Explainability-guided active learning

for the detection of money laundering

Rens van Haasteren

Host supervisor
Drs. Menno Wiersma,
Senior manager

VU supervisor
Dr. Emma Beauxis-Aussalet,
Assistant professor

Second reader
Prof. dr. Rob van der Mei,
Full professor

VU Amsterdam | Protiviti Netherlands

July 2025

P

PREFACE

This report is submitted as part of the requirements for the Master's programme in Business Analytics at Vrije Universiteit Amsterdam. Its purpose is to present research conducted on explainability-guided active learning techniques for Anti-Money Laundering (AML) detection systems. The research addresses the challenge of improving transaction labeling strategies in AML systems by developing a transaction selection strategy based on explainable AI to improve interpretability, while balancing performance.

This project was conducted at Protiviti Netherlands, within the Data Transformation team. I thank my host organization supervisor at Protiviti Netherlands, Menno Wiersma, for his assistance and insights throughout my research. I would also like to express my sincere gratitude to my university supervisor, Dr. Emma Beauxis-Aussalet, for her guidance and support.

During the writing process, I used Writefull and ChatGPT to restructure specific sentences and to suggest synonyms. All final interpretations and conclusions remain my own.

Contents

0

ABSTRACT

Money laundering sustains criminal enterprises and poses a global threat to financial stability. In 2024, an estimated $3.1T in laundered money flowed through the global economy. Financial institutions are responsible for detecting and reporting suspicious transactions to Financial Intelligence Units (FIUs), which are governmental bodies tasked with identifying money laundering and terrorist financing. Due to limited and often nonspecific feedback from FIUs, many banks have established internal Anti-Money Laundering (AML) teams to manually assess transactions and label transactions as potentially illicit. This process is resource-intensive, costly, and constrained by limited analyst capacity. As a result, supervised machine learning models are increasingly used to monitor transactions, and are typically trained on labeled data covering less than 2% of all transactions, making the selection of transactions for review a key component for effective money laundering detection.

This study introduces a comprehensive active learning framework designed to detect money laundering from limited labeled data. The framework accounts for the temporal structure of the transactions and dynamically adjusts the classification threshold during training at each iteration by maximizing a net value (NV) objective on a validation set. The net value is defined as: $\text{NV} = (b-c)\cdot\text{TP} - c\cdot\text{FP} - (b-c)\cdot\text{FN}$, where c is the cost of investigating an alert and b represents the benefit of detecting an illicit transaction. Although c is roughly estimated, b is varied to simulate different risk preferences of the financial institution; higher values indicate a risk-averse preference, while lower values reflect a risk-seeking approach. The framework evaluates both supervised and unsupervised query strategies, including a novel explainability-guided method, referred to as SHAP-guided profiling, which prioritizes transactions that are most dissimilar to the average SHAP feature importance profile of legitimate transactions. Throughout the active learning process, we explore how the interaction between the benefit per TP b and the query strategy influence recall, precision, and true negative rate (TNR).

During active learning using only 2.08% of the labeled data on the synthetic dataset AMLworld, a performance evaluated on the validation set is achieved that is comparable to passive learning (model trained on the entire training dataset). After applying feature selection and hyperparameter optimization, active learning continues to perform competitively across strategies, compared to passive learning, where query by committee achieves a higher recall and precision than passive learning. SHAP-guided profiling performs worse than random sampling in both recall and precision. Due to the limited interpretability and poor performance SHAP-guided profiling seems to not be suited for a high-risk domain such as money laundering detection, where performance and accountability are critical.

Comparing different values for b while keeping the query strategy fixed reveals that a larger b increases recall and decreases the TNR. This is in line with the risk preference framework. The results demonstrates the feasibility of assigning an economic value to correctly identify illicit transactions based on the risk preference of the financial institution.

The work concludes with a critical discussion on limitations and domain-specific challenges, such as synthetic datasets, evolving criminal tactics, and severe class imbalance. Several directions for future work are proposed to improve both the methodological approach and practical deployment of the detection model, such as introducing nonlinearity in the net value equation, suggestions for additional explainability-guided query strategies, and varying the number of labeled transactions.

Anti-Money Laundering · AML · Active learning · Passive learning · Explainable AI · XAI · SHAP · Explainability-guided query strategy · Risk-preference

1

INTRODUCTION

1.1 BACKGROUND

Crime is a tale as old as time, and traditionally, the proceeds of criminal activity were viewed as an unfortunate but unavoidable consequence of the crimes. However, this perspective has evolved significantly with advances in financial systems and digital technologies. The rise of electronic banking, transaction monitoring, and record-keeping has made it increasingly possible to trace illicit financial flows back to their sources. In response, criminals have developed sophisticated methods to ”launder” money, disguising the origin of the illegal activity by passing it through complex layers of financial transactions to make it appear legitimate. As a result, money laundering has become a critical global challenge, which means that detecting and preventing money laundering provides significant social benefits. In 2024, the estimated global volume of laundered money reached $3.1 trillion [69], underscoring its scale and systemic impact. It enables and sustains a wide range of criminal enterprises, such as drug trafficking, human trafficking, and terrorist financing, by injecting illegally obtained funds into the formal economy. Launderers exploit a variety of financial channels, including traditional bank transfers, payment processors, shell corporations, and, increasingly, cryptocurrencies.

Financial institutions play a central role in the fight against money laundering. They are legally obligated to monitor financial activity and report suspicious transactions through Suspicious Activity Reports (SARs) submitted to Financial Intelligence Units (FIUs). FIUs serve as national centers for the receipt, analysis, and dissemination of financial intelligence. They compile SARs and combine them with other data sources, such as information from other FIUs and Open Source Intelligence (OSINT) [73], to build cases on specific accounts or entities. If suspicion is substantiated, the case is escalated to law enforcement for potential criminal investigation. However, the effectiveness of this system is hampered by a lack of feedback from FIUs and the problem that only reported transactions are reviewed, meaning many illicit transactions likely go undetected. To address this, banks have developed internal AML teams that manually label transactions as suspicious or legitimate. But these teams face capacity constraints, limiting review to just 1–2% of daily transactions [55]. Expanding capacity is difficult due to the specialized expertise required. Moreover, regulatory penalties and reputational harm can result from failure to detect illicit activity. For example, according to an interview with a model auditor at a major Dutch bank, the institution aims to detect around 8% of illicit activity in its test data, reflecting practical constraints.

1.2 PROBLEM STATEMENT

Although machine learning has become central to AML systems, its effectiveness is limited by the scarcity of labeled data. Supervised models require large annotated datasets, yet banks can only label a small fraction of transactions. This label scarcity hinders model performance and slows iterative refinement.

Active learning offers a solution by selecting the most informative transactions for labeling, referred to as query strategies. However, traditional query strategies focus primarily on statistical performance and offer little interpretability. This lack of transparency reduces analyst trust and complicates regulatory compliance. Providing reasons for which transactions to label is crucial for money laundering detection to support AML teams during their investigations, and an interpretable reasoning why some transactions are picked helps financial institutions during potential model audits. Moreover, incorrect classification does not have the same consequence; false alerts (false positives) burden analyst teams unnecessarily, while missing illicit transactions (false negatives) result in undetected financial crime. Without cost-sensitive optimization, models may misalign with operational goals.

This thesis addresses these challenges by evaluating whether active learning can approximate fully supervised performance with far fewer labeled transactions, by introducing a novel SHAP-based query strategy that identifies unusual transactions in terms of feature importance, and by incorporating a cost-sensitive threshold optimization that reflects the risk preference of the financial institution. These contributions aim to enhance both performance (recall, precision, true negative rate) and real-world deployment under a given labeling budget.

1.3 Research questions

To address these challenges, this thesis investigates the following research questions (RQ):

RQ1. *How does active learning trained on 2.08% of the labeled data compare in performance (recall, precision, and true negative rate) to supervised learning using the complete training set?*

RQ2. *To what extent can an explainability-guided query strategy informed by SHAP values (a widely adopted XAI technique) effectively identify illicit transactions in a synthetic dataset, and how does its performance compare to that of established query strategies in terms of precision, recall and true negative rate?*

RQ3. *How do varying risk preferences, reflected in the cost-sensitive optimization, impact the trade-off between recall, precision, and true negative rate?*

1.4 Proposed contribution

This thesis proposes a comprehensive solution that addresses the research questions and provides two approaches:

1. A novel SHAP-based query strategy that selects transactions whose feature importances differ most from the average legitimate behavior, thereby selecting atypical transactions. This method provides example-specific explanations that help analysts understand why each transaction was prioritized.

2. A net value-based optimization of the classification threshold during training that incorporates both the benefit and cost associated with different detection outcomes (TN, FP, FN, TP). The net value equation consists of the value of correctly identifying illicit transactions, the cost of missing illicit transactions, and the cost of investigating alerts. The benefit parameter b encapsulates the risk preference of the financial institution; higher values reflect a risk-averse stance, while lower values represent a more risk-seeking perspective.

Together, these approaches aim to improve both the detection performance and deployment of machine learning models in money laundering detection, particularly under data scarcity and varying risk-preferences.

1.5 Structure

The remainder of this research is organized as follows. Section 2 provides the impact and structure of money laundering, and the anti-money laundering framework. In Section 3 the most relevant research is presented together. Section 4 describes the dataset and presents an initial analysis. Section 5 details the methodological approach. The results are presented in Section 6. Sections 7 and 8 provide the discussion and a conclusion, respectively. Finally, Section 9 lists the references and Section 10 includes the supplementary materials.

2

BACKGROUND

The concept of money laundering is elegantly captured in a quote from the former president of Mexico in June 2012. In this quote, he summarizes the essential role of money laundering in sustaining organized crime:

> *"Money laundering is giving oxygen to organized crime."*
> — Enrique Peña Nieto

At its core, money laundering refers to disguising the source of money gained illegally. The goal is to make the illicit funds appear as though they were obtained through lawful means, thus enabling criminals to use the money without attracting suspicion. Without the ability to launder money, organized crime would struggle to profit from its operations, effectively suffocating their own operation. Appendix 10.1 provides a short history of money laundering, divided into the key time periods: 1920s, 1980s and present day.

Impact of money laundering The act of laundering money implies that the funds have been obtained through illegal activities such as drug trafficking, human trafficking, fraud, corruption, or terrorist financing. Disguising the origins of illicit funds not only enables criminal enterprises to continue operating but also has far-reaching consequences for global economies, financial institutions, and society as a whole [44, 93]. It distorts fair competition as illicit businesses have hidden financial advantages. Gjoni, Gjoni, and Kora [42] identifies several additional economic impacts. The higher demand and willingness to pay inflated prices for high-value assets distort consumer spending. In addition, money laundering can impact economic growth by diverting funds from legitimate activities to riskier ventures. It can also destabilize industries when launderers abandon businesses that no longer serve their interests. To integrate illicit funds into the economy, money launderers may also bribe accountants, bankers, and lawyers to facilitate their activities. If they gain substantial economic influence, they may even attempt to corrupt law enforcement and lobby political institutions for their benefit.

The $3.1T of laundered money is estimated by Nasdaq and Verafin [69], who developed an in-depth report on global financial crime in 2024. This staggering amount highlights the scale of illicit financial activity and its influence on financial stability. As shown in Table 1, the largest source of illicit funds originates from drug trafficking, followed by human trafficking.

Crime	Illicit funds
Drug trafficking	$782.9B
Human trafficking	$346.7B
Terrorist financing	$11.5B
Other (corruption, fraud, organized crime, etc)	$1.9T
Total	**$3.1T**

Table 1: Sources of estimated illicit funds in 2024.

Stages Financial Action Task Force (FATF) [35] identifies that money laundering is typically a three-stage process, involving the introduction of the proceeds of crime into the economy, the layering of transactions by money launderers to obscure its origins, and the final integration of the funds into the legal economy:

1. Placement: Criminal proceeds are introduced into the financial system, either in cash, through bank accounts, or as virtual currency. Cash is typically handed to collectors who deposit it via cash-intensive businesses, casinos, or cross-border transport. Bank-generated proceeds, such as those from fraud or

tax crimes, may be moved via shell companies with accounts opened specifically for laundering. In the case of virtual currencies, criminals use e-wallets or blockchain addresses to store and initiate the laundering process.

2. Layering: The aim is to disguise the origin of the funds through a complex network of transactions. Money launderers coordinate the movement of funds using trade-based money laundering, fictitious trade, underground banking, or shell companies. Funds from multiple clients are often mixed in the same accounts, and funds are spread across multiple accounts making tracing the origin difficult. In the case of virtual currencies, funds pass through chains of wallets and money mule networks, increasing opacity. An example of a money mule network is the network of money brokers in the Black Market Peso Exchange mentioned in Appendix 10.1.

3. Integration: Finally, the laundered funds are returned to the client through investments in real estate, luxury goods, businesses, or (cross-border) transactions. These investments serve to legitimize the illicit funds by blending them in with legitimate economic activity, often in foreign jurisdictions or the origin country.

Note that while commonly described in three distinct stages, in practice these stages can intertwine, depending on the sophistication of the laundering scheme.

Anti-money laundering framework Establishing effective standards and guidelines for preventing, detecting, and prosecuting illicit financial activities is essential to mitigate the major impact of money laundering. The general structure of detecting and prosecuting money laundering is illustrated in Figure 1.

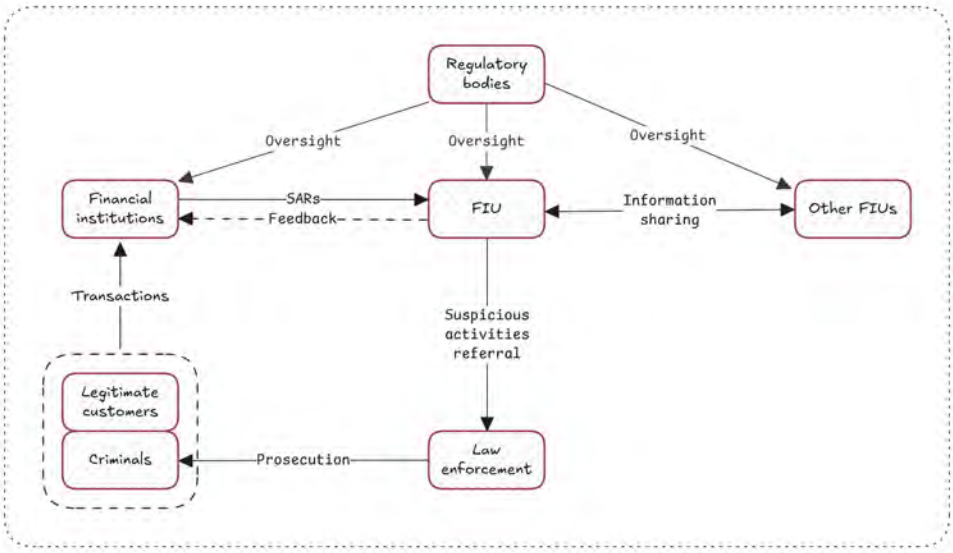


Figure 1: Simplified holistic anti-money laundering framework. Some FIUs do not provide financial institutions with feedback, hence the arrow from FIU to financial institution is dashed.

Anti-money laundering (AML) frameworks have been developed and refined over time at both national and international levels. They operate through the coordinated efforts of various entities, including financial institutions, Financial Intelligence Units (FIUs), regulatory bodies, and law enforcement agencies.

– Financial institutions: These institutions are responsible for transaction monitoring. This includes both pre-transaction, such as checks against sanction lists or politically exposed persons before the transaction goes through, and post-transaction where transactions are analyzed on, for example, the suspicion of money laundering. Analysts in AML teams investigate alerts created by the detection model. If they deem a transaction or a bundle of transactions sufficiently suspicious, they are sent to the national FIU in the form of a Suspicious Activity Report (SAR). These reports include detailed justifications for the suspicion, as well as relevant transaction data and client information.

- Financial Intelligence Units (FIUs): FIUs are national agencies responsible for receiving, analyzing, and disseminating financial information related to potential money laundering activities. FIUs consolidate information from various financial institutions, Open Source Intelligence (OSINT) [73], and sometimes even other FIUs into a single file called a case file. These cases expand with each transaction sent in and are ultimately shared with law enforcement if the suspicion is grounded [36]. Every FIU works with the fundamental condition that it is autonomous and operationally independent.
- Regulatory bodies: These bodies oversee the implementation and enforcement of AML legislation at both national and international levels. Examples of national authorities are the Financial Conduct Authority in the United Kingdom, the Netherlands Authority for the Financial Markets, and the U.S. Securities and Exchange Commission. Their responsibilities include conducting inspections, reviewing the effectiveness of internal controls, and imposing sanctions or penalties for non-compliance. These bodies also issue guidance on best practices, helping institutions improve their AML frameworks over time. The Financial Action Task Force (FATF) sets global standards for combating money laundering and terrorist financing. The Egmont Group, comprising of more than 170 national Financial Intelligence Units (FIUs), facilitates international information exchange and cooperation in financial crime investigations.
- Law enforcement agencies: These agencies are responsible for investigating and prosecuting money laundering offenses. They rely on intelligence provided by FIUs to identify potential criminal networks and trace the movement of illicit funds. Once suspicious patterns are identified, agencies may deploy a variety of investigative tools, including surveillance, undercover operations, and forensic accounting techniques to build legal cases. For cross-border cases, collaboration with international organizations such as Interpol and Europol is essential.

Each entity plays a unique role in identifying, analyzing, and prosecuting financial crimes, but their collective effort is what ensures the integrity of the financial system. Given the global nature of money laundering, cross-border cooperation and data sharing between countries and institutions play a crucial role in detecting and preventing illicit activities. By sharing information, refining detection methods, and enforcing compliance, these organizations form a robust defense against the evolving money laundering techniques.

3

RELATED WORK

This section reviews relevant literature on the key components of our research: machine learning applications in fraud detection, cost-sensitive learning, active learning, and the intersection of active learning and XAI. We also discuss optimization techniques.

Machine learning Numerous studies and surveys [8, 16, 33, 44, 54, 90, 93] have explored the use of machine learning in fraud detection. Traditional supervised models, such as logistic regression, decision trees, and support vector machines, have been widely applied. Ensemble methods like random forests and gradient boosting machines have gained prominence due to their robustness and strong predictive performance. Deep learning models, including autoencoders and convolutional neural networks, show promise in capturing complex and temporal fraud patterns, although their effectiveness is often limited by class imbalance and the scarcity of labeled fraudulent cases.

To address scalability and imbalance challenges, Tertychnyi et al. [89] propose a two-staged architecture combining a lightweight logistic regression filter with a gradient boosting machines classifier, optimizing feature extraction only for high-risk cases. Raghavan and El Gayar [76] conduct a comprehensive benchmarking study of both traditional machine learning and deep learning methods across three financial datasets (European, Australian, and German), evaluating performance using the area under the curve (AUC), Matthews correlation coefficient (MCC), and cost of failure. They emphasize the evolving and unpredictable nature of fraud, which complicates detection despite advances in modeling.

Cost-sensitive learning In fraud detection, the cost of misclassification is highly asymmetric: failing to detect fraudulent transactions can result in substantial regulatory penalty and reputational damage, whereas incorrectly flagging a legitimate transaction leads to operational costs and customer dissatisfaction. Cost-sensitive learning addresses this imbalance by optimizing a loss function that accounts for the costs and benefits associated for different classification outcomes [97].

Several studies have explored cost-sensitive approaches for credit card fraud detection [3, 76, 83], driven by the direct financial impact of fraud, that is money lost on the account. Sahin, Bulkan, and Duman [83] propose a cost-sensitive decision tree that minimizes misclassification costs at each split, outperforming traditional models in terms of accuracy, recall, and a custom cost-sensitive metric. They employ a 5:1 cost ratio, where misclassifying a fraudulent transaction is considered five times more costly than misclassifying a legitimate one. Similarly, Raghavan and El Gayar [76] introduce a realistic cost-based metric, the cost of failure, to evaluate three models for scenarios where the MCC and AUC are similar. They quantify the cost of false negatives (undetected fraud) at $\$1,000$ and false positives (legitimate transactions incorrectly flagged) at $\$100$.

