9527 to 0.9485), a small absolute change, so aggregate performance is robust to pattern leakage under the pattern-disjoint evaluation. Cycle and scatter-gather are unchanged (recall >0.99); fan_in and fan_out absorb most of the leakage penalty (LGB $-1.31/-0.77$ pp, XGB $-3.95/-4.48$ pp), consistent with their graph-topological nature, but the fan-out gap persists, aligning with Section 5.4. LightGBM is more robust than XGBoost (3–6× smaller per-typology drops), reinforcing its role as the primary detection model.

5.6 Multi-Perspective Explainability

To investigate the sources of the fan-out detection gap, we applied three complementary explainability methods across two model families. On LightGBM we used *TreeSHAP* [22] for marginal attribution and *DiCE* [24] for counterfactual perturbation. On TabNet we used intrinsic *attention masks* (sequential feature selection). We cross-check the attention findings with TreeSHAP and DiCE, whose formally established properties provide a paradigm-independent reference. The section reports the per-method findings, then attribution-sensitivity and family-level stability checks.

TabNet Attention Masks. TabNet's attention mechanism selects features at each of its five sequential decision steps. Figure 2 shows the aggregate attention weights across fraud-classified test samples.

The masks are sparse and concentrate on a few features per step: step 1 spreads attention across counterparty diversity, volume, and amount inequality; step 2 concentrates on inbound flow (*total_received*, 52%); step 3 on neighborhood context (*lifetime_vs_nbr*, 51.6%); steps 4–5 concentrate on a single feature

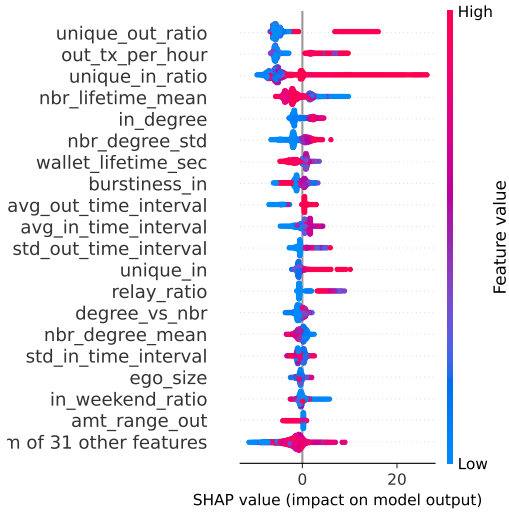

Fig. 3. Exact TreeSHAP attribution for LightGBM (AMLSim, top 20 features). An importance ordering, not a correctness proof; rankings are method-dependent (Table 8).

each (*unique_in* 82.2%, *out_degree* 99.8%). We read these as model-internal feature-selection traces and do not interpret the step order as a screening-to-gating decision process.

The strong concentration in steps 4–5 indicates sensitivity to a small number of features, which the counterfactual analysis below corroborates.

Counterfactual Analysis. DiCE generates diverse counterfactual examples identifying the minimal modifications needed to reverse a classification. We generate 150 counterfactuals for each of two scenarios (50 seed accounts, 3 counterfactuals each): high-confidence fraud and fan-out borderline. All numeric features were left mutable and no graph-consistency constraints were imposed, so these are feature-space counterfactuals: they characterize the model's decision boundary, not necessarily transaction graphs an account could realize.

Three observations emerge. First, TabNet's classifications can be reversed by modifying a median of just two features regardless of confidence level, with 77% of high-confidence counterfactuals requiring exactly two changes. This is consistent with the attention mask concentration in steps 4–5 and points to a simple decision boundary. Second, the most frequently modified features are *nbr_cc_mean* (26.7% of high-confidence counterfactuals; 8.0% of fan-out borderline) and *avg_out_time* (18.7%; 9.3%). Third, the fan-out boundary is even more fragile than the general fraud boundary: no counterfactual requires more than two modifications, and 35% of fan-out counterfactuals flip the classification by changing a single feature. This asymmetry is consistent with the low fan-out recall observed in Section 5.2.

Table 8. Deletion test: PR-AUC drop after masking the top-k features of each ranking (median imputation), against random-k (mean of 10 draws) and bottom-k controls. Baseline PR-AUC: 0.963 (LightGBM), 0.947 (TabNet).

