High-Performing Machine Learning Pipeline for Anti-Money Laundering Detection

Leonardo Rocci, Valeria Avino, Riccardo Soleo
Sapienza University of Rome

***Abstract*—This paper presents a high-performing anti-money laundering (AML) detection pipeline developed in a competitive hackathon setting. The model combines graph-based features, ensemble stacking, and tailored imbalance handling optimized for a multi-component evaluation metric. It was trained and evaluated on a dataset of anonymized banking transactions, comprising 55,307 labeled training and 23,743 unlabeled test transactions, with features including account identifiers, payment type, transaction amount, account types, and 30-day average balances. The dataset reflects realistic financial behavior, including noise, missing values, and complex interdependencies. Our approach captures coordinated, multi-entity laundering schemes, achieving strong predictive performance and demonstrating the effectiveness of graph-based reasoning in AML detection.**

I. Introduction

Money laundering poses a significant threat to financial systems worldwide. Financial institutions require robust Anti-Money Laundering (AML) measures to detect suspicious transactions. This work details a machine learning pipeline designed to classify transactions as legitimate or fraudulent, addressing the core challenge of highly imbalanced datasets, where fraudulent transactions are rare.

Graph-based methods have proven particularly effective for AML [1]–[3], as fraudulent accounts often operate in coordinated networks. By combining graph-derived features with ensemble learning, our pipeline can detect complex multi-entity schemes that evade traditional tabular analysis.

II. Contributions

The main contributions of this work are as follows:

1) **Graph-based feature engineering for AML:** We represent the transaction dataset as a directed, weighted graph and extract network-structural features that capture influence, communities, and transaction flows.
2) **Stacked ensemble pipeline:** Combining gradient boosting algorithms (LightGBM, CatBoost) with a meta-model (logistic regression) for improved predictive performance on highly imbalanced data.
3) **Evaluation with top-N ranking:** Predictions are calibrated and ranked according to a metric reflecting real-world AML priorities, emphasizing the capture of high-risk transactions.
4) **Hackathon-scale validation:** The model demonstrates strong performance on a realistic dataset with tens of thousands of transactions, reflecting noisy, incomplete, and interconnected financial behavior.

III. Dataset and EDA

A preliminary exploration of the transaction dataset highlighted both graph-structural patterns and label distribution characteristics relevant to model design.

Graph Structure

Representing accounts as nodes and transactions as directed edges yielded a **sparse graph** with the following properties:

- Skewed degree distribution: A few “hub” accounts exhibited very high in-degree or out-degree, while the majority of nodes had low connectivity. Such hubs are often associated with money “mixing” or central aggregation points in laundering schemes.
- Community structure: Louvain community detection revealed dense clusters of accounts transacting primarily within the same group, with occasional large-value transactions crossing communities — a pattern consistent with layering in anti-money laundering (AML) typologies.
- Flow asymmetry: Several accounts displayed unbalanced in/out degree ratios, acting either as sources (sending to many recipients) or sinks (receiving from many senders), both of which can be indicative of illicit consolidation or distribution of funds.
- Proximity of suspicious nodes: Fraudulent accounts tended to be only a few hops apart in the transaction network, suggesting the presence of interconnected illicit subgraphs.

These patterns motivated the inclusion of graph-based features such as degree centrality, betweenness centrality, PageRank, community assignments, and in/out degree ratios in the model pipeline.

Label Distribution

The (training) dataset exhibited severe class imbalance: fraudulent transactions constituted only a small fraction of the total, with the vast majority labeled as legitimate. This imbalance implied that:

- Standard accuracy was not an appropriate evaluation metric.
- A focus on precision in the top-ranked predictions (e.g., top-N transactions) would better align with operational goals.
- Ranking-based strategies combined with probability calibration were more suitable than fixed decision thresholds (e.g., 0.5).