Although cost-sensitive learning has been extensively studied in the context of credit card fraud, it is rarely applied in the domain of money laundering detection. One notable exception is Tertychnyi et al. [89], who penalize misclassification of customers previously reported to the FIU more heavily than others, although the specific cost ratios are not published. Thus, despite its practical importances, cost-sensitive learning remains underutilized in AML systems, largely due to the difficulty of precisely defining the cost of misclassifying illicit transactions, which is often depended on the risk preference of the financial institution. This thesis addresses that limitation by adopting an optimization approach to maximize the net value, allowing institutions to encode their risk preferences directly into the learning process.

Active learning Active learning is a form of supervised learning that improves model efficiency by iteratively selecting the most informative instances from an unlabeled pool $\mathscr{U}$ for expert labeling. The query strategy

selects a subset $\mathscr{Q} \subset \mathscr{U}$, which is labeled and added to the labeled set $\mathscr{L}$. The model is retrained on $\mathscr{L}$, and this process is repeated across iterations.

Active learning is particularly valuable in domains where labeled data is scarce or expensive to obtain. In fraud detection, where annotation requires domain expertise and fraud patterns are rare, active learning can help prioritize samples that contribute most to model improvement [19, 49, 55]. Cunha et al. [19] evaluate unsupervised anomaly detectors, such as isolation forest and elliptic envelope, as initial selectors in cryptocurrency markets under cold-start conditions (i.e., no initially labeled data). They find that these methods often fail to provide informative fraud examples, highlighting the challenge of distinguishing illicit activity from legitimate yet unusual behavior. Karlos et al. [49] apply active learning to the detection of financial statement fraud and report improved performance over standard supervised learning. Labanca et al. [55] propose novel query strategies that outperform traditional sampling techniques in a synthetic money laundering dataset, demonstrating the potential of active learning for label-efficient fraud detection.

Active learning and XAI Although many studies use XAI to interpret model predictions in the active learning pipeline [40, 50, 75], few incorporate explainability into the sample selection process itself, which we refer to as explainability-guided active learning.

Luo et al. [64] propose a framework in which ChatGPT-generated explanations are combined with model uncertainty to rank samples for labeling. Križnar et al. [53] introduce GradCAM$_{\text{avg}}$, a query strategy that selects image samples most dissimilar to the average GradCAM activation of previously labeled data using the structural similarity index (SSIM). However, they report that this approach underperforms compared to uncertainty and random sampling, suggesting that interpretability signals alone may not suffice as a selection criterion. This insight underscores the need for further research into how XAI signals can be effectively leveraged in active learning. Our proposed SHAP-guided strategy (detailed in Section 5.6) builds on this principle by computing dissimilarity between SHAP explanations for tabular financial data, rather than images.

Optimization techniques Hyperparameter optimization plays a critical role in maximizing model performance. Grid search, which exhaustively explores parameter combinations, is commonly used but becomes inefficient in high-dimensional search spaces [96]. Random search improves efficiency by sampling configurations at random, often achieving better coverage with fewer evaluations [6].

A more advanced approach is the Tree-structured Parzen Estimator (TPE), a Bayesian optimization algorithm that searches the performance space and selects configurations based on expected improvement [7]. By focusing on promising regions of the search space, it selects new parameters that are more likely to improve performance, making the search more efficient than random or grid search.

Despite substantial progress in machine learning, active learning, and explainability, limited research has explored the integration of explainability-guided query strategies into active learning, especially for tabular financial datasets and fraud detection. Cost-sensitive learning in money laundering detection remains similarly unexplored. This thesis aims to bridge this gap by proposing and evaluating a novel SHAP-based query strategy for explainability-guided transaction selection, in addition to adding a cost-sensitive threshold optimization, which introduces a risk preference framework.

4

DATA

Research on money laundering detection faces significant challenges due to the lack of publicly available datasets, as privacy regulations prevent institutions from sharing their data [48]. Moreover, anonymized data sharing is rare, making it difficult for researchers to access real-world transaction data. As a result, some researchers partner with financial institutions to gain access to this data [9, 26, 39]. When collaboration with an institution is not feasible, synthetic data offers a viable solution. However, simulating realistic financial flows in an economy is complex, as it heavily depends on modeling choices.

For this research, the synthetic data simulator AMLworld was selected. AMLworld was developed by Altman et al. [2] with three key motivations in mind:

1. Banks have a limited perspective, seeing only their own transactions without visibility into those at other institutions.
2. Ground-truth labels in real-world datasets are often incomplete, leading to undetected illicit transactions and false negatives.
3. Identifying intricate money laundering patterns in real data is a difficult task for individual banks without additional resources.

AMLworld is a multi-agent virtual environment that simulates financial transactions, including illicit activities. A substantial amount of research has been conducted using AMLworld [25, 27, 59, 63, 87, 86]. The agents in this world represent entities such as households, companies, and, most importantly, criminals. The behavior of agents is based on real-world data, such as the number of annual transactions per account from the U.S. Federal Reserve [51] and the frequency of different payment formats based on Federal Reserve statistics [43]. The simulator includes various currencies and payment formats, creating a diverse set of attributes for analysis.

Criminal agents within AMLworld engage in money laundering, and the simulator tracks illicit funds by assigning laundering tags to fund during the placement stage. This allows for detailed tracking of illicit money, which is impossible to achieve in real-world scenarios. The illicit funds originate from various sources, such as extortion, loan sharking, gambling, prostitution, kidnapping, robbery, embezzlement, drugs, and smuggling. Once placed in the system, these funds are layered through layering attacks, which are controlled by the criminals, who also decide when the money is integrated into the economy.

The layering patterns are based on Suzumura and Kanezashi [88] which defined them for a different AML transaction simulator, called AMLSim. These layering patterns are closely related to the patterns in Egressy et al. [27] and include common real-world laundering techniques. The patterns are illustrated in Figure 2.

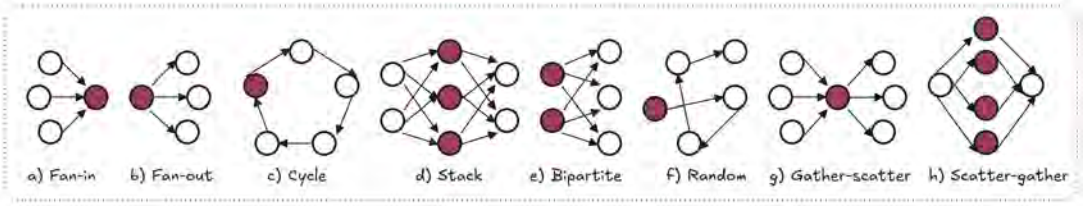


Figure 2: Layering patterns illustrated (Altman et al. [2])

In Chapter 4, Section 4.1 provides an explanation of the raw synthetic data. Section 4.2 discusses the steps taken to prepare the data for exploration. The prepared data is explored in Section 4.3. Finally, Section 4.4 highlights the most valuable insights.

4.1 DATASET

The authors of AMLworld have created six dataset, segmented into a small, medium, or large number of transactions, and low (LI) or high (HI) illicit activity. LI - small, the small dataset with low illicit activity is selected. The LI - small dataset strikes a balance by containing a manageable number of transactions while still including enough instances of money laundering to allow for effective model training and testing. The small dataset (6 million transactions) is selected primarily due to the computational intensity of the larger datasets (medium with 32 million and large with 180 million transactions). Additionally, a smaller dataset with low illicit activity reduces the risk of overfitting to fraudulent cases, which might result in a more generalizable and robust detection model. The features of LI - small are shown in Table 2 and the first three rows are displayed in Table 3.

Feature	Description
Timestamp	Moment at which the transaction was approved.
From Bank	Unique identifier for the originating bank of the transaction.
Account	Unique account number from which the transaction originated.
To Bank	Unique identifier for the destination bank receiving the transaction.
Account.1	Unique account number receiving the transaction.
Amount Received	Amount credited to the recipient in the receiving currency.
Receiving Currency	Currency in which the amount was received.
Amount Paid	Amount debited from the sender in the payment currency.
Payment Currency	Currency in which the amount was paid.
Payment Format	Method used to process the transaction (e.g., ACH, Cheque).
Is Laundering	Indicator of whether the transaction is part of a money laundering attempt (= 1) or not (= 0).

Table 2: Description of features of unprocessed AMLworld

Timestamp	From Bank	Paying Account	To Bank	Receiving Account	Amount Received	Receiving Currency	Amount Paid	Payment Currency	Payment Format	Is Laundering
2022/09/01 00:08	11	8000ECA90	11	8000ECA90	3195403.00	US Dollar	3195403.00	US Dollar	Reinvestment	0
2022/09/01 00:21	3402	80021DAD0	3402	80021DAD0	1858.96	US Dollar	1858.96	US Dollar	Reinvestment	0
2022/09/01 00:00	11	8000ECA90	1120	8006AA910	592571.00	US Dollar	592571.00	US Dollar	Cheque	0
...	...	...	...	...	...	...	...	...	...	...

Table 3: First three lines of unprocessed AMLworld dataset

Besides the transaction data, the authors of AMLworld also provided a text file, referred to as the pattern dataset, containing the layering attacks. This file lists which money illicit transactions are involved in one of the established 8 layering patterns shown in Figure 2. The first two layering attacks are presented in Table 4.

Timestamp	From Bank	Paying Account	To Bank	Receiving Account	Amount Received	Receiving Currency	Amount Paid	Payment Currency	Payment Format	Is Laundering
BEGIN LAUNDERING ATTEMPT - FAN-IN: Max 3-degree Fan-In										
2022/09/01 02:38	001812	80279F810	0110	8000A94C0	10154.74	Australian Dollar	10154.74	Australian Dollar	ACH	1
2022/09/02 14:36	022595	80279F8B0	0110	8000A94C0	5326.79	Australian Dollar	5326.79	Australian Dollar	ACH	1
2022/09/03 14:09	001120	800E36A50	0110	8000A94C0	4634.81	Australian Dollar	4634.81	Australian Dollar	ACH	1
END LAUNDERING ATTEMPT - FAN-IN										
BEGIN LAUNDERING ATTEMPT - FAN-IN: Max 8-degree Fan-In										
2022/09/01 03:17	003671	801BF8E70	002557	8016B3750	8099.96	Euro	8099.96	Euro	ACH	1
2022/09/01 06:27	015	80074C7E0	002557	8016B3750	10468.56	Euro	10468.56	Euro	ACH	1
2022/09/01 10:04	002557	80107C9A0	002557	8016B3750	10270.07	Euro	10270.07	Euro	ACH	1
2022/09/02 06:35	012	800A9B180	002557	8016B3750	15645.21	Euro	15645.21	Euro	ACH	1
2022/09/03 09:12	021393	801271170	002557	8016B3750	14139.75	Euro	14139.75	Euro	ACH	1
2022/09/03 13:45	002175	801E25F20	002557	8016B3750	6276.26	Euro	6276.26	Euro	ACH	1
2022/09/03 16:17	020	800043BE0	002557	8016B3750	1042.63	Euro	1042.63	Euro	ACH	1
2022/09/03 23:09	022124	8011D0180	002557	8016B3750	12795.57	Euro	12795.57	Euro	ACH	1
END LAUNDERING ATTEMPT - FAN-IN										

Table 4: First two layering attacks simulated in AMLworld

4.2 DATA PREPARATION

In this section the AMLworld data is cleaned. Clean data, where each variable conveys a meaningful and objective piece of information, is paramount to an effective and interpretable machine learning model. It generally increases model performance, by mitigating the curse of dimensionality and noise reduction, and allows for detection for biases and anomalies during training and deployment. The data is cleaned up in the following way:

- Merging: The pattern dataset merged into the transaction dataset.[1] This ensures that during data exploration and performance validation the pattern types can be investigated.
- Missing values: Merging the two datasets creates missing values because only some illicit transactions are involved in a layering attack. These missing values are filled with empty strings.
- Duplicates: Duplicates are removed as they are unrealistic; multiple transactions at the same date and time with the same transaction amount are probably the result of a modeling error during the synthetic data generation.
- Self-loops: In the dataset, there are accounts that send money to themselves. An example is shown in Table 5. While this could reflect legitimate behavior, such as transferring money from a checking to a savings account, there is no corresponding transaction indicating that the account was debited. As a result, it appears as though money is being created from nothing. This is likely due to a modeling error, so these self-loop transactions are excluded from the dataset, removing around 800,000 entries, of which 3 are illicit.

Datetime	Paying Bank	Paying Account	Receiving Bank	Receiving Account	Amount (EUR)
2022-09-01 00:08:00	11	8000ECA90	11	8000ECA90	2.726861e+06
2022-09-01 00:07:00	11	8000ECA90	11	8000ECA90	1.960191e+01
2022-09-01 00:10:00	11	8000ECA90	11	8000ECA90	2.561314e+03
2022-09-01 00:16:00	11	8000ECA90	11	8000ECA90	3.125366e+03
2022-09-02 14:55:00	11	8000ECA90	11	8000ECA90	7.670815e+04

Table 5: Example of self-loops in data.

- Currency conversion: The transaction discrepancy between the money sent and money received for transactions with the same currency is zero everywhere (AMOUNT RECEIVED == AMOUNT PAID for each intra-currency transaction). This indicates that there are no additional surcharges set by the bank for intercurrency payments. Therefore it is assumed that the difference between money sent and money received for different currencies is solely the exchange rate. Figure 3 shows the coefficient of variation (CV), or normalized standard deviation, of the exchange rates of the 25 currency pairs with the highest CV. One can see that the CV is non-zero for some currency pairs, meaning that the exchange rate changes over time. Despite that, due the low CV mean across all currencies of 0.015 and no outliers for the top currency pairs we will fix the exchange rate to its mean. This logic is necessary to standardize all amounts to a single currency, selected to be Euros. The information of the currencies of both parties is kept, only the amount is standardized to ensure fair comparison.

[1] There is no unique identifier to track specific transactions, so the merge is done on a match between the pattern and transaction dataset on features: DATETIME, PAYING BANK, PAYING ACCOUNT, RECEIVING BANK, RECEIVING ACCOUNT.

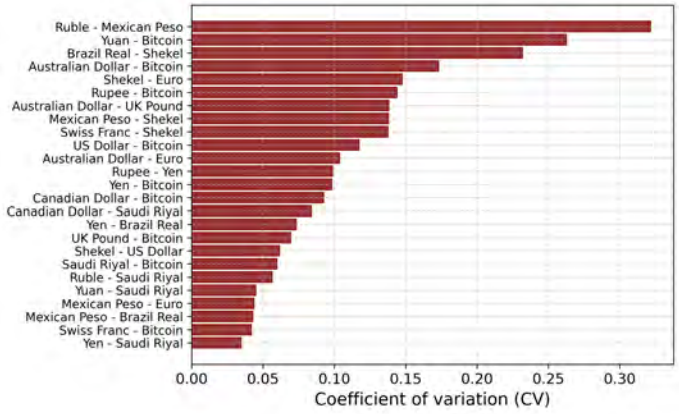


Figure 3: Coefficient of variation for the top 25 currency pairs. The mean over all 210 currency pairs is 0.015.

4.3 Data exploration

The exploratory analysis is structured in four levels: banks, accounts, transactions, and layering patterns. Investigations on bank- and account-level revealed no clear signals or features indicative of laundering activity, and are therefore kept in Appendix 10.2 for completeness. The analysis at the transaction and pattern levels yielded more informative insights, which will be used for the data preparation in Chapter 5.

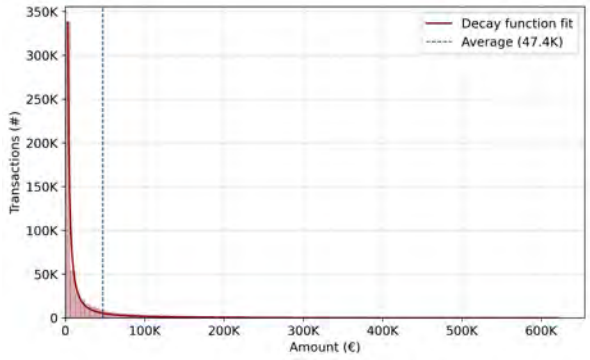
4.3.1 Transactions

The distribution of legitimate and illicit transactions is shown in Table 6. Some of the illicit transactions are part of the layering attacks, which are identified with the pattern dataset. It is assumed that the remaining illicit transactions correspond to either the placement and integration stages, but it remains unclear which of the two stages these illicit transactions belong to, as the dataset does not specify this.

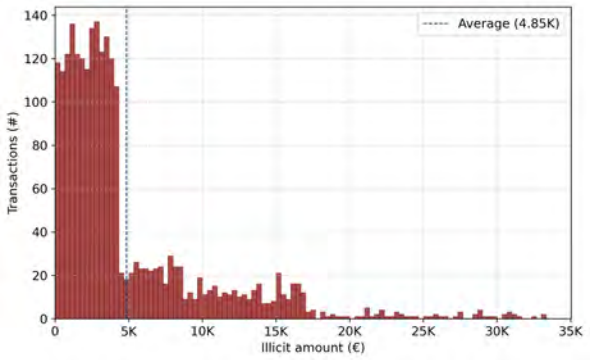
Label	Transactions	Share
Legitimate	6,116,002	99.942%
Money laundering	3,562	0.058%
- Layering	1,109	0.018%
- Placement and integration	2,453	0.040%
Total transactions	**6,119,564**	**100%**

Table 6: Transaction breakdown. The number of placement and integration is the difference between the total number of illicit transactions and the number of illicit transactions involved in a layering attack.

An informative starting point in understanding the dataset on transaction level is the distribution of transaction amounts, both across the entire set of transactions and within the subset of illicit transactions. Figure 4 presents histograms that illustrate these distributions. The histogram in Figure 4a, representing all transactions, follows an expected pattern: lower transaction amounts occur more frequently, and their prevalence gradually declines as the amount increases. In contrast, Figure 4b, which focuses on illicit transactions, reveals a notable concentration of transactions below €5,000. A sharp decline in frequency occurs just before the €5,000 threshold, suggesting that money launderers may intentionally keep transaction amounts below €5,000. However, typically the threshold is not €5,000 but it is set to around €10,000 [37] or $10,000 [48]. Additionally, a second significant drop is observed around €17,500. Similar to the drop near €5,000, this discontinuity also does not align with recognized regulatory thresholds and cannot currently be explained by domain knowledge. Moreover, note that due to the currency conversion described in Section 4.2, the observed thresholds represent approximate values in Euros and are therefore not easily retraced to a recognized threshold in the country representing that currency.



(a) The 90%-quantile histogram of transaction amount, highlighting the concentration of transactions within lower amounts. The overlaid decay function: $\frac{\phi}{\text{Amount}^{\psi}}$ is fitted with parameters $\phi = 72.5B$ and $\psi = 1.53$. The average transaction amount across all transactions is 3.06M.



(b) The 95%-quantile histogram of illicit amount. The average amount across all illicit transactions is 1.95M.

Figure 4: Comparison of transaction amount distributions: (a) overall 90%-quantile distribution and (b) 95%-quantile of illicit transactions. The vertical dashed line represents the average transaction amount of the quantile.

There may be a relation between the illicit activity and time; criminals might launder more money during the day to blend in. Figure 5 shows the hourly total transactions and total illicit transactions over time. Interestingly, on the first hour of the first day 471,153 transactions were processed, significantly higher than the rest of the observed time frame. Afterwards, the total transactions remain stable, with some periodic spikes, which might be due to scheduled payments. It is quite interesting that the total transactions are stable, as it might be a further indication that the banks are located all over the world. First, due to the diversity of currencies that are being traded (shown in Figure 8). Secondly, the total transactions across all banks is stable even though the transaction demand is typically higher during daytime and during the nightcycle of the automated clearing house (ACH) [46], implying that the banks are located in several different timezones. The total illicit transactions shows random behavior, with a sharp decline in total transactions at 11-09, which corresponds to the drop of the total transactions to almost zero. These days with near-zero transactions might be an indication of instability of the synthetic data generator, as the number of transactions should mimic those of a large economy with many banks.