	LightGBM (TreeSHAP ranking)			TabNet (attention ranking)		
k	top-k	random-k	bottom-k	top-k	random-k	bottom-k
1	0.014	0.002	0.000	0.019	0.013	0.008
3	**0.461**	0.029	0.000	0.122	0.069	0.032
5	0.471	0.025	0.000	0.446	0.157	0.032
10	0.729	0.085	0.001	0.677	0.189	0.077

Attribution Sensitivity and Family-Level Stability. As an attribution-sensitivity (importance sanity) check rather than a proof of correctness, masking the three features ranked highest by TreeSHAP collapses LightGBM PR-AUC from 0.963 to 0.502, against a mean drop of 0.029 for random feature triples and 0.000 for the three lowest-ranked (Table 8).[6] This corroborates the counterfactual analysis (median two-feature flips): two independent probes agree that the decision rests on a handful of features, which TreeSHAP identifies. At the family level, the two tree-side measures agree: the 13 C2 features carry 33.9% (TreeSHAP) and 29.4% (split count) of total importance. This is of the same order as the 26% chance share of 13 features out of 50 ($p \geq 0.19$, randomization test); the claim is cross-measure stability, not outsized importance.

By contrast, agreement on *individual* features is weak. We measure it via the Jaccard similarity between top-k feature lists ($k = 10$): the overlap across pairs ranges from 0.00 (TabNet attention vs. LGB split count) to 0.33 (SHAP-LGB vs. LGB split count). This is the pattern predicted by the attention-as-explanation literature [14]: methods that operate on different mathematical objects need not agree pointwise. Feature-level explanations are therefore method-dependent and should not be read as a single ground-truth ranking. Indeed, a global attention ranking is aggregation-dependent (C2-family share 19–75% across two conventions), with deletion selectivity within 1.5–4.5× the random baseline, versus up to 19× for TreeSHAP (Table 8). We argue, though we do not validate this with practitioners, that the operationally relevant level of agreement is the family level, since AML auditors plausibly reason in terms of evidence *types* (temporal anomaly, neighborhood deviation, fan-out signature) rather than individual numerical features.

Replication on LightGBM: TreeSHAP and DiCE. Explainability conducted on a non-best model risks conclusions that are model-specific. We therefore replicate the per-typology analysis on LightGBM (best F1, exact TreeSHAP).

The DiCE counterfactual analysis supports the fragility hypothesis: across the four typologies, the median number of feature changes needed to flip a fraud prediction to licit is 2 (cycle: 3, IQR $[2, 7]$; fan_in / fan_out / scatter_gather:

[6] Median-imputed inputs are off-manifold; the random- and bottom-k controls share the imputation and bound this artifact.

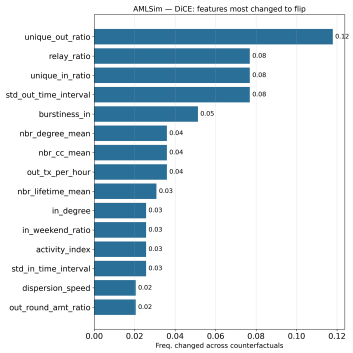

Fig. 4. DiCE on LightGBM (AMLSim): share of counterfactuals modifying each feature (top 15). The most-changed features (*unique_out_ratio*, *relay_ratio*, *unique_in_ratio*, outbound timing) match the flow/timing families identified by TreeSHAP.

2, IQR $[1, 2]$). Cycle requires more changes because the model is most confident on cycle (recall 0.997), so even the "hardest" cycle frauds (probability closest to 0.5) sit further from the boundary than the hardest fan-out frauds. The top 2 features flip on fan-out replicates the TabNet finding on a model trained from a different inductive bias (Figure 4).

The top-10 feature-set overlap between DiCE-LGB and SHAP-LGB top features is 0.250, 0.176, 0.538, and 0.176 for cycle, fan_in, fan_out, and scatter_gather, respectively. Agreement peaks on the typology where the model struggles most: on fan-out the two LightGBM-side methods share 7 of 10 top features (including `nbr_lifetime_mean` and `out_tx_per_hour`, both NEW). The higher agreement observed for fan-out suggests that both methods emphasize similar feature families in this case. The difference between SHAP (average contribution) and DiCE (boundary proximity) is informative rather than contradictory and supports using both [19].