IV. Methodology

The AML detection pipeline combines advanced feature engineering, ensemble learning, and careful calibration. The workflow consists of three main stages:

1) Graph-based feature extraction capturing structural patterns of suspicious activity.
2) Base and meta-model training using stacked ensembles of gradient boosting algorithms.
3) Final prediction calibration and top-N ranking optimized for AML detection metrics.

No explicit class reweighting or resampling techniques were applied during training. Instead, the pipeline relied on a combination of rich feature engineering — including graph-derived features — and an ensemble stacking approach. This, together with the multi-component evaluation metric used in the competition, implicitly encouraged the model to improve detection of the minority (fraudulent) class despite the pronounced class imbalance.[1]

Figure 1 illustrates the full pipeline.

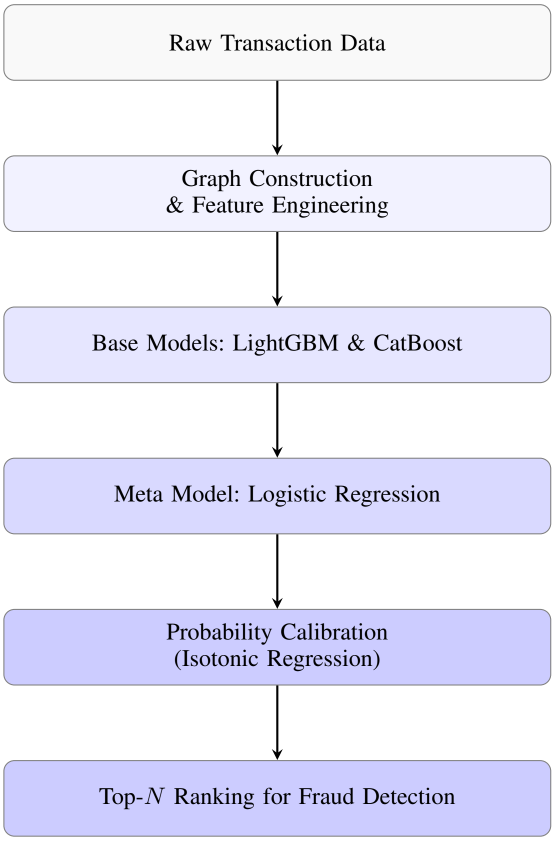
Raw Transaction Data
Graph Construction & Feature Engineering
Base Models: LightGBM & CatBoost
Meta Model: Logistic Regression
Probability Calibration (Isotonic Regression)
Top-N Ranking for Fraud Detection

Fig. 1. Overview of the High-Performance AML Detection Pipeline

V. Graph-Based Feature Engineering

Fraudulent actors often route funds through multiple accounts to conceal illicit activity. We represent the transaction dataset as a **directed, weighted graph** $G = (V, E)$, where nodes $v \in V$ are accounts, and edges $(u, v) \in E$ correspond to transactions with weights equal to the transaction amount.

Graph-derived features capture *structural and relational patterns* that are critical for detecting suspicious behavior, including:

- Node influence (PageRank of accounts) [1].
- Community membership and size (Louvain clusters) [2].
- Flow positions (net sender vs receiver behavior).
- Local clustering coefficients to detect layering structures.
- Transaction pair frequencies to detect repeated suspicious patterns.

TABLE I
Engineered Graph-Based Features for AML Detection

Feature Name	**Description**
pr_from, pr_to	PageRank score of sender/receiver accounts
flow_position_from, flow_position_to	Net flow tendency: $(\text{Sent} - \text{Received})/(\text{Sent} + \text{Received} + \epsilon)$
community_from, community_to	Louvain community ID for sender/receiver
is_same_community	Binary flag indicating same community
community_from_size	Size of sender's community
clustering_from, clustering_to	Local clustering coefficient: $C_v = 2T(v)/[k_v(k_v - 1)]$
pair_count	Number of transactions between same sender–receiver pair

VI. Stacked Model Pipeline

To address the binary classification problem of detecting suspicious transactions, we employed an ensemble learning strategy leveraging gradient boosting and stacking, optimized via Bayesian hyperparameter tuning with Optuna.