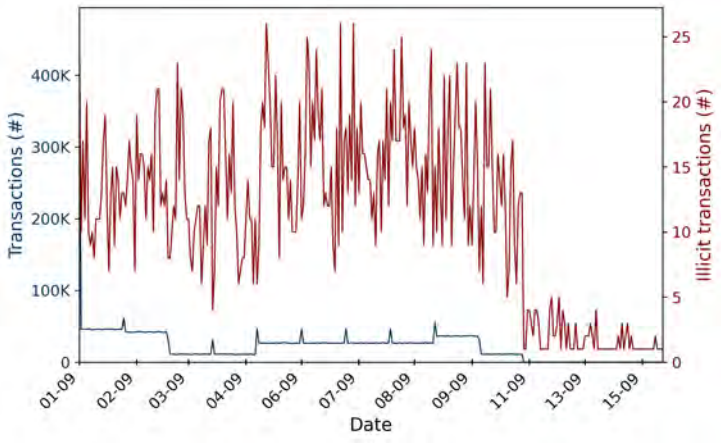


Figure 5: The hourly total transactions (blue) and total illicit transactions (orange).

We do not have access to the true creation dates of accounts, as the synthetic dataset only contains transactions within a fixed time frame (from 01-09 to 15-09) and no customer metadata is provided. Nonetheless, we can still analyze how long each account was active during the observation period. The distribution of active days helps assess whether money laundering is more likely to occur early or late in an account's observed activity history. Figure 6 shows the number of days each account was active, considering both paying and receiving roles. Notably, there is a sharp discrepancy in the laundering rate: transactions involving accounts

active for more than 10 days exhibit an extremely high laundering rate. This phenomenon is likely connected to the reduced number of transactions after 11-09, as shown in Figure 5.[2]

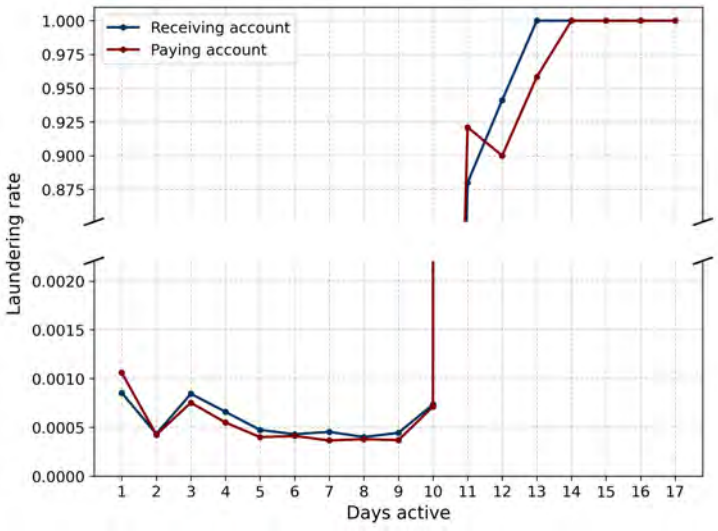


Figure 6: Distribution of the number of days accounts were active.

Figure 7 shows six aggregations of the payment format. Wire and reinvestment do not have illicit transactions with a total of almost million transactions. Automated clearing house (ACH) has the most illicit activity. Card and cash have relatively low illicit amounts compared to their transactions, mainly because the transaction amount per transaction is low.

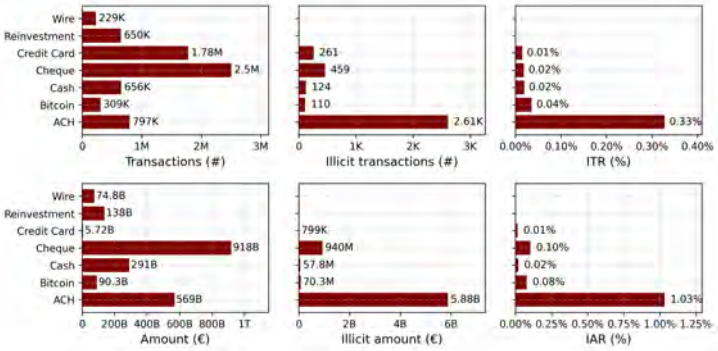


Figure 7: Transaction analysis by payment format. Both for the transactions and amount the total, the illicit transactions and the rate are displayed.

Figure 8 shows the non-zero illicit transaction rates for each currency pair. Surprisingly, no illicit activity has been observed between different currencies, which is unexpected given that money laundering often occurs cross-country and involves the exchange of different currencies to obscure the origin of illicit funds [1]. This suggests that the dataset may not fully capture the cross-border money laundering activities. Furthermore, Figure 8 shows that the majority of illicit transactions are between US Dollar - US Dollar and Euro - Euro, indicating that the criminals are active on these currencies. The sharp drop-off just before €5K and €17.5K in Figure 4b becomes more substantiated given that many transactions are in euros or dollars, and the exchange rate between the Euro and US Dollar is close to 1. However, since these thresholds are not employed in real-world regulation (and may vary significantly across jurisdictions), the pattern of the threshold will not be used during feature engineering. Information about the currencies of the transaction is kept for further exploitation.

[2]Accounts that are active for more than 10 days must have participated in transactions on or after 11-09, since the dataset begins on 01-09.

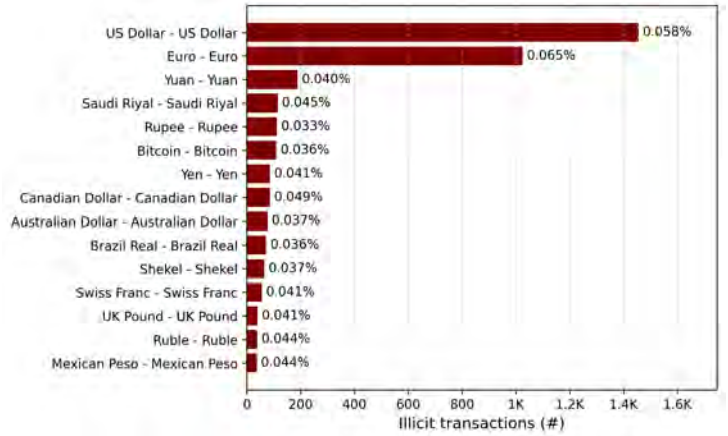


Figure 8: The non-zero total illicit transactions for each currency pair, with the illicit transactions rate (Equation 22) indicated as a percentage label next to each bar.

4.3.2 *Layering*

In the pattern file accompanying the transactions, the authors specify which of the eight layering patterns from Figure 2 each layering attack corresponds to. Figure 9 shows the occurrences of each pattern type separately to get an idea of the shape and frequency of these patterns in the data. It is noteworthy that all pattern attacks are disconnected graphs, whereas in practice money laundering will intertwine multiple pattern attacks to obscure the path further [15]. Looking at the stack and bipartite attacks, it shows that these patterns might not be implemented correctly, as the attacks do not exhibit the same structure that the pattern types of Figure 2 refer to. For example, the layering attacks of 'bipartite' shows that each attack contains transactions from multiple accounts but Figure 9 shows that the transactions are not linked together.

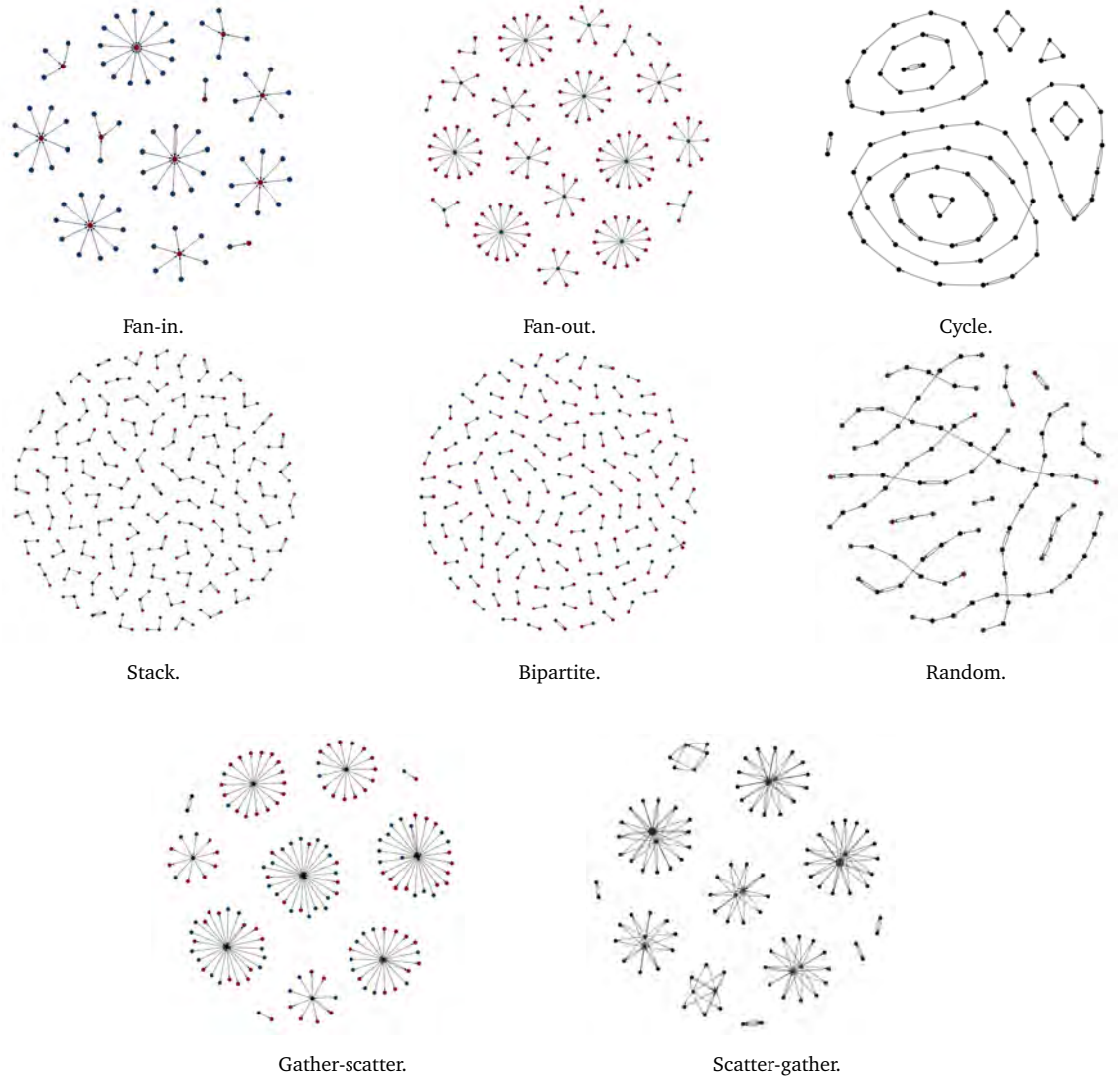
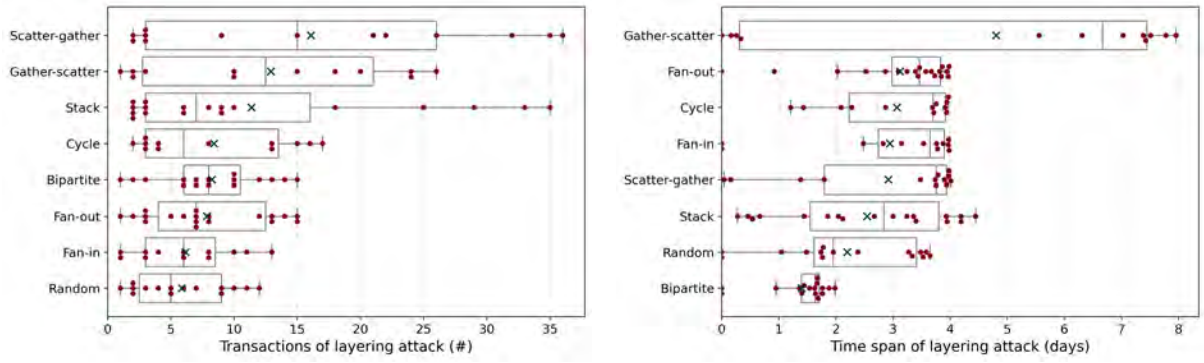


Figure 9: Graph of pattern attacks, where each node represents an account. The blue nodes send money to other nodes, red are on the receiving side and black nodes are both sending and receiving transactions involved in the layering attack.

Layering patterns vary in the number of connected accounts and transactions, as reflected in the neighbor count of the 'gather-scatter' pattern (Figure **??**). Figure 10a provides insight into total transactions per layering attack. The 'scatter-gather' and 'gather-scatter' patterns exhibit high variance and 'stack' has several outliers with high values. The low values in the graph (see the several patterns with a single transaction) suggest that the dataset captures a snapshot in time where some patterns are evolving (instead of each attack being complete), which mimics the real world where it is also unknown when a specific pattern stops evolving. The time frame of the attacks in Figure 10b shows that the majority of the attacks last about four days. The 'gather-scatter' shows high variance, and 'bipartite' has a span of typically 1 to 2 days.



(a) Layering attack sizes.

(b) Layering attack time spans.

Figure 10: Distributions of the layering attack sizes and time spans across the 8 different patterns. The boxplots show the spread and a blue "x" indicates the mean.

The number of layering attacks occurring over time is shown in Figure 11. Each pattern is approximately equally likely, and no relationship is exhibited by the different pattern types.

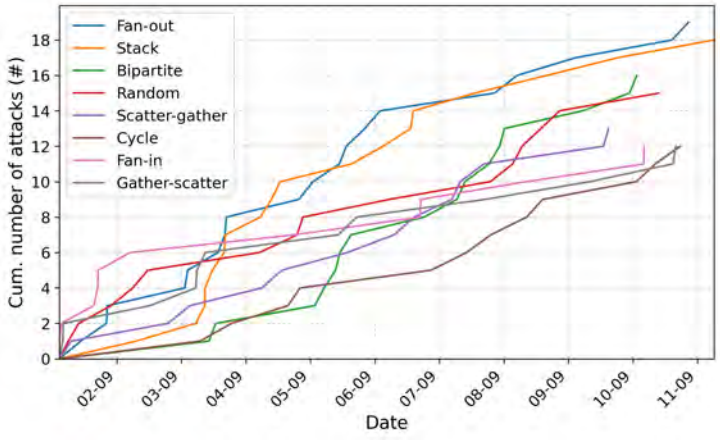


Figure 11: The cumulative number of pattern attacks identified over time. Each line represents a different pattern, with the cumulative count of occurrences increasing the moment a new attack is initiated.

4.4 Summary

The data exploration showed a diverse exposition of information. Key insights, which will be important for the preparation in Section 5, are:

- The transactions are stable over time (Figure 5).
- The number of days the paying and receiving account are active has influence on the laundering rate (Figure 6).
- The payment format has influence on the money laundering rate (Figure 7).
- The currency of the transaction has influence on the laundering rate (Figure 8).
- The layering attacks typically last four days (Figure 10b).

5

METHODOLOGY

This chapter details each method used: the model selection, data preparation, training, and model evaluation. The general model architecture is depicted in Figure 12. The dataset is prepared for input into the model by combining the pattern dataset with the transaction dataset from the AMLworld synthetic data generator, engineering features, and splitting the data into training, validation, and test datasets. The training data forms the basis of the active learning environment and is used as the unlabeled pool. In the first iteration, some instances from the unlabeled pool $\mathscr{U}$ are labeled and added to the labeled pool $\mathscr{L}$. The model is trained on $\mathscr{L}$ for the first time and the classification threshold is optimized. In the following iteration, the model selects new instances to label, based on the a predefined query strategy. These labeled transactions are added to $\mathscr{L}$ and the model is retrained from scratch on $\mathscr{L}$. When the training is finished, top features are selected and the hyperparameters of the model are optimized on the validation set to evaluate the model on the test data.

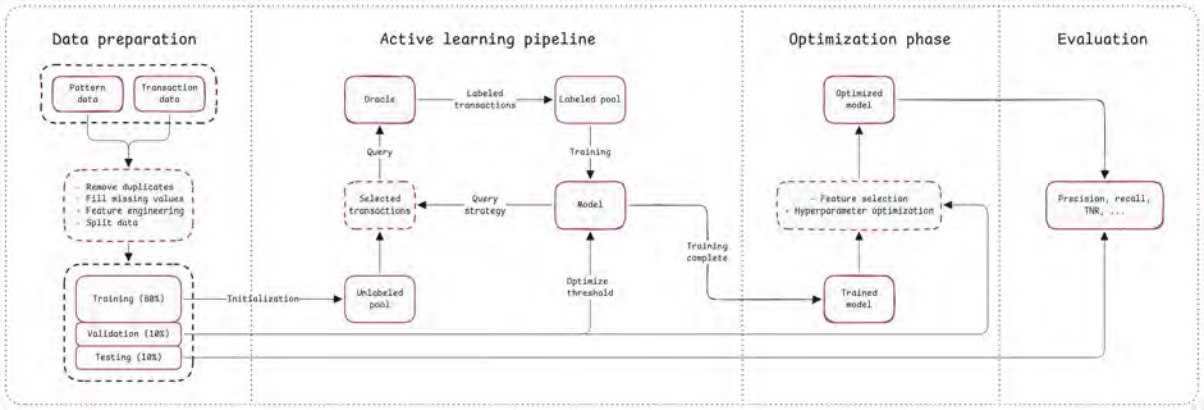


Figure 12: Model architecture.

Chapter 5 starts with the model selection in Section 5.1. Afterwards, the data is discussed in Section 5.2. Section 5.3 describes the training of the model. Section 5.4 describes the evaluation metrics, of which some are used in the optimization, discussed in Section 5.5. Section 5.6 details the query strategies.

5.1 MODEL SELECTION

In active learning, traditional machine learning models such as decision trees and ensemble methods are still widely used, especially when interpretability and training efficiency are priorities. One of the most prominent models in this category is the random forest algorithm, an ensemble learning method based on decision trees that performs well on structured, tabular datasets.

Random forests are well-suited for practical tasks like money laundering detection, where both performance and explainability are important. They are among the best-performing traditional models, as demonstrated in the active learning framework by Labanca et al. [55]. Compared to neural networks, random forests have fewer parameters, require less tuning, and are easier to interpret. Although neural networks build smooth decision boundaries and require lots of tuning, random forests create sharp, step-like boundaries [82]. This helps them to pick up patterns in tabular data more easily [77].

Decision tree A random forest is an ensemble of decision trees. Each decision tree recursively splits the feature space using rules of the form $x_j \leq t$, where x_j is the value of the j-th feature and $t \in \mathbb{R}$ is a threshold. This process forms a binary tree, where each internal node corresponds to a split, and each leaf node represents a prediction.

For classification, the class label of a leaf node is made by majority vote among the training samples that fall into that leaf. Let N_{leaf} be the number of such samples, and let $y_i \in \{1, \ldots, K\}$ be the class label of the i-th sample in that leaf. The predicted class y^{pred} is then:

$$y^{\text{pred}} = \arg\max_k \; p_k = \arg\max_k \; \frac{1}{N_{\text{leaf}}} \sum_{i=1}^{N_{\text{leaf}}} \mathbb{1}_{\{y_i = k\}}, \tag{1}$$

where p_k is the proportion of samples in the leaf that belong to class k, and $\mathbb{1}_{\{y_i=k\}}$ is the indicator function equal to 1 when $y_i = k$, and 0 otherwise.

Random forest. A random forest builds an ensemble of T decision trees, denoted as $\{h_t(\mathbf{x})\}_{t=1}^{T}$. Each tree is trained on a bootstrap sample of the training data, created by randomly drawing N instances with replacement from the original dataset of size N. Furthermore, at each split, a random subset of m features is considered, which introduces randomness and helps to decorrelate the trees, such that each tree makes different errors and learns different patterns.