5.7 Replication on a More Realistic Synthetic Benchmark: AMLworld

To test whether the per-typology findings survive on more realistic data, we ported the pipeline to AMLworld HI-Medium [1] with minimal adaptations: 2.08M accounts and 31.9M transactions over 28 days, with 5 currencies, 7 payment formats, and 7 named laundering typologies (Figure 1) plus a diffuse "other" class. We address the account-level *structured* task (accounts whose alerts carry a named typology; "other" excluded) under a stratified 72/8/20 split: the held-out test set contains 411,489 accounts, of which 4,456 (1.08%) are illicit. The feature set extends the 50-feature descriptor of Section 4 with 13 subgraph features adapted from the Graph Feature Preprocessor (GFP) [6] (reciprocity, two/three-hop cycle participation, two-hop reachability, fan-in/out concentration, relay shares) and 16 channel/currency features (payment-format ratios and entropy,

Table 9. AMLworld HI-Medium: per-typology recall at the F1-optimal threshold (LightGBM, 79 features), with 95% Wilson intervals. Overall recall 0.691.

Typology	n	Recall	95% CI
Cycle	336	0.872	[0.832, 0.904]
Scatter-gather	390	0.828	[0.788, 0.862]
Fan-in	755	0.789	[0.759, 0.817]
Random	320	0.750	[0.700, 0.794]
Stacked bipartite	1,049	0.677	[0.648, 0.704]
Fan-out	806	**0.624**	[0.590, 0.657]
Bipartite	800	**0.516**	[0.482, 0.551]

currency counts, cross-currency and foreign-currency shares), for 79 features in total. The detector is LightGBM with cost-sensitive class weighting, the variant selected by the sampling-robustness study of Section 5.4, with no per-dataset hyperparameter search.

Detection. On the structured-typology subset, a restricted diagnostic task, the model reaches PR-AUC 0.818 and ROC-AUC 0.986; on the full population (including the diffuse "other" class) PR-AUC falls to 0.516 (Section 6), so the structured figure is a diagnostic upper bound rather than the general AMLworld result. At the F1-optimal threshold: precision 0.906, recall 0.691, minority-F1 0.784, balanced accuracy 0.845, MCC 0.789; recall at 1% false-positive rate is 0.825 (threshold selected on the validation split). At this threshold the model raises 3,399 alerts on the 411,489-account test set (3,078 true positives, 321 false positives; precision 0.906): an alert volume of 0.83% of accounts, or 1.10 alerts per true positive. The 16 channel/currency features alone contribute +8.0 pp of PR-AUC (from 0.738 to 0.818): on this more realistic synthetic data, where channel and currency heterogeneity exists, targeted feature engineering in the spirit of C2 remains the single most effective intervention we tested.

Per-typology replication. Table 9 disaggregates recall over the seven named typologies. The AMLSim findings replicate: cycle, scatter-gather, and fan-in remain the easiest patterns (0.79–0.87), and fan-out is again systematically missed (0.624, ∼25 pp below cycle). A *second* blind spot, which AMLSim could not reveal, also emerges: bipartite laundering is detected at 0.516, as its many-to-many flows overlap with legitimate high-volume activity (marketplace- and payroll-like behavior). Aggregate metrics (PR-AUC 0.82, balanced accuracy 84.5%) hide that roughly half of bipartite laundering goes undetected, reinforcing C1 on more realistic synthetic data.

Temporal split. The headline figures use a random split. Under the canonical temporal protocol (train on the first 70% of the timeline, test on the future 30%; identical 79-feature pipeline), structured PR-AUC falls to 0.578 (from 0.821) and full-population to 0.373 (from 0.517); the engineered families still help (channel/currency +8.4 pp) and discrimination stays high (ROC 0.964). We report this as the temporal-split performance.

Table 10. Published AMLworld results vs. this work. Levels differ (edge = transaction classification, node = account classification) and are *not* directly comparable; minority-class F1 as reported.

Study	Dataset	Level	min-F1 (%)
Altman et al. [1]	HI-Medium	edge	65.7
Blanuša et al. [6]	HI-Medium	edge	65.7
Egressy et al. [10]	HI-Medium	edge	66.5
GCPAL [21][a]	AMLworld	node	≈78
This work	HI-Medium	node	78.4[b]

[a] Semi-supervised (∼1% labels); evaluation unit not verifiable from the published version.
[b] Structured-typology subset; PR-AUC 0.818; full-population PR-AUC 0.516.

Positioning and caveats. Published AMLworld results (Table 10) are predominantly *edge-level* (transaction classification); our 78.4% is *account-level* on the structured subset (a different unit of analysis), reported for context rather than as a direct comparison. To our knowledge, no standard account-level AMLworld benchmark exists. Two caveats: the headline uses a random stratified split (the temporal result is reported above), and the full-population task (including the diffuse "other" class) is harder still (PR-AUC 0.516 random,0.373 temporal).