1) Base Learners: We selected two high-performance gradient boosting algorithms as base learners, each with complementary strengths:

- **LightGBM** [4]: An efficient histogram-based gradient boosting framework designed for large-scale datasets. LightGBM grows trees *leaf-wise* rather than level-wise: at each step, it expands the leaf that provides the largest reduction in loss, producing deeper, asymmetric trees that capture complex interactions. This strategy, combined with histogram-based feature binning, enables fast training and strong predictive power while maintaining memory efficiency. LightGBM also supports native handling of missing values and offers extensive regularization options, which are crucial in financial applications with noisy or incomplete data.
- **CatBoost** [5]: A gradient boosting library tailored for categorical data and robust generalization. Unlike LightGBM, CatBoost grows trees *symmetrically* (level-wise), resulting in balanced tree structures that often generalize better under noisy or sparse conditions. Its key innovation is *ordered boosting*, which prevents target leakage by ensuring that the encoding of categorical features is based

[1] A possible extension would be the use of conditional diffusion models to learn the distribution of fraudulent transactions and generate synthetic samples for data augmentation.

only on information available at training time. CatBoost's efficient target-based encoding makes it especially suitable for high-cardinality categorical features, such as community identifiers or transaction types.

Formally, for a binary classification problem with labels $y_i \in \{0,1\}$, the prediction of a gradient-boosted tree ensemble (LightGBM, CatBoost) is:

$$\hat{y}_i = \sigma\Big(\sum_{m=1}^{M} f_m(\mathbf{x}_i)\Big),$$

where $f_m \in \mathcal{F}$ are decision trees, M is the number of boosting iterations, and $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic link.

At each iteration t, the boosting update adds a new tree fitted to the negative gradient of the loss:

$$f_t = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell\Big(y_i, \hat{y}_i^{(t-1)} + f(\mathbf{x}_i)\Big),$$

with $\ell(y,\hat{y}) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$ for the binary cross-entropy loss.

By combining LightGBM's leaf-wise, high-capacity learners with CatBoost's balanced, categorical-aware learners, our ensemble benefits from both the ability to model complex patterns and the robustness to overfitting and data leakage.

2) Meta Learners:

- **Logistic Regression**: Predictions from the base learners were combined via a Logistic Regression meta-model, forming a two-layer stacking ensemble. This meta-model learns an optimal linear combination of the base predictions:

$$p_i = \hat{y}_i^{\text{meta}} = \sigma\Big(\beta_0 + \sum_{j=1}^{K} \beta_j \hat{y}_{i,j}^{\text{base}}\Big)$$

  where:
  - $\hat{y}_i^{\text{meta}}$ is the out-of-fold prediction of base model j for sample i,
  - K is the number of base models (here, 2: LightGBM, CatBoost),
  - β_j are logistic regression coefficients,
  - σ is the sigmoid function.
- **Isotonic Regression**: Since the evaluation metric focuses on the top-N ranking of predicted frauds, it is crucial that the model outputs well-calibrated probabilities. We applied Isotonic Regression for probability calibration:

$$\hat{p}_i^{\text{calibrated}} = \text{Isotonic}(\hat{y}_i^{\text{meta}})$$

  where f_{iso} is a non-decreasing function minimizing

$$\sum_i (y_i - f(\hat{y}_i))^2.$$

  This ensures that the predicted scores are monotonic and reflect the true likelihood of fraud.

VII. Top-N Fraud Ranking

The final output consists of a ranked list of transactions by their calibrated fraud probability. Only the top-N transactions (top 485) are flagged for review, aligning with the evaluation metric emphasizing precision at high ranks.

VIII. Hyperparameter Optimization

We utilized **Optuna** [6] to perform Bayesian optimization of the hyperparameters for both LightGBM and CatBoost simultaneously. The objective function maximized the custom competition metric:

$$\text{Score} = \frac{\text{AUC} + \text{Balanced Accuracy} + \text{Fraud Capture Rate}}{3},$$

where the Fraud Capture Rate measured the proportion of fraud cases among the top-N predicted transactions ($N = 485$).