For classification, the random forest predicts the class $\hat{y}$ of an input $\mathbf{x}$ by taking a majority vote among the predictions of all trees:

$$\hat{y} = \text{mode}\left(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})\right). \tag{2}$$

This ensemble mechanism reduces variance significantly compared to a single decision tree, while maintaining low bias. It is particularly effective in handling high-dimensional, nonlinear, and categorical data with limited preprocessing. One key limitation of random forests in the context of active learning is that they do not support incremental learning. Once trained, the model cannot be updated with new labeled data without retraining the entire forest from scratch. This contrasts with incremental learning models, such as neural networks, which can be refined iteratively with new batches of labeled data. Despite this, their training speed, stability, and high baseline performance make random forests an attractive choice for active learning frameworks.

In summary, random forests provide a strong balance between performance and robustness, making them well-suited for money laundering detection in an active learning setting. Although individual decision trees are easy to interpret, the ensemble of many trees in a random forest can be more challenging to explain. Nonetheless, various tools and techniques exist to improve the interpretability of random forest models. For instance, random forests are not only compatible with the explainable AI (XAI) technique SHAP, but Yang [95] proposed the FastTreeSHAP $v2$ algorithm, which exploits the structure of the tree-based model to optimize the SHAP calculation without sacrificing accuracy.

5.2 Data preparation

The processing of the data requires special attention to feature engineering, class imbalance, and split of the dataset. The feature engineering addresses feature encoding, high-cardinality features, feature addition, and feature scaling.

5.2.1 Low cardinality features

Low cardinality categorical features are commonly encoded using one-hot encoding, which is simple and interpretable. However, as noted by Roam Analytics [79], this encoding introduces sparsity in the feature space for tree-based models, which can bias the tree-building process. Specifically, because one-hot encoding expands a single categorical feature into multiple binary features, continuous variables tend to have more consistent and informative splits. As a result, trees often prefer to split on continuous features in early stages, while one-hot encoded categorical features may be less likely to be selected initially due to their sparse representation. This phenomenon may affect how the model prioritizes features, but does not necessarily degrade overall model performance. The tree can still achieve good predictive accuracy by splitting on relevant features later in the hierarchy. Nonetheless, the reduced early selection of categorical features may impact the importance attributed to these features.

Due to the lack of sufficient alternatives, the features PAYMENT FORMAT and CURRENCY are one-hot encoded. For currency-related data, a single unified feature is constructed by combining PAYING CURRENCY and RECEIVING CURRENCY into a single indicator for each currency type. This binary feature is set to 1 if a given currency appears on both the paying and receiving side of the transaction. This consolidation is chosen due to the significant overlap between the two original currency columns. Encoding them separately would unnecessarily increase dataset sparsity, potentially degrading model performance.

5.2.2 High cardinality features

The AMLworld dataset contains PAYING BANK, RECEIVING BANK, PAYING ACCOUNT, and RECEIVING ACCOUNT, which are features with extremely high cardinality. Collectively, the banks account for 41,815 unique values, and the accounts for 705,903. While high-cardinality features can enhance model performance when handled appropriately [66], their inclusion poses significant challenges in this setting. Carneiro et al. [11] list various techniques to address high cardinality variables in fraud detection, such as target encoding, frequency encoding, and embeddings.

However, in this particular case, these methods are either computationally too expensive, introduce unacceptable risks of overfitting, or do not provide sufficient explanatory power. For example, dummy encoding leads to an unmanageable number of dimensions and sparsity, invoking the 'curse of dimensionality' [5]. Target encoding, while effective, uses the target label multiple times, leading to label leakage and increased overfitting risk. Frequency encoding oversimplifies complex behaviors, potentially grouping entities with similar volumes but vastly different money laundering risk profiles together. Lastly, although embedding layers could provide a more elegant solution, they introduce additional model complexity and require substantial training data to learn useful representations, resources that may not be available or justifiable in this context.

Given these trade-offs, all high-cardinality variables were removed from the dataset.

5.2.3 Feature addition

The insights gathered in the data exploration in Chapter 4 show that apart from the structure of the layering patterns, there is not much separability between legitimate and illicit transactions. To capture these layering patterns, the following features have been constructed based on the pattern architecture shown in Figure 2 and the specific time and size characteristics from Figure 10:

- Fan-in and fan-out: For fan-in and fan-out the number of unique receiving and sending accounts, respectively, of the last four days might be a good indicator of a pattern attack. A time frame of four days is chosen as a fan-in/fan-out attack typically lasts no more than four days. Eddin et al. [24] also implemented these features in their research and called them degree features.

- Gather-scatter and scatter-gather: These patterns involve a sequence of fan-in followed by fan-out (gather-scatter) or fan-out followed by fan-in (scatter-gather). Therefore, it is expected that model can also capture these patterns using the features added for the fan-in and fan-out patterns. Some gather-scatter attacks span more than four days, so additional features that compute the number of unique receiving and sending accounts with an eight-day time frame are included to capture these longer-lasting patterns.

- Bipartite and stack: These patterns are combinations of multiple fan-in and fan-out attacks, based on Figure 2. A stack pattern consists of several fan-out patterns followed by several fan-in patterns, whereas a bipartite pattern is composed of multiple fan-out patterns stacked on top of each other. So again, we expect that the model can capture these patterns with the previously added features. However, as seen by Graphs **??** and **??**, the actual structures in AMLworld do not correspond with Figure 2, indicating that these structures can not be exploited for feature addition.

- Cycle: Cycles typically last no more than four days and show a length of at most 17 steps. While criminals could theoretically create arbitrarily long cycles, the diminishing returns of longer cycles likely discourage them. To balance computational efficiency[3] and practical relevance, cycle detection is limited to a maximum length of 15. A binary feature indicates whether a cycle occurred within 15 steps or not.

[3] Cycle detection is performed using Depth-First Search (DFS), which has a time complexity of $\mathcal{O}$(V+E). In the worst case, the number of nodes explored grows exponentially with the maximum cycle length due to the branching factor of the graph.

– Random: Since random attacks exhibit no discernible pattern by design, no specific feature is designed to capture them.

Criminals tend to operate within networks, so accounts involved in money laundering are likely to have a higher probability of being connected to other accounts engaged in money laundering. Inspired by the GuiltyWalker feature from Eddin et al. [24], a simplified version is created, referred to as CRIMINALS IN NETWORK. The CRIMINALS IN NETWORK feature keeps track of the number of illicit transactions each account has been involved in during training. Specifically, each time an illicit transaction is observed, the count for both the sender and the receiver accounts is incremented by 1. Then, for any new transaction, the feature value is computed by summing the counts for the sender and receiver accounts. For example, the feature value is 1 if only one of the accounts has been involved in a single illicit transaction before. Similarly, the feature value is 2 if both accounts have been part of a single illicit transaction before or one of the two accounts has taken part in 2 illicit transactions. This feature quantifies the combined suspicious activity connected to the accounts involved in the current transaction.

Additionally, DAYS ACTIVE PAYING ACCOUNT and DAYS ACTIVE RECEIVING ACCOUNT are included, features containing the number of days since the paying and receiving account's first transaction. This follows Figure 6 where a large discrepancy in laundering rate is shown between different numbers of days active. An intra-currency transaction flag is also added, which complements the encoding of currency such that all information about the currencies is kept and the importance of a transaction being intra-currency can be measured.

Some researchers introduce aggregations that summarize transaction data by specific fields within defined time windows [10, 55, 24]. These aggregations include metrics such as the sum, mean, median, standard deviation, minimum, maximum, and count of the amount sent or received. However, no noticeable separation between legitimate and illicit transactions based on aggregations. To maintain a small, yet rich feature space, these aggregations are therefore not included in the model.

A complete overview of all features, including their descriptions, types, and pairwise correlations, can be found in Section 10.3 of the appendix.

5.2.4 Feature scaling

Random forests are tree-based models that partition the feature space based on thresholds and thus are inherently insensitive to the scale of input variables. Consequently, they do not require feature scaling, unlike many distance- or gradient-based methods. Similarly, the cosine similarity metric employed in SHAP-guided profiling measures the angle between two vectors and normalizes by their magnitude. As a result, cosine similarity is scale-invariant at the vector level and does not necessitate feature scaling. In contrast, the Mahalanobis distance used by the elliptic envelope strategy is sensitive to feature scales, as it relies on the covariance matrix of the data. While this matrix inherently adjusts for variance and correlation among features, extreme scale differences can destabilize covariance estimation, leading to unreliable distance calculations.

Despite this practical benefit, feature scaling comes with notable downsides. Scaling can reduce the interpretability of features by transforming them, making direct domain understanding more challenging. Additionally, many scaling methods, such as min-max normalization and standardization, are sensitive to outliers, which can distort the scaling parameters. Based on these considerations, we choose not to apply feature scaling. The inherent scale insensitivity of random forests and cosine similarity reduces the necessity for normalization, while the potential downsides, loss of interpretability and sensitivity to outliers, outweigh the benefits in our context.

5.2.5 Class imbalance

In money laundering detection, class imbalance is a significant challenge due to the vast disparity between legitimate and illicit transactions. The minority class (illicit transactions) can be as rare as 1 in 1,000 to 1 in 100,000 instances. Such imbalance often leads to models that struggle to identify illicit activities, resulting in high false alert rates.

Common strategies to address class imbalance include resampling techniques such as oversampling and undersampling. Oversampling methods, like SMOTE, increase the representation of the minority class by generating synthetic samples, but they risk overfitting by repeatedly presenting similar or artificially created instances to the model. As noted by Carvalho, Pinho, and Brás [12], oversampling may cause overfitting while undersampling can discard important data. but this can cause the loss of valuable information. These issues are especially pronounced in active learning contexts, where training occurs on small, imbalanced batches.

Maintaining consistent class proportions across validation and test sets is important for reliable evaluation. When the class distribution matches the target distribution, performance metrics better reflect the model's true effectiveness. This alignment ensures that tuning decisions made on the validation set generalize appropriately to the test set and, ultimately, to deployment scenarios.

During training, instead of relying heavily on resampling, active learning helps to alleviate imbalance by focusing labeling efforts on the most informative or uncertain instances. This targeted selection allows the model to learn efficiently even from highly imbalanced pools. Additionally, balanced class weights are used during training to counteract imbalance by assigning greater importance to errors on minority class samples. This weighting encourages the model to better recognize illicit transactions without altering the underlying class distributions.

5.2.6 Data split

The dataset is first chronologically ordered and then divided in 80% training set, a 10% validation set, and a 10% test set. Chronologically ordering the dataset avoids temporal leakage. Temporal leakage occurs when temporal data is not carefully split, which allows the model to learn information from the future. If the test set contains instances that are earlier in time than the instances that the model has trained on, this effect propagates through to the test set. In these scenarios, it can heavily inflate model performance.

To align the class imbalance of the test set with that of the validation set, a controlled undersampling procedure was applied. Specifically, a number of illicit transactions are removed from the test set and an equal number of legitimate transactions were randomly removed from the validation set.[4] This adjustment ensures the reliability of model evaluation metrics by reducing the disparity in class distributions. Importantly, undersampling was performed instead of oversampling, as with oversampling illicit transactions from the validation set would be used multiple times during optimization, which may result overfitting on those specific illicit transactions.

As shown in Figure 5, almost zero transactions occur after 11/09, which means that training, validation, and testing are primarily conducted on transactions between 01/09 and 11/09. The distribution of the dataset is presented in Table 7.

	Transactions	**Transaction split**	**Illicit transactions**	**Laundering rate**	**Time frame**
Training dataset	4,895,651	80.01%	2,724	0.056%	01-09; 08-09
Validation dataset	611,632	9.995%	257	0.042%	08-09; 09-09
Test dataset	611,633	9.995%	257	0.042%	09-09; 17-09
Total	**6,118,916**	**100%**	**3,238**	**0.053%**	01-09; 17-09

Table 7: Dataset split for model training, validation, and testing. The laundering rate for each dataset is provided to show the equal laundering rate between the validation and test datasets.

5.3 Training procedure

The active learning setup uses an expanding window on the chronologically ordered training set, as outlined in Algorithm 1. In each iteration, a batch of transactions is selected and labeled of size B, and subsequently used to retrain the model. This process ensures that the labeled dataset evolves to include the most informative transactions over time. While the training dataset is incrementally labeled and used for model updates, the validation set is used to optimize the classification threshold and evaluate performance at each iteration.

Whereas some previous research define iterations using fixed temporal windows (e.g., daily batches) [55], our dataset spans only seven days that contain a significant number of transactions (Figure 5). Therefore, we partition the training data into 1% percentiles of transaction volume; each "chunk" consists of 48,956 transactions.

To provide a strong warm start, iteration 1 begins by randomly labeling 4,895 transactions (10%) of Chunk 1 and training the initial model on this subset. In subsequent iterations, a batch of $B = 979$ transactions (2% of each chunk) is selected from the next chronological chunk using a query strategy. For supervised query strategies, predictions (or the prediction probabilities) are required to rank unlabeled instances; in contrast,

[4] The original validation set contained 257 illicit transactions, and the test set contained 581. So 324 illicit transactions were removed from the test set and 324 legitimate transactions were removed from the validation set.

unsupervised strategies are model-independent and do not rely on prediction scores. The model is retrained from scratch at every iteration using all labeled data available at that iteration.

Labanca et al. [55] report that the financial institution they worked with considered it feasible for analysts to manually inspect between 1% and 2% of daily transactions. Based on this, our batch size B is fixed at 2% of each chunk. After labeling 10% of Chunk 1 and 2% of Chunks 2 to 100, approximately 2.08% of the training data has been labeled.

Algorithm 1 Training procedure

1: **Initialize** the unlabeled dataset $\mathcal{U}$ as the entire training dataset, ordered chronologically and split in 100 chunks of 1% of the transactions.
2: **Initialize** the labeled dataset $\mathcal{L} = \emptyset$.
3: **Select** Chunk 1 of $\mathcal{U}$, denoted as $\mathcal{U}_0$ (short-hand notation of $\mathcal{U}_{[0\%,1\%)}$).
4: **Label** 10% of $\mathcal{U}_0$ at random, store it as $\mathcal{L}$.
5: **Train** model M on the labeled dataset $\mathcal{L}$.
6: **for** $i = 1$ to 99 **do**
7: **if** query strategy is supervised **then**
8: **Predict** all transactions in $\mathcal{U}_i$
9: **end if**
10: **Apply** query strategy to select and label B transactions from $\mathcal{U}_i$, denoted as $\mathcal{Q}_i$.
11: **Expand** $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{Q}_i$.
12: **Remove** $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{Q}_i$.
13: **Retrain** model M on the updated labeled dataset $\mathcal{L}$.
14: **Optimize** the classification threshold based on the validation set.
15: **end for**
16: **Return** the trained model M, the unlabeled dataset $\mathcal{U}$, and the labeled dataset $\mathcal{L}$.

The labeled dataset is used for model training, hyperparameter tuning, and final evaluation. The unlabeled data is not used after training. The iterative approach dictated by Algorithm 1 ensures that the model continuously improves by learning from the most informative samples, while adhering to realistic labeling constraints.

5.4 Evaluation approach

Evaluation metrics are essential to assess model performance, serving as aggregated indicators of its effectiveness. Researchers often rely on a handful of well-established metrics, which will be discussed in this section.

In money laundering detection, a true positive (TP) occurs when the model correctly identifies an illicit transaction, while a true negative (TN) refers to correctly classifying a legitimate transaction. In contrast, a false positive (FP) arises when a legitimate transaction is misclassified as illicit, and a false negative (FN) occurs when an illicit transaction is mistakenly classified as legitimate. These four outcomes can be summarized in a confusion matrix, shown in Table 8.

		Predicted	
		Legitimate (negative)	Illicit (positive)
Actual	Legitimate	True Negative (TN)	False Positive (FP)
	Illicit	False Negative (FN)	True Positive (TP)

Table 8: Confusion matrix for money laundering detection.

Several metrics help assess a model's overall effectiveness in classification models, namely accuracy, recall, precision, and the F_1 score are commonly used. However, as Lu and Wang [63] also states, accuracy is not suitable for the evaluation of highly imbalanced data, as the number of FN far outweighs the number of TP. Even though, some research in money laundering still base their key metrics on accuracy [41, 48]. The ROC curve and precision-recall curve are also commonly used to assess model performance [48].

True negative rate The true negative rate (TNR) quantifies the proportion of legitimate transactions correctly classified as legitimate:

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{3}$$

In fraud and money laundering detection, TNR reflects the system's ability to avoid raising false alarms, thus reducing unnecessary manual reviews and preserving operational efficiency. The false positive rate (FPR) is its complement, defined as $1 - \text{TNR}$. Due to the typically high volume of legitimate transactions, the TNR in practice tends to be close to 1, which can mask the impact of even modest increases in FPR.

Recall Also known as the true positive rate (TPR), recall measures the proportion of illicit transactions correctly identified out of all actual illicit transactions:

$$\text{Recall / TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

In the context of money laundering detection, recall is of paramount importance: failing to detect illicit activity undermines the purpose of the system, with potential regulatory, financial, and reputational consequences. Similar to the FPR, the false negative rate (FNR) is defined as $1 - \text{TPR}$, highlighting the complementarity between missed detections and successful ones. However, a high recall alone may come at the cost of increased false positives, making it necessary to consider precision and false positive rates in parallel.

Precision Precision addresses this concern by measuring how many transactions flagged as illicit are genuinely illicit. However, optimizing solely on precision might lead to the model missing a significant portion of illicit transactions. The precision is computed the following:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

F_1 score The F_1 score is a widely used metric to evaluate classification model performance, especially in scenarios where balancing false positives and false negatives is crucial. It represents the harmonic mean of precision and recall, providing a single score that captures both aspects:

$$F_1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}. \tag{6}$$

Although the F_1 score assumes equal importance of precision and recall, the more generalized F_β score allows weighting one more heavily than the other. However, the F_1 score remains a standard baseline metric due to its interpretability and balanced nature. Many recent studies in fraud detection and AML incorporate the F_1 score alongside other metrics such as precision, recall, and accuracy to evaluate model effectiveness [2, 27, 59, 63, 81, 87]. Because the F_1 score offers a balanced combination of precision and recall, it collapses two distinct metrics into a single value, obscuring important trade-offs between them. Moreover, in the context of anti-money laundering, recall is often more critical than precision. For this reason, the more flexible F_β score is preferred, as it allows recall to be weighted more heavily than precision. However, this approach requires careful selection of an appropriate β that reflects the relative importance of recall in the application context, which is outside the scope of this research. In this work, the F_1 and F_β scores are therefore not used.

Precision-recall curve The precision-recall (PR) curve plots the recall on the x-axis and the precision on the y-axis, and shows the outcomes for different classification thresholds. Davis and Goadrich [21] stated in 2006 that for highly skewed data, PR curves provide a more accurate performance evaluation than the ROC curve. This is because the ROC curve can indicate that the algorithm is close to being optimal while that might not be the case. In these situations PR curves show room for improvement, meaning that they provide a more realistic metric. Furthermore, PR curves can accentuate differences between different algorithms better, which ROC curves struggle with. The Area Under the Precision-Recall Curve (AUC-PR) is computed as:

$$\text{AUC-PR} = \int_0^1 \text{Precision}(\text{Recall})\, d\text{Recall}. \tag{7}$$

The AUC-PR of a random classifier is the proportion of illicit transactions in the dataset. A perfect classifier has an AUC-PR equal to 1.

Jaccard index Jaccard [47] independently developed the coefficient of community, which is now considered the Jaccard index. It is commonly used in various fields, including informatics, meteorology, and botany, to evaluate the overlap between two sets of data. This metric is particularly useful for comparing the similarity of selected transaction in the active learning process, where different strategies and configurations may select varying subsets of data. The Jaccard index of set A and set B is computed as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{8}$$

where $|A \cap B|$ is the size of the intersection, representing the number of elements that are common to both sets and $|A \cup B|$ is the size of the union, representing the total number of unique elements in both sets. A Jaccard index value close to 1 suggests that the selected sets share a significant amount of overlap, meaning that the strategies are selecting similar sets of data. In contrast, a value close to 0 indicates that strategies are selecting very different sets of data. The Jaccard index can visualize the similarity between the labeled datasets developed during training of different model configurations. The Jaccard index will also be used to visualize the similarity of the sets of remaining features after feature selection.