Explainability replication. TreeSHAP on the 79-feature model ranks neighbor lifetime (*nbr_lifetime_mean*, a C2 neighborhood-context feature) as the top driver, and the 29 added features (subgraph + channel) carry 17.2% of total importance with no single feature above 3.8%, a distributed signal rather than an encoding artifact. DiCE counterfactuals most frequently modify the channel/currency features (credit-card share in 64% of counterfactuals, ACH 49%, cross-currency 44%): the engineered families are pivotal at the decision boundary, replicating the family-level convergence of Section 5.6 on more realistic synthetic data.

6 Discussion and Conclusions

The per-typology analysis (C1) shows that aggregate metrics mask large disparities: with all four architectures above 0.93 F1, fan-out recall is only 0.693–0.761. This points to a persistent limitation of the evaluated account-level representation: illicit fan-out overlaps with legitimate distribution, the missed dispersers being nearly indistinguishable from legitimate high-out-degree accounts (mean $|d| = 0.30$, Section 5.2). Account-level feature engineering improves aggregate performance (F1/PR-AUC) but does not, by itself, close the fan-out gap: once the 37-feature baseline is retuned at a matched budget, the added features leave fan-out recall essentially unchanged (Section 5.3). This points to a structural limitation of the account-level representation rather than a lack of features, and the gap is also robust across sampling strategies and seeds (Section 5.4), so it is not a tuning, sampling, or run-to-run artifact.

The explainability analysis (C3) probes rather than assumes the explanations: a TreeSHAP deletion check confirms the model relies on the top-ranked

features (PR-AUC from 0.963 to 0.502 under top-3 masking, an importance test, not a correctness proof), the tree-side measures agree at the family level (29–34%), but per-feature agreement is weak (top-10 Jaccard 0.00–0.33) and attention rankings are aggregation-dependent: feature-level explanations are method-dependent. The AMLworld replication closes the loop: fan-out stays low on more realistic synthetic data (0.624), bipartite emerges as a second blind spot (0.516), and channel/currency features give the largest single gain (+8.0 pp PR-AUC).

Several limitations should be acknowledged. First, AMLSim patterns are generated by fixed rules and are more regular than real laundering schemes; the high aggregate AUC (>0.99) should not be interpreted as indicative of expected real-world performance, where lower accuracy is typically reported [17]. The AMLworld validation now quantifies this drop directly: PR-AUC falls from 0.963 (AMLSim) to 0.818 (HI-Medium, structured task) and to 0.516 on the full population. The synthetic fraud prevalence (1.38%) is also one to two orders of magnitude higher than in real banking; at lower prevalence, the false positive burden would increase substantially even at comparable precision. Second, the per-typology test set (∼300 accounts per pattern) limits the statistical precision of within-typology model comparisons. Third, no prior AMLSim-based method was reproduced on our configuration; Table 1 contextualizes the configuration differences but does not substitute for a like-for-like benchmark. Similarly, we train no GNN baselines ourselves: the published edge-level GNN results on AMLworld (Table 10) serve as the external reference point, and an account-level GNN comparison remains future work.

Three directions emerge from this study. First, the residual fan-out gap appears structural at the account level (Sections 5.3–5.4); a subgraph-level reformulation, in which the entire dispersal pattern is scored as a single sample (as in [23,5]), is the natural next step. Second, the neighborhood representation could be enriched by extending from 1-hop to 2-hop or more statistics or by integrating lightweight GNN embeddings as additional features, which could strengthen the fan-out decision boundary where current hand-crafted features appear insufficient. Third, the random-to-temporal gap quantified in Section 5.7 (structured PR-AUC from 0.82 to 0.58) and the full-population "other"-inclusive task remain the hardest open settings. Evaluation on real banking data with typology annotations remains the ultimate validation target, though access to such data is restricted by confidentiality constraints.