IX. Empirical Observations from Trials

In our experiments, we first replicated the pipeline from [2]—including graph features, an ensemble of LightGBM + CatBoost, a Logistic Regression stacker, and Isotonic Regression calibration—achieving a baseline score of **0.58600**.

A simple yet impactful modification—ranking transactions by fraud probability and selecting the top 485 instead of applying a fixed > 0.5 threshold—increased our score dramatically to **0.76594** (our champion model).

Adding more advanced graph features (e.g., HITS scores, additional community metrics) slightly reduced performance to **0.73560**, suggesting that a targeted, high-signal set of features was optimal in this context.

Final fine-tuning with a proxy leaderboard validation method yielded **0.76594**, but the top-485 tuned ensemble with our core graph features remained the most effective.

X. Results

The proposed stacked model pipeline demonstrates strong performance on both internal cross-validation and the challenge leaderboard. Key findings include:

- Graph-based features contributed significantly to capturing suspicious account behavior.
- Stacking LightGBM and CatBoost improved detection over single models.
- Probability calibration improved the precision of top-N predictions.

XI. Conclusion

This work presents a high-performing AML detection pipeline that leverages graph-based features and stacked ensemble learning to detect fraudulent transactions in large, imbalanced datasets. The model effectively captures coordinated laundering schemes, demonstrating the power of combining network analysis with advanced machine learning.

Acknowledgment

The authors thank University of Genoa – Departments of Political and International Sciences, Economics and Business Studies and Mathematics in collaboration with Young SIS for organizing the event, and Rulex for sponsoring the hackathon.

Appendix

PageRank. PageRank measures the relative importance of an account within the transaction network by considering not only the number of its connections but also the importance of the connected nodes. Accounts with high PageRank often act as central hubs in the network, which makes this metric particularly useful for highlighting potentially critical nodes in fraudulent schemes.

Flow Position. The flow position captures the net tendency of an account to send or receive funds, normalized as

$$\text{Flow Position} = \frac{\text{Sent} - \text{Received}}{\text{Sent} + \text{Received} + \epsilon}.$$

This feature reflects the transactional behavior of each node, distinguishing primarily outgoing accounts from primarily incoming ones—a pattern often indicative of suspicious or anomalous activity.

Community Features. The Louvain algorithm is used to detect communities within the network, assigning each node to a community based on modularity optimization. Features derived from this include the sender's and receiver's community IDs, a binary flag indicating whether the two nodes belong to the same community, and the size of the sender's community. These measures help capture localized clusters and tightly-knit groups, which are often relevant in detecting coordinated fraudulent activity.

Clustering Coefficient. The local clustering coefficient quantifies how close an account's neighbors are to forming a fully connected clique, calculated as

$$C_v = \frac{2T(v)}{k_v(k_v - 1)},$$

where $T(v)$ is the number of triangles around the node and k_v its degree. Accounts embedded in tightly connected neighborhoods may reveal dense substructures typical of coordinated networks, making this metric a useful indicator of suspicious behavior.

Pair Count. The pair count represents the total number of transactions between a given sender and receiver. Repeated interactions between the same accounts can signal structured or orchestrated transfers, which are often a hallmark of fraudulent schemes.

References

[1] B. Dumitrescu, A. Băltoiu, and S. Budulan, "Anomaly detection in graphs of bank transactions for anti money laundering applications," *IEEE Access*, 2021.

[2] N. E. Ahmad, "Anti-money laundering using graph techniques," Doctoral Thesis, University of Porto, 2024.

[3] R. Karim, F. Hermsen, Felix, S. A. Chala, P. D. Perthuis, and A. Mandal, "Scalable semi-supervised graph learning techniques for anti money laundering," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[4] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[6] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2019.