5.5 Optimization

To enhance model performance, optimization is applied at multiple stages of model development. During training, the focus is on adjusting the classification threshold to maximize the net value on the validation set. Post-training, feature selection followed by hyperparameter tuning based on the validation set are performed to improve generalizability.

5.5.1 During training

The standard classification threshold of 0.5 is often ineffective in transaction monitoring due to extreme class imbalance. At this threshold, models tend to predict almost all transactions as legitimate, resulting in missed detections. Applying balanced weights to the model mitigates some of this issue, but the threshold must be refined for further model performance. Prior studies, such as Bakhshinejad et al. [4], found improved recall with thresholds as low as 0.32, maintaining a false alert rate below 95%. Similarly, Chen et al. [17] proposed a Recall-First approach, optimizing the threshold to maximize recall at acceptable precision levels.

As banks are driven by the cost of investigating alerts, risk of regulatory penalties and impact on reputation, a more practical setting than solely optimizing on statistical metrics is introduced. Investigating alerts is expensive, and scaling analyst teams is slow due to the deep domain expertise required. To address these trade-offs, a cost-sensitive classification threshold optimization strategy is adopted. After each iteration, the threshold that maximizes the net value is selected:

$$\text{NV} = (b-c)\cdot \text{TP} - c \cdot \text{FP} - (b-c)\cdot \text{FN}, \tag{9}$$

where c is the cost of investigating an alert, and b the monetary value for catching an illicit transaction. The "$(b-c)\cdot$TP" term is the net value of catching illicit transactions, the "$-c\cdot$FP" term represents the costs incurred due to false alerts, and "$-(b-c)\cdot$FN" is the net value from missing illicit transactions. Although b will be varied, the cost of investigating an alert c is set at €31.73, according to a rough approximation detailed in Appendix 10.4.

The optimization is done on the validation set using TPE over 100 trials, with a step size of 0.01 in the interval [0,1]. If multiple thresholds yield the same net value, the largest threshold is selected to reduce the randomness that TPE might otherwise introduce. Note that this optimization only influences query strategies that rely on model predictions. Unsupervised or random strategies remain unaffected. Nevertheless, optimizing the threshold during training is critical for allowing the model to meaningfully identify illicit activity and improve validation and test performance.

Optimizing based on net value This cost-sensitive threshold optimization allows institutions to align model behavior with their specific risk preference and operational constraints. The benefit of a true positive, indicated by b, is inherently difficult to quantify due to the intangible outcomes of detecting illicit transactions. Consequently, we assume that b is institution-dependent. Banks that prioritize operational efficiency may assign a relatively low value to b, reflecting a more risk-seeking profile. In contrast, banks that emphasize

regulatory compliance and reputation management are likely to assign a higher value to b, indicating greater risk aversion.

To illustrate the impact of risk preferences, consider Bank A and Bank B. Bank A assigns a monetary value of $b = €1$ million, while Bank B assigns $b = €1K$. Bank A is risk-averse: their net value, computed using Equation 9, heavily prioritizes catching illicit transactions, even at the cost of generating a high number of false alerts. In contrast, Bank B is risk-seeking, placing more weight on minimizing false alerts than on detecting illicit activity. This preference may be driven by practical budget constraints and limited investigative capacity, as Bank B may not have the resources to handle a large volume of alerts.

5.5.2 Post-training

After training, the model is optimized based on the validation set. The primary goal of post-training optimization is to adapt the model to newly labeled data by increasing generalizability. Starting with feature selection alleviates computational burden as the hyperparameter search is done on fewer features. This is the main reason for performing hyperparameter optimization after feature selection. Note that this choice does come at the risk of prematurely discarding potentially informative features.

Feature selection The dataset consists out of 46 features derived from domain expertise and feature engineering. Although a large feature space can improve performance for the training data, excessive or redundant features risk overfitting, increased computational cost, and reduced interpretability [22]. Feature selection mitigates these issues by retaining only the most informative features.

Adopting the terminology of [71], passive learning refers to the standard supervised learning in which a model is trained on the full labeled dataset. In passive learning, feature selection typically relies on:

1. Filter methods based on statistical criteria (e.g., correlation, mutual information).
2. Recursive feature elimination.
3. Embedded methods for integrating selection into training (e.g., LASSO, tree-based importance).

However, active learning complicates this process. The iterative selection of data to label means early feature selection based on the small, biased sample can prematurely exclude features that become relevant later in training. Moreover, query strategies often rely on model uncertainty or data distribution; altering the feature space mid-process can destabilize these mechanisms. For this reason, feature selection is deferred until after training.

To preserve interpretability, PCA and similar dimensionality reduction methods are avoided. Instead, SHAP is employed on the validation set to quantify feature contributions. The SHAP-based selection after training enhances robustness without compromising the active learning process.

Hyperparameter optimization Following feature selection, hyperparameter optimization is conducted on the reduced feature set by maximizing the net value defined in Equation 9, evaluated on the validation set. To efficiently explore the hyperparameter space, TPE is employed again. However, passive learning imposes a significant computational burden: each hyperparameter configuration requires retraining the model on approximately 5 million transactions. Consequently, the number of optimization trials is constrained to 10. This limited budget has two implications. First, the classification threshold is not included in the optimization process but fixed to the threshold from the final iteration of the training phase. Jointly optimizing the threshold with other parameters under a small number of trails leads to instability in the parameters and unreliable outcomes, as interactions between parameters cannot be properly explored. Second, the search space must be compact to prevent ineffective exploration. A large space relative to the number of trials increases the likelihood of suboptimal configurations due to insufficient sampling.

To ensure a fair comparison, all active learning strategies are subjected to the same optimization constraints: the same number of trials (10), the same hyperparameter ranges, and no re-optimization of thresholds. Only two parameters are tuned, number of estimators (T) and maximum tree depth (D), with their respective ranges and step-size detailed in Table 9. Importantly, optimization is conducted exclusively on the labeled dataset obtained during training. The full active learning process is not repeated per hyperparameter configuration, to avoid overfitting to the validation set.

Hyperparameter	Symbol	Range	Step
Number of estimators	T	[50, 300]	50
Maximum tree depth	D	[2, 16]	2

Table 9: Hyperparameters and their optimization ranges

5.6 Query strategies

The query strategy in active learning determines which instances should be labeled to maximize the model's performance with minimal labeled data. For money laundering detection the instances are the transactions. The choice of instances to label significantly impacts the effectiveness of the learning process. The selection of query strategies is based on Lorenz et al. [62] and Cunha et al. [19]; a randomness-driven baseline, two supervised query strategies (uncertainty sampling and query by committee), and two unsupervised query strategies (isolation forest and elliptic envelope). The novelty of this work is an explainability-guided query strategy based on the explainable AI (XAI) technique SHAP, referred to as SHAP-guided profiling.

An overview of the query strategies considered is provided in Table 10. Following the terminology of Li et al. [57] and Du et al. [23], the query strategies that we use can be broadly categorized by their focus on informativeness or anomalousness. Informativeness-based methods aim to reduce model uncertainty, typically by selecting instances near the decision boundary (uncertainty sampling) or those with high predictive disagreement (query by committee). Most anomalous sampling identifies outliers, i.e. instances that deviate significantly from the bulk of the data, as informative signals, particularly in unsupervised/semi-supervised settings (isolation forest and elliptic envelope). SHAP-guided profiling also falls under the anomaly focused strategies, however, it explicitly leverages model explainability during its selection process.

Query strategy	Supervision type	Selection focus	Selection principle	Dependencies
Random sampling	Unsupervised	Baseline	Uniform random selection	Model-agnostic
Uncertainty sampling	Supervised	Informativeness	Distance to decision threshold	Requires probabilistic model output
Query by committee	Supervised	Informativeness	Disagreement among ensemble members	Ensemble-based classifiers
Isolation forest	Unsupervised	Most anomalous	Anomaly score via isolation paths	Unsupervised tree-based anomaly model
Elliptic envelope	Unsupervised	Most anomalous	Mahalanobis distance from Gaussian fit	Assumes that the data follows a Gaussian distribution
SHAP-guided profiling	Semi-supervised	Most anomalous	Dissimilarity to vector of average legitimate feature importance profile	Requires a SHAP-compatible model

Table 10: Comparison of query strategies used for active learning. Each strategy is categorized by its supervision type, selection focus, principle of selection, and model dependencies.

5.6.1 Random sampling

The baseline query strategy is to randomly select B instances. This approach serves as a control to compare the performance of more targeted strategies to simply random sampling instances.

5.6.2 Uncertainty sampling

Uncertainty sampling selects instances for which the model is most uncertain about its prediction. For random forest classifiers, instead of relying solely on the majority vote (as described in Section 5.1), we estimate the probability that an instance x_j is being labeled as the positive class ($y = 1$) by averaging the predictions across all T trees:

$$p(y = 1 \mid x_j) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(h_t(x_j) = 1), \tag{10}$$

where $h_t(x_j)$ is the prediction of tree t for instance x_j, and $\mathbb{I}$ is the indicator function.

The model's uncertainty is defined as the absolute difference between this estimated prediction probability and the current classification threshold $\tau \in [0, 1]$. For each iteration i, the query strategy selects the B

instances in the i-th chunk $\mathscr{U}_i := \mathscr{U}_{[i\%,(i+1)\%)}$ with the smallest distance to the threshold:

$$\mathscr{Q}_i = \underset{x_j \in \mathscr{U}_i}{\arg\min^B} \left| p(y = 1 \mid x_j) - \tau \right| \tag{11}$$

where $\arg\min^B$ is a practical notation adopted for this research to denote the B elements with the smallest values, and x_j are the individual instances in $\mathscr{U}_i$.

5.6.3 Query by committee

Query by committee, introduced by Seung, Opper, and Sompolinsky [85], selects instances for labeling by identifying those that lead to the most disagreement among a committee of models. For random forests, a common way to measure disagreement is to evaluate the variance of the predicted class labels between the trees in the forest. The disagreement measure $D(x_j)$ can be defined as the variance in the predictions of the committee:

$$D(x_j) = \frac{1}{T} \sum_{t=1}^{T} \left(h_t(x_j) - \frac{1}{T} \sum_{t'=1}^{T} h_{t'}(x_j) \right)^2. \tag{12}$$

This variance reflects the level of disagreement among the trees regarding the classification of instance x_j. The query set $\mathscr{Q}_i$ is then formed by selecting the B instances with the highest disagreement values:

$$\mathscr{Q}_i = \underset{x_j \in \mathscr{U}_i}{\arg\max^B} D(x_j). \tag{13}$$

5.6.4 Isolation forest

The isolation forest selects instances based on their anomaly score, which quantifies how isolated or unusual an instance is compared to the rest of the data [60]. The anomaly score for a given instance x_j, denoted as $s(x_j, n)$, is defined as:

$$s(x_j, n) = 2^{-\frac{E(h(x_j))}{c(n)}}, \tag{14}$$

where $E(h(x_j))$ is the average path length of x_j across all isolation trees, $h(x_j)$ is the number of edges traversed by x_j from the root to a leaf node in a single tree, and $c(n)$ is the average path length of a random binary search tree with n instances. This normalizes the score between different dataset sizes. The top B instances with the highest anomaly scores are then selected:

$$\mathscr{Q}_i = \underset{x_j \in \mathscr{U}_i}{\arg\max^B} s(x_j, n). \tag{15}$$

5.6.5 Elliptic envelope

The elliptic envelope models the central data distribution by fitting a multivariate Gaussian distribution to a subset of data that represents the core structure, typically the labeled dataset in active learning contexts [80]. This fitting estimates a mean vector T_0 and covariance matrix S_0, defining an ellipsoidal boundary that approximates the main data mass. The Mahalanobis distance of an instance x_j, which measures how far the point deviates from this distribution, is calculated as:

$$\mathrm{MD}(x_j) = \sqrt{(x_j - T_0)^T S_0^{-1} (x_j - T_0)}. \tag{16}$$

Instances with the largest Mahalanobis distances are selected:

$$\mathscr{Q}_i = \underset{x_j \in \mathscr{U}_i}{\arg\max^B} \mathrm{MD}(x_j). \tag{17}$$

5.6.6 SHAP-guided profiling

SHAP-guided profiling uses the XAI technique known as SHAP to guide sample selection for labeling. This query strategy based on XAI in the context of money laundering detection is a new strategy, unique to this work. At the start of training, after fitting the model to the initial labeled set, SHAP values are computed for each transaction in the labeled set, forming vectors that capture feature contributions of the transactions.

Because the model is trained on these transactions, the feature importances are highly confident. These feature contributions reflect how much changing a feature impacts the model's predictions and form the basis to develop the profiles. The average vector for the transactions with the legitimate ground-truth $\boldsymbol{\mu}_{\text{legitimate}}$ is a vector that is computed component-wise, referred to as the legitimate profile:

$$\boldsymbol{\mu}_{\text{legitimate}} = \frac{1}{|S_{\text{legitimate}}|} \sum_{\mathbf{x}_i \in S_{\text{legitimate}}} \text{SHAP}(\mathbf{x}_i), \tag{18}$$

where $S_{\text{legitimate}}$ is the set of transactions with ground-truth legitimate, $|S_{\text{legitimate}}|$ the number of transactions in that set, and $\text{SHAP}(\mathbf{x}_i) \in \mathbb{R}^d$ is the SHAP vector of transaction $\mathbf{x}_i$. For each unlabeled transaction $\mathbf{x}_j$ (in the next chunk), the SHAP vector $\mathbf{s}_j$ is compared to the legitimate profile using cosine similarity:

$$\text{Similarity}(\mathbf{s}_j, \boldsymbol{\mu}_{\text{legitimate}}) = \cos(\theta) = \frac{\mathbf{s}_j \cdot \boldsymbol{\mu}_{\text{legitimate}}}{\|\mathbf{s}_j\| \cdot \|\boldsymbol{\mu}_{\text{legitimate}}\|}. \tag{19}$$

SHAP-guided profiling assumes that illicit transactions do not have explanations similar to that of the average legitimate profile. By selecting the B transactions that are least similar to the legitimate profile, outliers to the legitimate class in terms of feature importances are searched:

$$\mathscr{Q}_i = \underset{x_j \in \mathscr{U}_i}{\arg\min^B} \text{Similarity}(\mathbf{s}_j, \boldsymbol{\mu}_{\text{legitimate}}). \tag{20}$$

To incrementally update the legitimate profile, a weighted average of the previous legitimate profile and the SHAP vectors of the newly labeled ground-truth legitimate transactions is computed. Let $N^{(i)}_{\text{legitimate}}$ be the number of ground-truth legitimate transactions labeled up to iteration i, and $S^{(i+1)}_{\text{legitimate}}$ the set of ground-truth legitimate transactions labeled in iteration $i+1$. The updated profile becomes:

$$\boldsymbol{\mu}^{(i+1)}_{\text{legitimate}} = \frac{N^{(i)}_{\text{legitimate}} \cdot \boldsymbol{\mu}^{(i)}_{\text{legitimate}} + \sum_{\mathbf{x}_j \in S^{(i+1)}_{\text{legitimate}}} \text{SHAP}(\mathbf{x}_j)}{N^{(i)}_{\text{legitimate}} + |S^{(i+1)}_{\text{legitimate}}|}. \tag{21}$$

This update reflects both the quantity and characteristics of newly labeled ground-truth legitimate transactions. When few ground-truth legitimate transactions are added, the profile remains similar. In contrast, if many are added, especially with extreme feature importances, the legitimate profile shifts more noticeably.

Computing SHAP values is computationally heavy. Due to this high computational burden, the FastTreeSHAP library is used, which optimizes the SHAP calculation without sacrificing accuracy. Execution time can be further reduced by applying the FastTreeSHAP $v2$ algorithm proposed in [95], which reduces the time complexity from $\mathcal{O}(MTLD^2)$ to $\mathcal{O}(TL2^D D + MTLD)$, with M the number of samples to be explained, T the number of trees, L the maximum number of leafs of any tree, and D the maximum depth of any tree. The computational advantage of algorithm $v2$ is achieved for large M, however simultaneously, D is upper-bounded according to a memory-constraint.[5] Due to the memory constraint and exponential complexity on D, the maximum depth that we consider is 8.

[5] FastTreeSHAP $v2$ is preferred when $M > 2^{D+1}/D$. For example for $D = 8$, this means $M > 64$ transactions. The accompanying memory constraint is $\mathcal{O}(L2^D) <$ memory tolerance [95]. For a completely balanced tree ($L = 2^D$) with maximum depth $D = 8$ the memory tolerance is 0.5GB, however for larger D it is harder to construct a completely balanced tree, implying that $L \ll 2^D$.

6

RESULTS

6.1 COMPUTATIONAL INTENSITY

As the dataset contains a large volume of transactions, loading in the data and the feature engineering require significant memory and processing power. Additionally, training machine learning models is demanding, especially with active learning, where models are trained numerous times for different configurations and predict large volumes of transactions. Moreover, passive learning needs to retrain the model several times on 5 million transactions.

The experiments were conducted on a Lenovo ThinkPad T490s laptop, equipped with Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz 2.11 GHz, and 16 GB RAM. To assess the computational efficiency, Table 11 provides an overview of execution times for key processes.

Process	**Active learning**	**Passive learning**
Data preparation	08h 13m	08h 40m
Training	112h	31m
Validation	32h 27m	25h 16m
— Feature selection	— 7h 28m	— 1h 03m
— Hyperparameter optimization	— 24h 59m	— 24h 13m
Testing	19m	4m
Total	**152h 59m**	**34h 31m**

Table 11: Execution times for key computational processes for the active and passive learning. The data preparation time of passive learning is 27m longer because the CRIMINALS IN NETWORK feature must be pre-computed for all transactions in the training data.

6.2 TRAINING RESULTS

Training is performed across multiple configurations of query strategy and benefit per TP (denoted b). The training data is chronologically split in 100 chunks. As a reminder, the first is labeled for 10% at random, thereafter 2% of the transactions in each chunk is selected according to the query strategy and labeled. After this step in each iteration, the classification threshold is optimized to maximize the net value (Equation 9), where $c = €31.73$ (Appendix 10.4).

6.2.1 Training behavior

We conducted preliminary experiments to determine a suitable domain for the benefit parameter b. These early results revealed that for $b < c$, the classification threshold converges to 1. This behavior is expected, since in this range the term '$-(b-c) \cdot \text{FN}$' in the objective (Equation 9) becomes increasingly positive as false negatives accumulate. In other words, missing illicit transactions imposes a net gain rather than a cost, incentivizing extremely conservative predictions. In contrast, for extremely high values of b (that is, $b >$ €1M), the threshold converges to 0, as the net value is maximized by detecting as many true positives as possible. Based on these insights, we designed the domain of b to span a wide range of plausible benefit values. We began with two low-risk-seeking values: €50 and €100, followed by medium-scale increments:

€500 and €1K. Beyond this, we used steps of €2.5K up to €20K, then switched to €5K intervals up to €50K. Finally, we included €100K, €500K, and €1M to capture behavior under extreme risk aversion.[6]

Figure 13 shows recall, precision, and true negative rate across the six query strategies, evaluated at various b values. These metrics are computed at each training iteration using predictions on the validation set. As expected, recall remains near 0 and true negative rate near 1 throughout training for the extremely risk-seeking scenario ($b = €50$). The opposite occurs for the risk-averse extreme ($b = €1M$), where recall is consistently 1 and true negative rate is 0.

Between these extremes, the trade-offs become more nuanced. Recall gradually increases for $b \leq €1K$, though this trend becomes less visually distinct for larger b due to sharp spikes in recall approaching 1. True negative rate remains relatively stable up to $b = €30K$, after which it drops intermittently to 0. Precision values are generally too small to yield visually meaningful distinctions.

SHAP-guided profiling performs competitively in terms of recall, with the exception of $b = €15K$, €20K, and €50K. Uncertainty sampling performs poorly at several benefit levels, particularly at $b = €30K$ and €40K.

Figure 13 includes the passive learning baseline, also evaluated on the validation set. The performance gap between the passive learning model and the active learning strategies (trained on only 2.08% of the data) is remarkably small across all evaluation metrics. This modest decrease in performance demonstrates the effectiveness of active learning in resource-constrained environments. It also highlights the importance of well-designed query strategies for achieving strong performance with significantly fewer labeled samples. Nonetheless, it should be noted that the performance of the passive baseline could be further improved during feature selection and hyperparameter optimization.