References

1. Altman, E., Blanuša, J., Von Niederhäusern, L., Egressy, B., Anghel, A., Atasu, K.: Realistic synthetic financial transactions for anti-money laundering models. Advances in Neural Information Processing Systems **36**, 29851–29874 (2023)
2. Anaroch, C., Tosakulvong, W., Wattanapongsakorn, N.: Anti-money laundering detection using traditional, deep, and hybrid machine learning: A performance comparison. In: 2025 6th International Conference on Big Data Analytics and Practices (IBDAP). pp. 239–244. IEEE (2025)

3. Arik, S.Ö., Pfister, T.: Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6679–6687 (2021)
4. Barabasi, A.L.: The origin of bursts and heavy tails in human dynamics. Nature **435**(7039), 207–211 (2005)
5. Bellei, C., Xu, M., Phillips, R., Robinson, T., Weber, M., Kaler, T., Leiserson, C.E., Chen, J., et al.: The shape of money laundering: Subgraph representation learning on the blockchain with the elliptic2 dataset. arXiv preprint arXiv:2404.19109 (2024)
6. Blanuša, J., Cravero Baraja, M., Anghel, A., Von Niederhäusern, L., Altman, E., Pozidis, H., Atasu, K.: Graph feature preprocessor: Real-time subgraph-based feature extraction for financial crime detection. In: Proceedings of the 5th ACM International Conference on AI in Finance. pp. 222–230 (2024). https://doi.org/10.1145/3677052.3698674
7. Cardoso, M., Saleiro, P., Bizarro, P.: Laundrograph: Self-supervised graph representation learning for anti-money laundering. In: Proceedings of the third ACM international conference on AI in finance. pp. 130–138 (2022)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
9. Cheng, D., Ye, Y., Xiang, S., Ma, Z., Zhang, Y., Jiang, C.: Anti-money laundering by group-aware deep graph learning. IEEE Transactions on Knowledge and Data Engineering **35**(12), 12444–12457 (2023)
10. Egressy, B., von Niederhäusern, L., Blanuša, J., Altman, E., Wattenhofer, R., Atasu, K.: Provably powerful graph neural networks for directed multigraphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 11838–11846 (2024). https://doi.org/10.1609/aaai.v38i10.29069
11. 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)
12. European Parliament and Council of the European Union: Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act). Official Journal of the European Union, L series (2024), entered into force 1 August 2024
13. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Advances in neural information processing systems **35**, 507–520 (2022)
14. Jain, S., Wallace, B.C.: Attention is not explanation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3543–3556 (2019)
15. Jensen, R.I.T., Iosifidis, A.: Fighting money laundering with statistics and machine learning. IEEE Access **11**, 8889–8903 (2023)
16. Johannessen, F., Jullum, M.: Finding money launderers using heterogeneous graph neural networks. The Journal of Finance and Data Science **11**, 100175 (2025). https://doi.org/10.1016/j.jfds.2025.100175
17. Jullum, M., Løland, A., Huseby, R.B., Ånonsen, G., Lorentzen, J.: Detecting money laundering transactions with machine learning. Journal of Money Laundering Control **23**(1), 173–186 (2020)

18. Karim, M.R., Hermsen, F., Chala, S.A., De Perthuis, P., Mandal, A.: Scalable semi-supervised graph learning techniques for anti money laundering. IEEE Access **12**, 50012–50029 (2024)
19. Kute, D.V., Pradhan, B., Shukla, N., Alamri, A.: Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. IEEE Access **9**, 82300–82317 (2021)
20. Lo, W.W., Kulatilleke, G.K., Sarhan, M., Layeghy, S., Portmann, M.: Inspection-l: self-supervised gnn node embeddings for money laundering detection in bitcoin. Applied Intelligence **53**(16), 19406–19417 (2023)
21. Lu, H., Wang, H.: Graph contrastive pre-training for anti-money laundering. International Journal of Computational Intelligence Systems **17**(1), 307 (2024)
22. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
23. Meng, Y., Li, Z.: A money laundering structure detection method based on incremental transaction analysis. In: 2023 6th International Conference on Data Science and Information Technology (DSIT). pp. 20–30. IEEE (2023)
24. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 conference on fairness, accountability, and transparency. pp. 607–617 (2020)
25. Phyu, T.H., Uttama, S.: Improving classification performance of money laundering transactions using typological features. In: 2023 7th International Conference on Information Technology (InCIT). pp. 520–525. IEEE (2023)
26. Phyu, T.H., Uttama, S.: Enhancing money laundering detection addressing imbalanced data and leveraging typological features analysis. In: 2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE). pp. 330–336. IEEE (2024)
27. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. arXiv preprint arXiv:1811.05868 (2018)
28. Suzumura, T., Kanezashi, H.: AMLSim: Anti-money laundering simulator. https://github.com/IBM/AMLSim (2021)
29. Trerotola, M., Parente, M., Calvaresi, D.: A hybrid multi-agent system for early scam detection in crypto-assets. Applied Sciences **16**(7), 3122 (2026)
30. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
31. 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)