[6] The full set of evaluated benefit values is: €50; €100; €500; €1K; €2.5K; €5K; €7.5K; €10K; €12.5K; €15K; €17.5K; €20K; €25K; €30K; €35K; €40K; €45K; €50K; €100K; €500K; €1M.

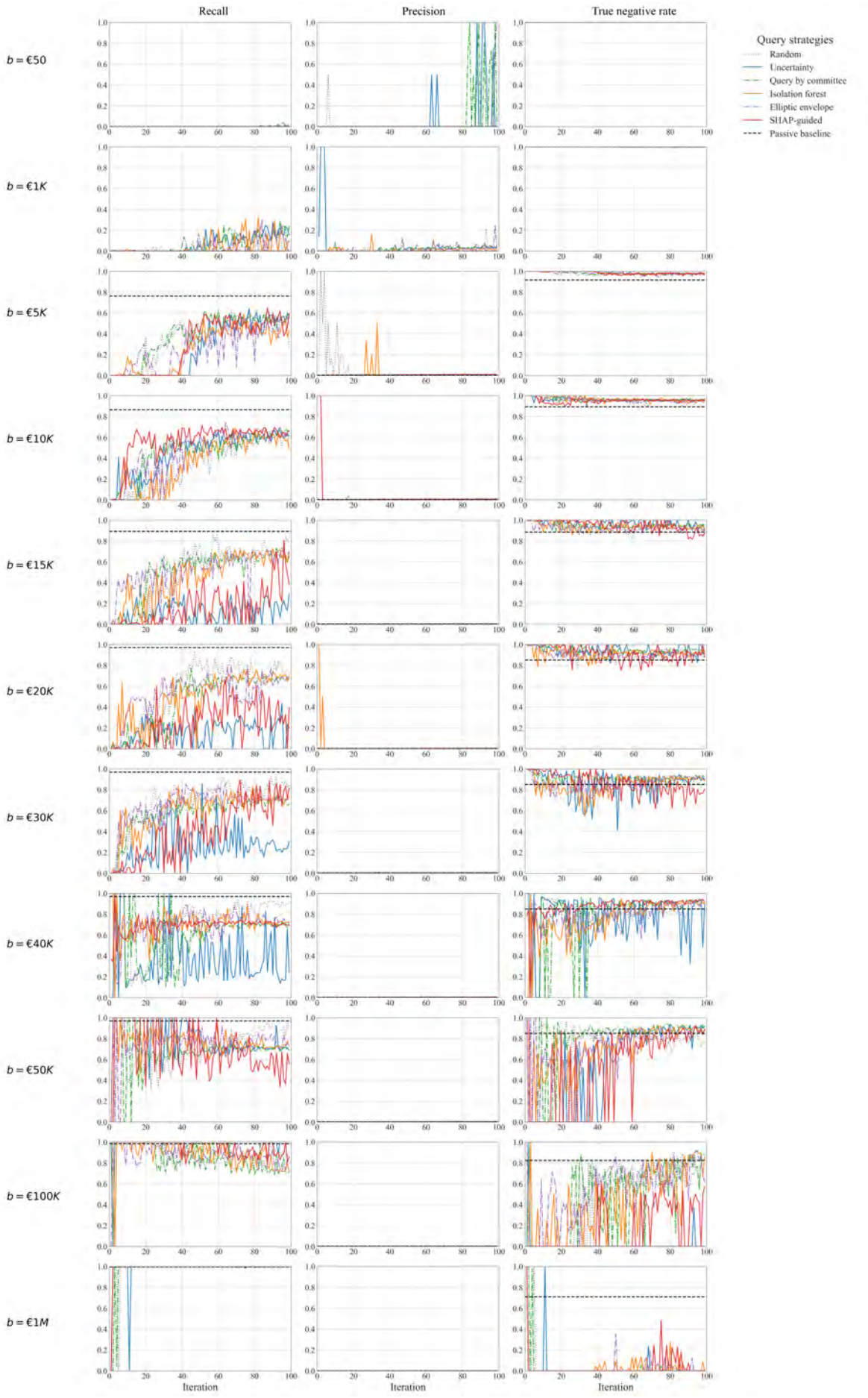


Figure 13: Validation performance metrics (recall, precision, and true negative rate) plotted across different values of benefit per true positive b. Each curve in the plot corresponds to a distinct query strategy. For clarity, only a representative subset of b values is shown.

Similarly to Figure 13, Figure 14 presents the recall, precision, and true negative rate on the validation set across varying benefit values b, this time organized per query strategy. Each line in a plot corresponds to a different value of b. With the exception of the uncertainty and SHAP-guided strategies, most methods exhibit an increasing recall with larger b values, which is visually seen in the gradient of the line colors. The observed spikes in precision across some strategies are largely driven by lower b values, where the threshold optimization becomes unstable due to the prioritization of minimizing false positives.

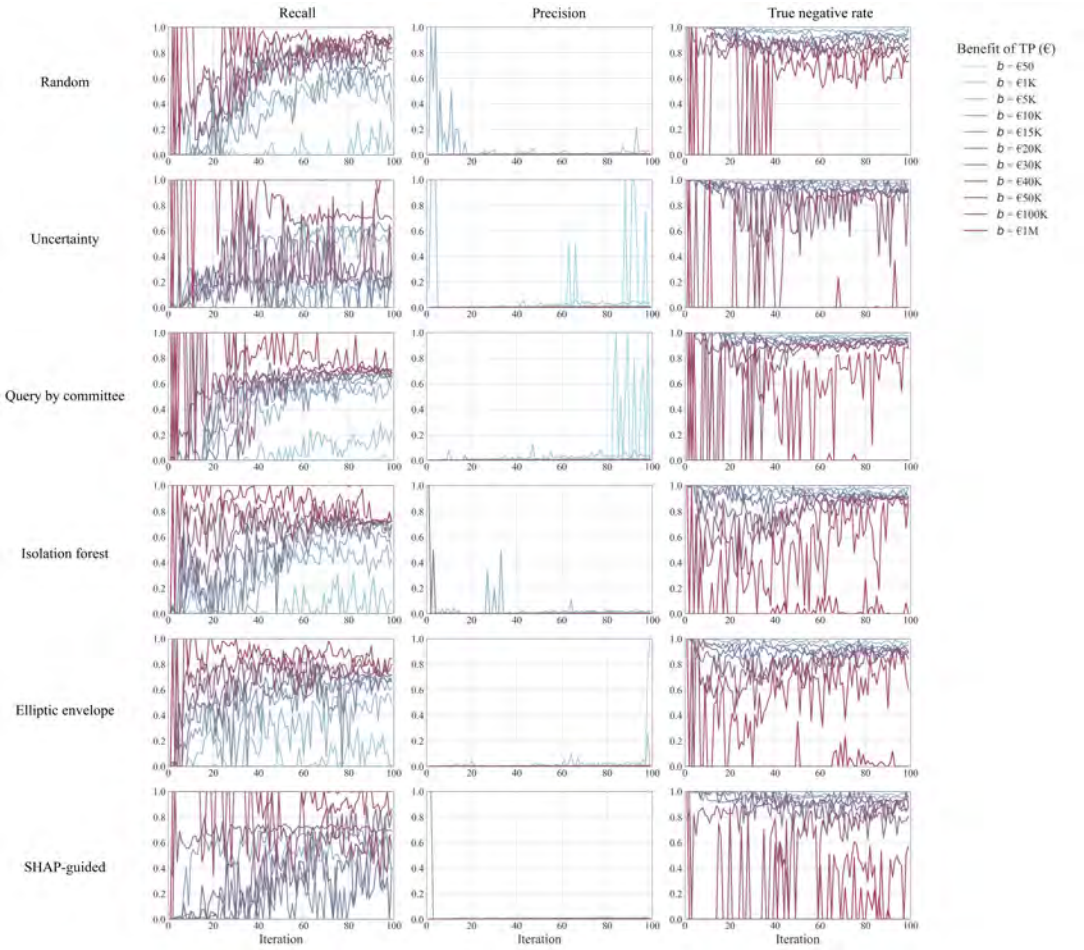


Figure 14: Validation performance (recall, precision, and true negative rate) for each query strategy (rows), evaluated across varying benefit per true positive values (b). Lines represent different values of b, as indicated in the legend. Only a representative subset of b values is shown for clarity.

Figure 15 illustrates the evolution of the classification threshold throughout active learning, optimized at each iteration as described in Section 5.5. The figure confirms that increasing b leads to lower threshold values across all strategies. This behavior reflects the model's prioritization of capturing true positives as the perceived value of detecting illicit transactions increases. Importantly, this threshold adjustment is more interpretable and stable as a result of the use of class-balanced weighting. Without balanced weighting, thresholds would drop toward zero due to the severe class imbalance, complicating optimization and requiring high precision (i.e. very small step-size) in threshold tuning.

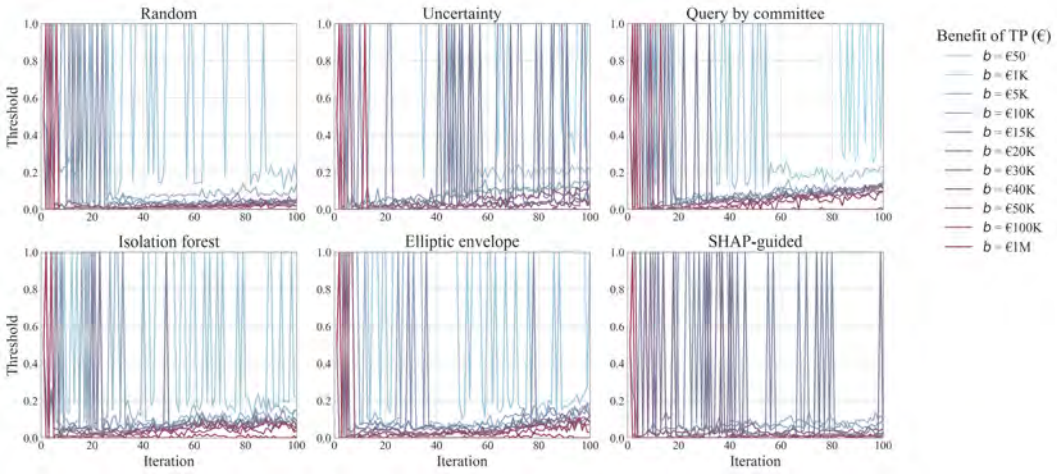


Figure 15: Evolution of the classification threshold across iterations for different values of b across query strategies.

6.2.2 Jaccard similarity

Figure 16 presents the Jaccard similarity index between the labeled training sets (when the training is finished) across all combinations of query strategies and benefit values b. By design, the diagonal elements of the matrix are equal to 1, as they represent comparisons between identical configurations. The off-diagonal values generally lie between 0 and 0.2, indicating minimal overlap between the labeled training sets across different configurations. This low overlap suggests that the specific query strategy and risk preference (reflected by b) strongly influence which transactions are selected for labeling during active learning.

Interestingly, when comparing only within a single query strategy (that is, along the blocks of the matrix corresponding to a fixed strategy), the Jaccard index still varies depending on the value of b. This pattern is especially evident for strategies such as uncertainty sampling and query by committee. In contrast, for the isolation forest strategy, the Jaccard index remains high across all b values, indicating that the same data points are consistently selected. The elliptic envelope strategy exhibits a moderate Jaccard index of around 0.5 across its internal comparisons, revealing a dependence on the specific transactions labeled in the initial chunk. Since the elliptic envelope models the distribution of the initial labeled points and computes distances relative to this distribution, the initial sample has a strong influence on subsequent query selections. This makes the strategy more sensitive to the randomness of the first chunk and introduces greater variation in the labeled datasets.

There is no strong relationship found between different query strategies, so each strategy has their unique way of selecting transactions. However, there is some overlap between combinations of uncertainty sampling and query by committee, both of which utilize disagreement/uncertainty in their selection process. A similar weak relationship is seen between isolation forest and elliptic envelope, which are both unsupervised anomaly detection models.

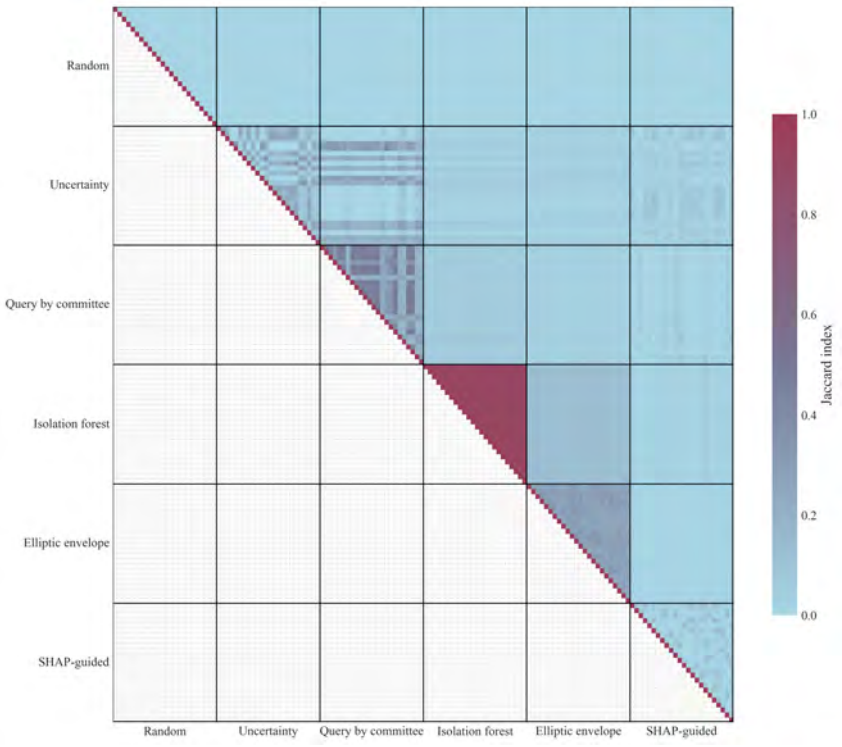


Figure 16: Jaccard index between labeled transaction set of all configurations. The query strategy rectangles are distributed on the grid in the ascending order of the b domain (€50; €100; . . . ; €500K; €1M), starting from the top vertically, and horizontally on the left.

6.2.3 Class Imbalance

Differences in selected transaction sets between configurations (query strategy and benefit) do not necessarily reflect the quality of those selections. To assess how well each configuration identifies illicit transactions, Figure 17 shows the proportion of the minority class (that is, the proportion of illicit transactions in the labeled transactions) among the labeled transactions.

As expected, random sampling results in a class distribution similar to the passive baseline. Uncertainty sampling shows unstable behavior across benefit values. Query by committee stands out with a substantially higher minority class proportion, up to 14 times greater than passive, indicating strong enrichment of illicit transactions. The isolation forest exhibits nearly identical class proportions across benefits due to near-identical labeled sets (see Figure 16). SHAP-guided selection performs marginally better than passive, meaning that SHAP-guided profiling does not substantially improve minority class detection relative to the overall distribution.

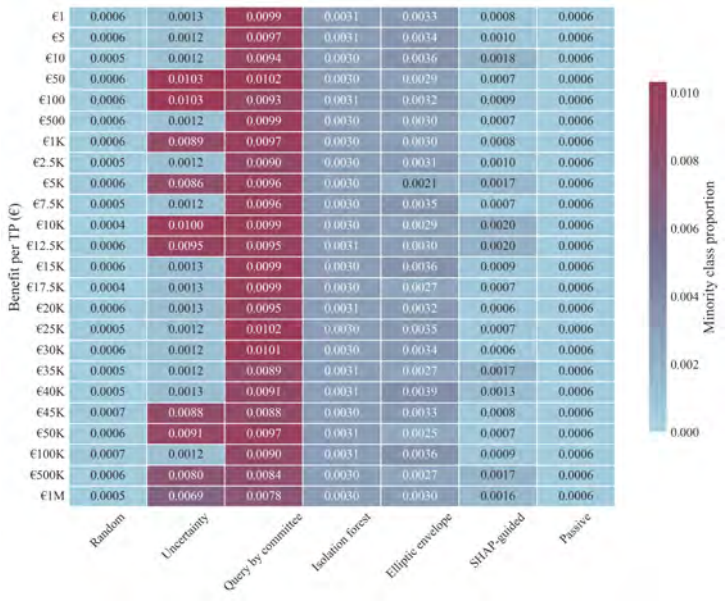


Figure 17: Proportion of the minority class in the labeled data for each combination of query strategy and benefit. The baseline passive strategy is included for comparison.

While the overall class imbalance provides a high-level indication of how effectively each query strategy identifies illicit transactions, it does not reveal which specific behavioral patterns are being prioritized. Figure 18 offers a more granular breakdown by showing the minority class proportion within the labeled data for each laundering pattern type and query strategy. Illicit transactions without an associated pattern are categorized as ”no pattern” and generally correspond to the placement or integration stages of money laundering, as the patterns are associated with the layering stage. Although the minority class proportions shown are averaged over all benefit per TP values, similar trends hold consistently across individual benefit settings.

In the figure, the minority class proportion is calculated as the number of illicit transactions within a given pattern type, divided by the total number of transactions selected for labeling. The ”no pattern” category consistently shows the highest minority class proportion across all query strategies, indicating that a substantial portion of selected illicit transactions fall outside the pattern types. Notably, the high overall class imbalance achieved by query by committee (as shown in Figure 17) is largely driven by its ability to identify illicit cases in the ”no pattern” category. Nevertheless, both uncertainty sampling and query by committee also effectively enrich the labeled dataset with illicit transactions from known laundering patterns. In contrast, random sampling, isolation forest, and SHAP-guided profiling exhibit only marginal improvements in pattern-specific minority class proportions.

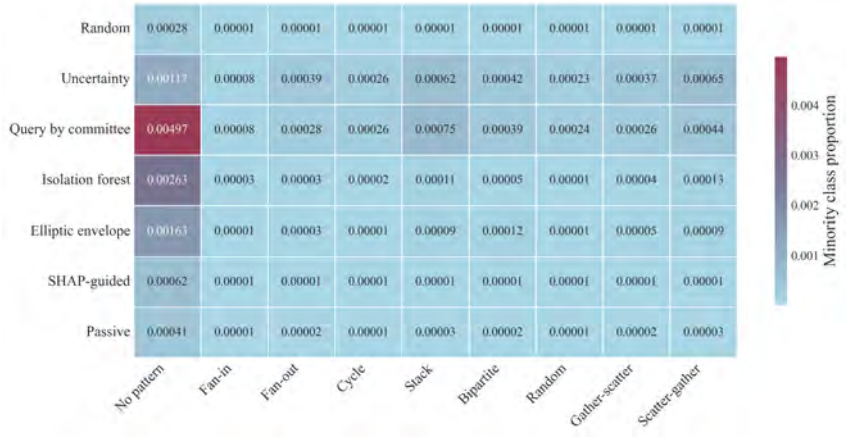


Figure 18: Proportion of the minority class in the labeled data for different pattern types, as well as ”no pattern” and all query strategies.

6.2.4 SHAP-guided profiling

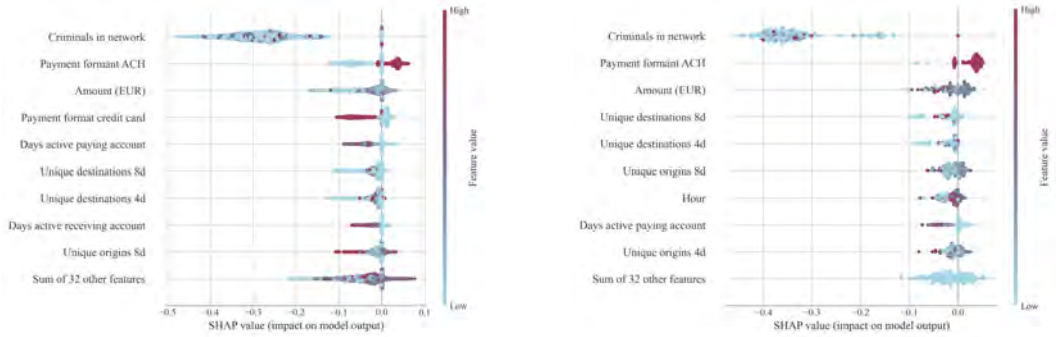
As part of the SHAP-guided profiling strategy, SHAP values are computed each iteration for all transactions in the next chunk. These values quantify the contribution of individual features to the model's prediction, showcasing an interpretable view of the model's decision process. Positive SHAP values push the model's output toward predicting the transactions as illicit. Figure 19 displays a comparative analysis of SHAP values for selected legitimate transactions (left) and selected illicit transactions (right). This visualization reveals how the model relies on different feature sets depending on the transaction class, and offers insights into the model's selection logic with respect to ground-truth labels. Several patterns emerge from this comparison.

First, with respect to overall feature importance: the feature CRIMINALS IN NETWORK consistently shows the highest importance for both types of transactions. The feature HOUR exhibits higher overall importance in illicit transactions than in legitimate ones. Furthermore, the features DAYS ACTIVE (for both the receiving and the paying accounts) demonstrate moderate importance in both cases.

Second, when examining specific feature values in relation to their corresponding SHAP values, several notable trends appear. For illicit transactions, high SHAP values are rarely associated with the feature PAYMENT FORMAT CREDIT CARD, suggesting that such transactions are generally not classified as illicit. Similarly, when the feature PAYMENT FORMAT CHEQUE receives a high SHAP attribution and the transaction is a cheque, it tends to be classified as legitimate. In contrast, ACH transactions tend to be associated with high SHAP values for the feature PAYMENT FORMAT ACH, particularly when the transaction is actually an ACH transactions. These findings are consistent with the empirical distribution of the payment formats presented in Figure 7.

Third, higher transaction amounts are more strongly linked to illicit classifications compared to the legitimate profile, indicating that transaction size is a distinguishing factor, whereas this was not seen during data exploration.

In general, these SHAP profiles enhance the interpretability of the SHAP-based query strategy and serve as a diagnostic tool to evaluate whether the model is learning from informative and representative examples. Differences in feature contributions between classes suggest that the strategy successfully identifies distinct patterns associated with illicit behavior.



Legitimate transactions. Illicit transactions.

Figure 19: Comparison of SHAP profiles for selected legitimate and illicit transactions. Each point represents a selected transaction, with feature contributions visualized as SHAP values. The SHAP-guided profiling query strategy is shown for $b = €10K$.

To illustrate how SHAP-guided profiling differentiates typical (legitimate) from atypical (potentially illicit) transactions, Table 12 presents an explanation report. The transaction with the highest SHAP profile dissimilarity from the legitimate profile at iteration 50 for $b = €10K$ is showcased. The table highlights the 10 features with the largest absolute deviation in SHAP values, calculated as the SHAP value in the legitimate profile minus that of the selected transaction. Negative deviations indicate features that push the model more toward an illicit classification in the transaction, while positive deviations suggest a shift toward legitimacy.

In particular, while the legitimate profile assigns a slightly negative SHAP value to AMOUNT (EUR), indicating low risk, the positive SHAP value of the inspected transaction suggests the amount raises suspicion. A similar pattern emerges for PAYMENT FORMAT CASH, which shifts from a mildly legitimate influence (−0.0116) to one indicating increased illicit risk (+0.0054), resulting in a positive deviation of 0.0171. These changes suggest an atypical use of cash and an unusually small transaction amount relative to the legitimate baseline. In contrast, features such as PAYMENT FORMAT ACH and PAYMENT FORMAT CREDIT CARD show negative SHAP

deviations, indicating that the inspected transaction exhibits stronger illicit signals in these channels. For example, the SHAP value for PAYMENT FORMAT CREDIT CARD drops from −0.0090 to −0.0566, reflecting a higher illicit contribution.

Despite a high overall cosine similarity (0.9449), these SHAP deviations reveal subtle but important divergences from the normative pattern. This contrast enables analysts to pinpoint key behavioral discrepancies, such as anomalous payment channels, timing, or network relationships, that justify further investigation and potentially explain the alert.

Feature	Avg. legitimate	Selected	Feature dev.	SHAP legit.	SHAP selected	SHAP dev.
Payment formant ACH	0.275	0	-0.275	-0.0223	-0.0767	-0.0544
Payment format credit card	0.273	1	0.727	-0.0090	-0.0566	-0.0476
Amount (EUR)	€1,365,379.72	€2871.90	€-1,362,507.82	-0.0340	0.0046	0.0386
Criminals in network	27.243	1	-26.243	-0.2247	-0.2567	-0.0320
Payment format cash	0.262	0	-0.262	-0.0116	0.0054	0.0171
Days active paying account	1.609	5	3.391	-0.0073	-0.0228	-0.0155
Days active receiving account	1.677	5	3.323	-0.0060	-0.0188	-0.0128
Weekday	3.868	0	-3.868	-0.0013	-0.0136	-0.0123
Dawn	0.384	1	0.616	-0.0018	-0.0120	-0.0102
Unique origins 8d	4.904	3.000	-1.904	-0.0087	0.0012	0.0099

Table 12: Explanation report. Top 10 features with the largest SHAP deviation between the legitimate profile and the most dissimilar transaction (cosine similarity = 0.9449). Deviation is defined as: legitimate SHAP – transaction SHAP. Negative values indicate stronger illicit contribution in the transaction compared to the legitimate profile.

6.3 VALIDATION RESULTS

Feature selection is performed to select important features, and then the hyperparameters are optimized. Both procedures are conducted on the labeled transactions and are evaluated using the validation set.

6.3.1 Feature selection

Figure 20 shows the mean absolute SHAP value for all features across all configurations, providing global insight into overall feature importance prior to feature selection.

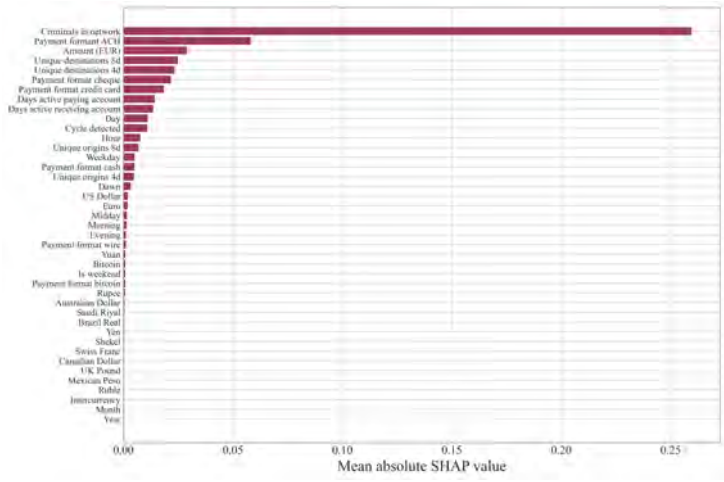


Figure 20: Mean absolute SHAP value across all query strategies and benefit per TP combinations.

Feature selection is applied to reduce the risk of overfitting on the validation set. As shown in Figure 20, the lower half of features contribute minimally to the model's predictions. As a result, only the 20 features with the highest average importance are included in the model. This approach not only reduces noise, but also highlights consistently important features across configurations. The ordering of the features in Figure 20 represents a global average, but specific rankings vary by configuration.

To assess the consistency of selected features across configurations, the Jaccard index is computed. As shown in Figure 21, the selected feature sets are largely consistent, although there is some variation depend-

ing on the query strategy and b configuration. These differences occur primarily among the lower-ranking features. The order of the feature importances between two configurations can not be captured by Figure 21, however they vary between different configurations, even when two configurations select the same top 20 features.

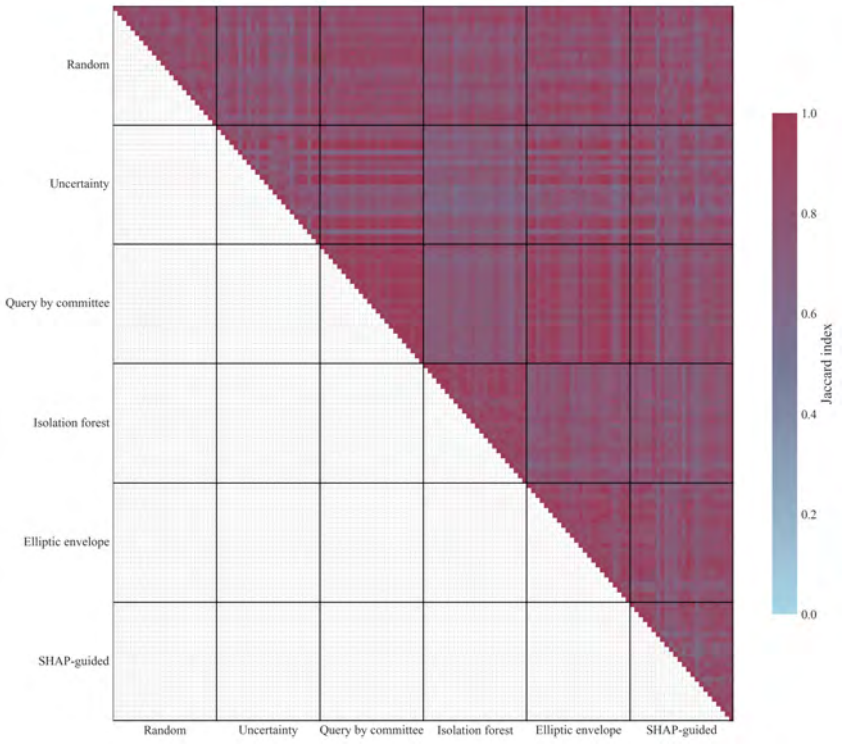


Figure 21: Jaccard index for features selected during feature selection across all query strategy and b configurations.

6.3.2 *Hyperparameter optimization*

Model refinement is conducted by evaluating a range of hyperparameter configurations, selecting the one that achieves the highest net value score on the validation set. The search space used during the TPE optimization procedure is summarized in Table 9, while the optimal hyperparameters for each configuration are provided in Appendix 10.5. Note that the classification threshold τ refers to the final threshold used during training, as visualized in Figure 15.

Several patterns emerge from the optimized configurations shown in Table 18. A maximum tree depth D of approximately 6 appears to be optimal for most non-extreme risk preference scenarios, except for uncertainty sampling at $b = €30K$. One surprising outcome is the threshold $\tau = 0.26$ for the elliptic envelope strategy at $b = €50$, despite the dominance of false positive costs in this setting, which would typically push the threshold closer to 1, as seen with the other query strategies. A similar selection is seen for $b = €1M$, where all thresholds are 0, emphasizing a strong preference for recall over precision.

6.4 Test results

Each configuration is evaluated on an unseen test set consisting of 611,957 transactions, of which 581 are labeled as illicit. Before interpreting the raw test results, it is important to assess whether the optimization, consisting of feature selection and hyperparameter optimization, yielded performance improvements. A structural decrease in performance would warrant a more critical examination of the methodology and its assumptions. Figure 37 in Appendix 10.6 displays the average change in key metrics (Δ-metrics), computed as the difference between post- and pre-optimization scores. On average, the Δ recall is slightly positive across the various strategies. However, the optimization also led to large drops in the true negative rates, especially for large values of b. Since the increase in recall across most configurations is did have an associated sharp decline in TNR, the optimization is kept.

A detailed breakdown of confusion matrix components, recall, precision, true negative rate for the evaluated configurations is provided in Table 13.

Query strategy	Benefit	TN	FP	FN	TP	Recall	Precision	True negative rate
Random	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	610,740	636	524	57	0.098	0.082	0.999
	€10K	580,054	31,322	152	429	0.738	0.014	0.949
	€25K	504,411	106,965	55	526	0.905	0.005	0.825
	€50K	497,797	113,579	21	560	0.964	0.005	0.814
	€100K	463,945	147,431	22	559	0.962	0.004	0.759
Uncertainty	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	609,616	1,760	448	133	0.229	0.070	0.997
	€10K	586,442	24,934	233	348	0.599	0.014	0.959
	€25K	547,267	64,109	468	113	0.194	0.002	0.895
	€50K	519,906	91,470	93	488	0.840	0.005	0.850
	€100K	0	611,376	0	581	1.000	0.001	0.000
Query by committee	€50	611,368	8	571	10	0.017	0.556	1.000
	€500	610,349	1,027	485	96	0.165	0.085	0.998
	€1K	605,660	5,716	349	232	0.399	0.039	0.991
	€10K	588,236	23,140	173	408	0.702	0.017	0.962
	€25K	568,547	42,829	146	435	0.749	0.010	0.930
	€50K	340,063	271,313	5	576	0.991	0.002	0.556
	€100K	406,940	204,436	54	527	0.907	0.003	0.666
Isolation forest	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	611,376	0	581	0	0.000	0.000	1.000
	€10K	561,701	49,675	137	444	0.764	0.009	0.919
	€25K	604,698	6,678	375	206	0.355	0.030	0.989
	€50K	496,494	114,882	69	512	0.881	0.004	0.812
	€100K	338,130	273,246	7	574	0.988	0.002	0.553
Elliptic envelope	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	611,372	4	576	5	0.009	0.556	1.000
	€10K	553,287	58,089	117	464	0.799	0.008	0.905
	€25K	569,951	41,425	211	370	0.637	0.009	0.932
	€50K	241,087	370,289	19	562	0.967	0.002	0.394
	€100K	440,697	170,679	37	544	0.936	0.003	0.721
SHAP-guided	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	611,376	0	581	0	0.000	0.000	1.000
	€10K	574,559	36,817	147	434	0.747	0.012	0.940
	€25K	503,230	108,146	80	501	0.862	0.005	0.823
	€50K	409,112	202,264	21	560	0.964	0.003	0.669
	€100K	284,635	326,741	19	562	0.967	0.002	0.466
Passive	€50	611,376	0	581	0	0.000	0.000	1.000
	€500	611,376	0	581	0	0.000	0.000	1.000
	€1K	611,376	0	581	0	0.000	0.000	1.000
	€10K	496,419	114,957	14	567	0.976	0.005	0.812
	€25K	494,301	117,075	12	569	0.979	0.005	0.809
	€50K	497,850	113,526	7	574	0.988	0.005	0.814
	€100K	503,108	108,268	16	565	0.972	0.005	0.823

Table 13: Comparison of metrics (TN, FP, FN, TP, recall, precision, true negative rate) for each query strategy across different b values.

Although Table 13 provides detailed metrics, it may be difficult to interpret trends between different strategies. A classee plot, commonly employed to illustrate confusion matrix components across thresholds, is used in Figure 22 to visualize the classification performance. In this case, b implicitly drives the classification

threshold, so it is used as the x-axis. Instances with illicit ground-truth labels are displayed above the zero line, showing the trade-off between recall and false negative rate (FNR), which sum to 100%. Below the zero line, ground-truth legitimate instances are used to illustrate the trade-off between true negative rate (TNR) and false positive rate (FPR), which also sum to 100%.

For most b values, the FPR remains near zero, but becomes noticeable at higher b, except for passive learning where the FPR remains around 20%. Given the severe class imbalance, even these seemingly small increases in FPR translate to a large number of false positives, as seen in Table 13. In all query strategies, recall and FPR typically increase with for configurations with higher b, supporting the expected trend: optimizing the threshold for higher values of b yields more true positives but as the cost of false positives. Query by committee shows a very stable increase, whereas uncertainty sampling and passive learning have more variability in performance across different b values. Some configurations have particularly poor performance. Examples are random at $b = €15K$, elliptic envelope with $b = €30K$, and SHAP-guided at $b = €20K$. Random at $b = €15K$ coincides with the performance drop shown in Appendix 10.6. However, the other poor performances do not, indicating that the hyperparameter optimization did not increase performance for the other two configurations.

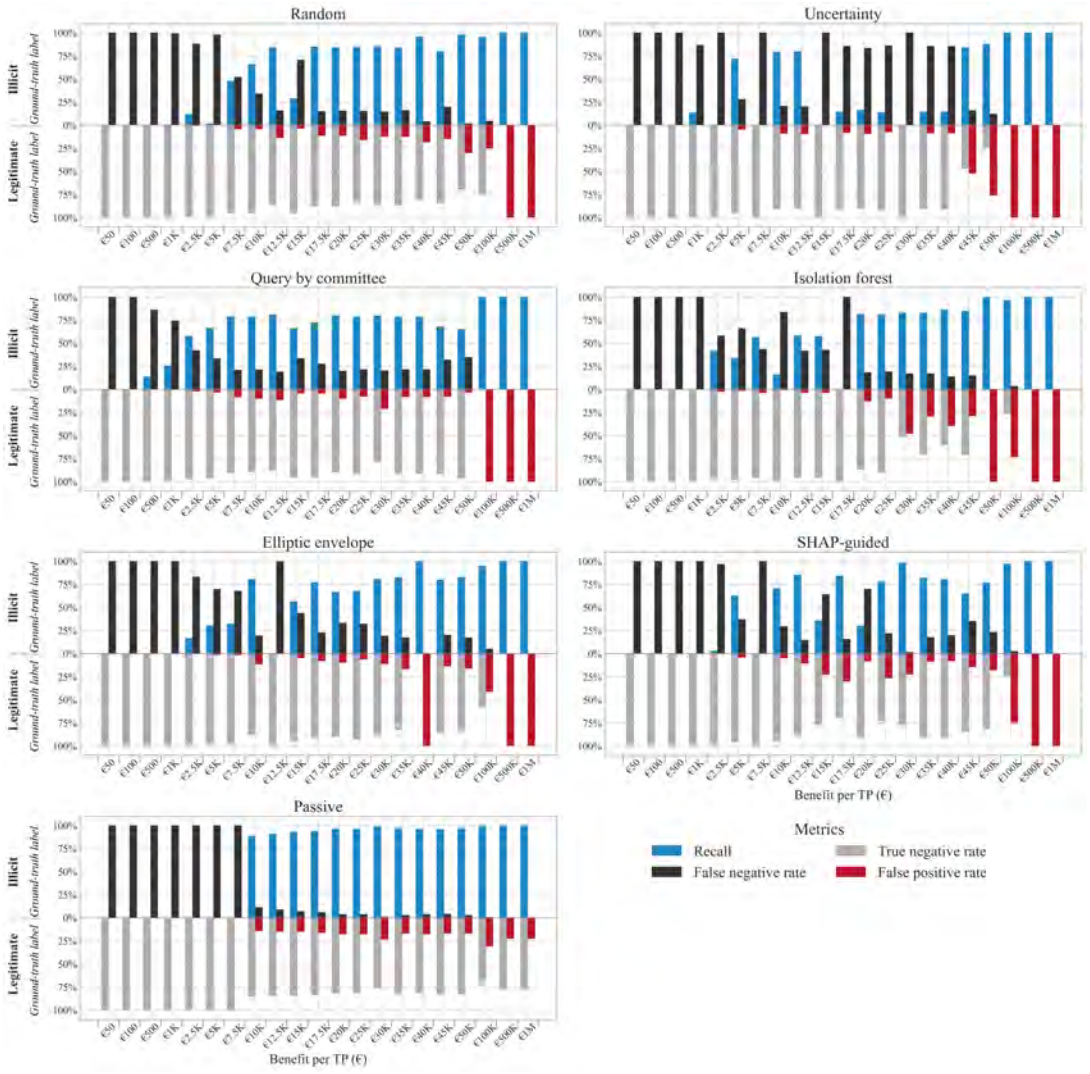


Figure 22: Classee plot showing the recall, false negative rate, true negative rate, and false positive rate for varying b, across all query strategies.

The fundamental challenge of detecting money laundering is further highlighted in the precision-recall (PR) curves shown in Figure 23. These curves reveal a steep trade-off: improving recall often comes at a substantial cost to precision. This behavior is typical in highly imbalanced classification tasks, where the

minority class (illicit transactions) is exceedingly rare. Each line in the plot represents a model's performance (recall, precision) across a range of classification thresholds.

Although the risk preference parameter b influences the threshold selected during training, the PR curves themselves are computed by sweeping over all possible thresholds. As a result, the direct effect of b is limited to only how it formed the labeled dataset under each query strategy, but is otherwise neutralized in the plot. Despite this, a consistent trend emerges: pushing recall higher leads to steep drops in precision. This indicates that the model finds it increasingly difficult to separate illicit from legitimate transactions at scale, underscoring the inherent challenge of minimizing false positives while maximizing detection.

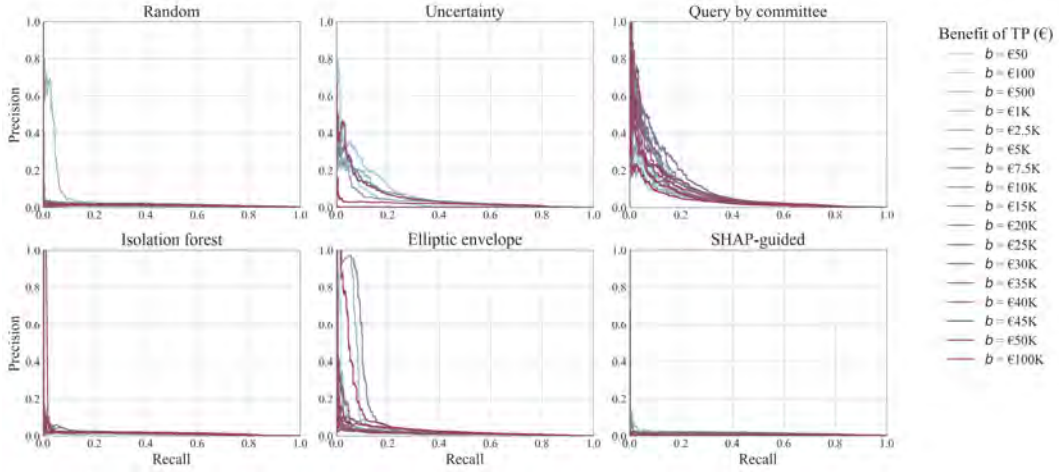


Figure 23: PR curves with varying b values, across all query strategies.

To show this trade-off in terms of false alerts and missed illicit transactions, Figure 24 is made. The non-linear shape of the datapoints reflects diminishing returns: at low b values, only high-confidence cases are flagged, resulting in few false positives. For higher b values, under the same query strategy, lower-confidence cases are also flagged. This reduces false negatives but introduces disproportionately more false positives, because less certain predictions are also included in the illicit classifications. Due to the unstable datapoints of uncertainty sampling and passive learning, the fitted curve does not follow the datapoints well.

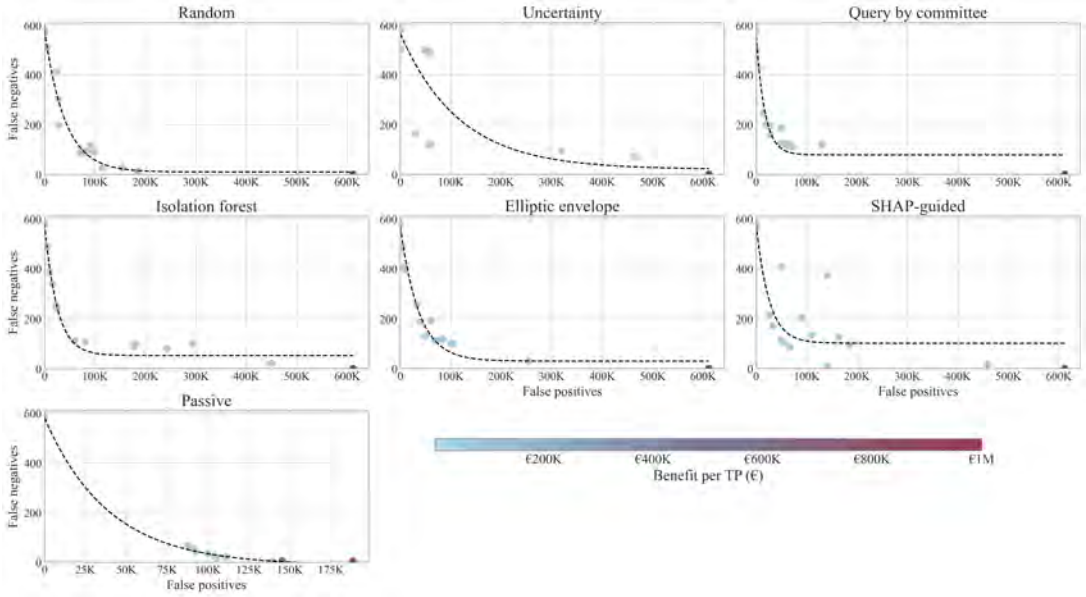


Figure 24: False positives plotted against false negatives for each query strategy. Each datapoint corresponds to a different risk preference, parameterized by b. A decaying exponential function of the form $\alpha \cdot \exp(-\beta \cdot \text{FP}) + \gamma$ is fitted over the datapoints.[7]

6.5 PATTERN RECOGNITION

To assess how well the models detect specific money laundering patterns, performance is analyzed per pattern type (Figure 2). Figure 25 reports the average recall and precision per pattern, aggregated across all benefit levels for each query strategy. Recall is defined as the proportion of correctly identified illicit transactions of a given pattern, and precision as the fraction of those predictions that are indeed illicit.

The results reveal little variation in pattern detectability. Most patterns are more detectable than "no pattern", but the severity depends on the query strategy. Query by committee consistently yields the highest recall across patterns and outperforms passive learning in both recall and precision. SHAP-guided profiling has outperforms uncertainty sampling and has similar performance to isolation forest. However, it underperforms relative to random sampling in both metrics, suggesting that its profiling approach does not generalize well to the test data. While it does not correspond to a high minority class proportion (shown in Figure 18), random sampling achieves high precision on gather-scatter despite obtaining moderate recall.

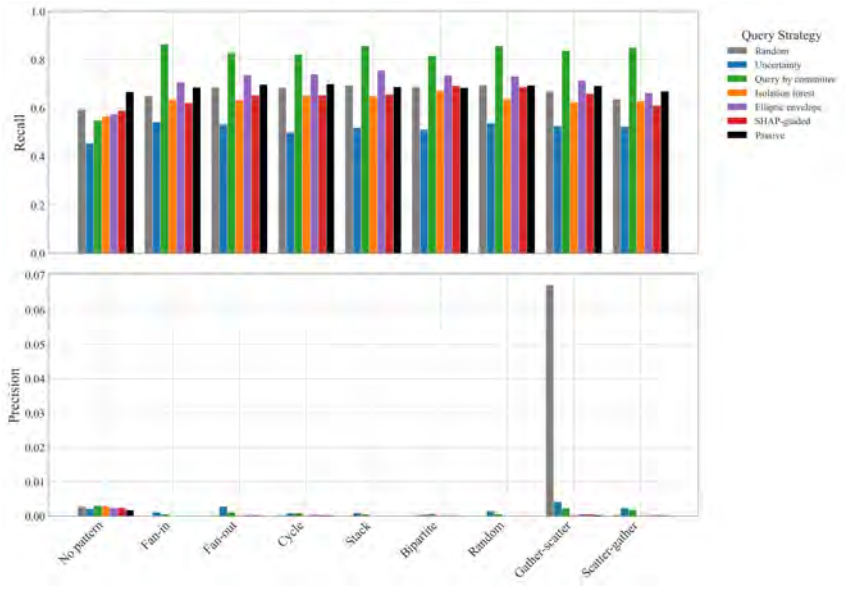


Figure 25: Average recall and precision per money laundering pattern (including "no pattern") across all benefit values, shown per query strategy.

To further quantify the model's effectiveness in identifying patterned behavior, Figure 26 presents the difference in recall between illicit transactions that follow a known pattern and those that do not ("no pattern"). For each query strategy and benefit level, the average recall difference (Δ recall = pattern recall minus no-pattern recall) is shown. All averages are positive, indicating that the models associated detect illicit transactions more effectively when they are part of a layering pattern. A more uniform observation holds for query by committee, elliptic envelope and passive learning, where the model is better at pattern detection for all b. In contrast, the strategies random sampling, uncertainty sampling and SHAP-guided profiling detect patterns worse than illicit transactions associated with no pattern for some b.

[7]Fitted parameters (α, β, γ) for each query strategy: random sampling: (570.7, 2.484e-05, 9.569); uncertainty sampling: (546.2, 8.53e-06, 19.98); query by committee: (479.5, 5.876e-05, 78.31); isolation forest: (528.6, 4.021e-05, 52.47); elliptic envelope: (539, 2.737e-05, 30.08); SHAP-guided profiling: (476, 3.963e-05, 102); passive learning: (593.3, 2.557e-05, -12.28).

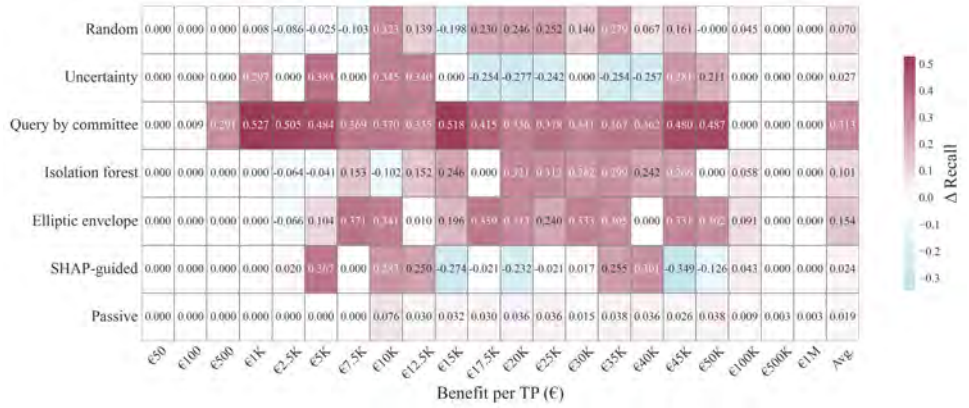


Figure 26: Δ Recall: the difference between recall on patterned illicit transactions and not associated with a pattern (”no pattern”), shown per query strategy and benefit level.

7

DISCUSSION

The discussion begins with an interpretation of the results (7.1) regarding the research questions. The discussion also addresses limitations of the current work (7.2) and broader challenges (7.3).

7.1 INTERPRETATIONS

Active learning vs. passive learning (RQ1) *How does active learning trained on 2.08% of the labeled data compare in performance (recall, precision, and true negative rate) to supervised learning using the complete training set?*

Our experiments demonstrate that active learning, using only 2.08% of the labeled data, can achieve performance that approaches that of fully supervised learning on the entire training dataset (passive learning). As shown in Figure 13, passive learning generally achieves higher recall across all benefit levels b on the validation set. However, the performance gap is modest, and both precision and true negative rate (TNR) remain of similar magnitude across approaches. Considering that passive learning uses approximately 50 times more labeled data, the relative performance of active learning is notable. A key explanation lies in the class imbalance dynamics observed during training (Figure 17). All active learning strategies disproportionately sample illicit transactions (with the exception of random sampling), with query by committee identifying up to 14 times more illicit cases than their overall prevalence in the dataset. It should be noted that the main contributor of the inflated minority class proportions of the query strategies is due to the selection of illicit transactions with no associated pattern, as shown in Figure 18. Nonetheless, this targeted selection likely enhances learning efficiency, compensating for the limited label budget.

After feature selection and hyperparameter tuning, active learning strategies such as query by committee, isolation forest, and elliptic envelope achieve high recall on the test set (Figure 25). Notably, query by committee surpasses passive learning in both recall and precision across all pattern types, including "no pattern".

These findings suggest that active learning offers a viable approach for reducing alert investigation costs in AML pipelines, provided that an appropriate query strategy is used.

Effectiveness of SHAP-guided querying (RQ2) *To what extent can an explainability-guided query strategy informed by SHAP values (a widely adopted XAI technique) effectively identify illicit transactions in a synthetic dataset, and how does its performance compare to that of established query strategies in terms of precision, recall and true negative rate?*

SHAP-guided profiling was designed to select transactions whose feature importance profiles are most dissimilar to the average profile of legitimate transactions, referred to as the legitimate profile. This selection mechanism aims to flag transactions that deviate from legitimate feature importance behavior and was evaluated against established strategies including random sampling, uncertainty sampling, query by committee, isolation forest, and elliptic envelope.

During training (Figure 13), SHAP-guided profiling achieves recall roughly in line with other strategies, but shows instability across certain benefit per true positive values (notably at €1; €15K; €20K; and €50K). True negative rates were generally comparable, though precision was not reliably comparable due to the severe class imbalance. Further analysis reveals that SHAP-guided profiling consistently selects a distinct subset of transactions compared to other strategies (Figure 16), suggesting it captures a different, possibly narrower, part of the decision space[8]. However, results on the test set (Figure 25) highlight critical limitations: SHAP-guided profiling underperforms random sampling in both recall and precision across all layering

[8]The decision space refers to the range of possible inputs and how a model distinguishes between classes based on those inputs.

patterns, including illicit transactions with no associated pattern. This indicates a structural flaw, as by focusing on transactions that deviate from the legitimate profile, the strategy overemphasizes obvious cases and fails to detect more ambiguous or subtle laundering behavior that are essential for improving recall. This limited scope is also reflected in class imbalance outcomes (Figure 17): while SHAP-guided profiling selects a slightly higher proportion of illicit transactions than random or passive sampling, the difference is marginal. The assumption that strong deviation from the legitimate profile correlates with illicitness does not hold up empirically.

The interpretability advantage of SHAP-guided profiling also proves overstated. While Table 12 provides an explanation report for a selected transaction which is generated due to the inherent interpretability aspect of the strategy. In contrast, black-box strategies like query by committee or uncertainty sampling need post hoc application of XAI methods to explain decisions made by the model. In a high-risk setting like money laundering detection, where decisions require both accuracy and accountability, the embedded interpretability of SHAP-guided profiling may appear transparent but it lacks diagnostic depth, especially when coupled with underwhelming performance.

Taken together, these findings indicate that SHAP-guided profiling is not suitable as a standalone query strategy for AML applications. It fails to deliver adequate detection performance and does not provide interpretability benefits strong enough to justify that shortfall. A more viable approach would decouple selection and explanation, using a high-performing strategy such as query by committee for selecting transactions, and applying SHAP post hoc to generate explanations aligned with the decision. This layered setup supports both performance and trust, something SHAP-guided profiling cannot achieve as a standalone strategy.

Trade-off for different risk-preferences (RQ3) *How do varying risk preferences, reflected in the cost-sensitive optimization, impact the trade-off between recall, precision, and true negative rate?*

We investigated how variations in the benefit per TP parameter b within our cost-sensitive classification threshold optimization influence the trade-off between recall, precision, and true negative rate (TNR). Larger values for b, which assigns greater importance to identifying true positives, results in lower classification thresholds, thereby favoring recall at the expense of both precision and TNR. This dynamic is clearly demonstrated in the thresholds evolution during training in Figure 15, where larger b values push the model toward more aggressive flagging of potentially illicit transactions. Similarly, Figure 14 shows that while recall improves at larger b values, it also incurs an increase in false positives, a critical consideration in AML systems with high transaction volumes and limited analyst capacity. This trade-off persists during testing, as tabulated for the test results in Table 13.

It is important to note that these interpretations rely on the design of the net value function (Equation 9). The general applicability of these conclusions would benefit from further validation by domain experts and regulators to ensure that they are in line with institutional risk tolerances and compliance constraints.

7.2 Limitations

This study faces several limitations stemming from the used dataset, design choices and practical limitations. These limitations may affect the generalizability and practical application of the results.

AMLworld Various limitations originate from the nature of the AMLworld dataset, which is synthetic and relatively compact in terms of feature space.

- Limited feature space: The dataset lacks a variety of contextual and supplementary data types that are critical in operational AML systems, most notably customer metadata and multi-dimensional data sources. Attributes such as company structure, Ultimate Beneficial Ownership (UBO), industry classification, and geographic risk exposure are essential for recognizing atypical financial behavior. Furthermore, the absence of multi-source data (e.g. FIU trend reports, device fingerprinting and behavioral analytics) limits the model's ability to identify nuanced laundering tactics [18]. As highlighted by Li, Ranbaduge, and Ng [58], effective AML detection relies on more than transaction data alone. The absence of such enriched feature sets restricts the realism for detection modeling.

- Limited transaction history: The LI-small subset used in this study contains approximately 5 million transactions over a 10-day span (Figure 5). This narrow time frame restricts the model's capacity to learn and exploit long-term behavioral dynamics, which are crucial for detecting sophisticated laundering strategies that evolve gradually over time.

- Simplified and isolated laundering patterns: In AMLworld, laundering strategies are implemented as static and independent patterns (Figure 9), whereas real-world money laundering operations are typically adaptive and involve the dynamic combination of multiple tactics [15]. This simplification limits the realism of the simulated behavior and increases the risk of model overfitting to simulation-specific regularities. As a result, models trained on AMLworld may struggle to generalize when deployed in operational settings that involve more complex, layered, and evolving laundering schemes.
- Simulation instability: According to AMLworld documentation, small changes in simulation parameters can result in large and sometimes erratic variations in the data [2]. Found instabilities include self-looping transactions (Table 5), liquidity imbalances across banks (Table 28b), and sudden drops in transaction activity near the end of the simulation period (Figure 5). These instabilities undermine the model's capacity to generalize to real-world AML detection scenarios.

Absence of human-in-the-loop A major limitation of this study is the absence of human-in-the-loop validation. In operational AML settings, domain experts are central to interpreting flagged transactions, assessing risk, and refining detection strategies. The current framework assumes perfect label quality and does not incorporate expert feedback in evaluating model decisions or SHAP-based explanations. For example, while SHAP-guided explanations may appear informative from a model-centric view, it remains unclear whether AML analysts find them actionable or trustworthy. Moreover, active learning strategies that prioritize uncertainty or explanation diversity may produce different results when guided by human judgment, potentially leading to a more effective or interpretable query strategy.

Computational load The proposed active learning framework imposes a significant computational burden. This is further compounded by the SHAP-based feature selection and hyperparameter optimization. As reported in Table 11, the total runtime for training, validation, and testing exceeds 150 hours for the active learning pipeline, while the passive learning took over 34 hours. The contributors to this overhead are primarily the repeated retraining of models and the computation of SHAP values. Due to these constraints, it was not feasible to explore the full range of budget values (b), and several promising research directions were deferred to future work (Section 8.2).

Limited robustness evaluation This study does not evaluate the model's robustness to adversarial scenarios [68] such as concept drift, adversarial evasion, or noisy labels, each of which is prevalent in real-world AML systems. Since laundering techniques and financial patterns evolve over time, models must maintain stability under shifts of the underlying distribution for illicit transactions. While robustness testing (e.g., through detecting adversarial examples or drift detection [68]) could offer valuable insights, the short time horizon of the AMLworld dataset and due to the computational overhead of the research, robustness is left for future studies.

Benchmarking A notable limitation of this work is the absence of benchmarking across a wide range of machine learning models. While such comparisons are standard in academic research and useful for establishing performance baselines, they can distract from the core practical challenges in AML, such as extreme class imbalance, limited labeling budgets, and the need for interpretability. Moreover, benchmarking would significantly increase computational and design complexity, especially when considering multiple query strategies, risk-preferences and evaluation metrics. As a result, this research opts for a focused methodological approach rather than exhaustive model comparison. However, this choice does limit the ability to generalize findings across different models or to position results within broader state-of-the-art performance benchmarks.

7.3 Challenges

Developing effective AML models involves navigating a range of structural and technical challenges. These challenges are largely external and reflect the nature of the domain, its data limitations, and institutional constraints.

Data access, privacy, and fragmentation One of the most pressing challenges in AML research is the lack of access to real, labeled transaction data. Financial institutions operate under strict legal, privacy, and regulatory constraints, such as GDPR and banking secrecy laws, which severely limit the sharing of

sensitive customer and transaction information. As a result, most publicly available datasets are synthetic or heavily anonymized. While synthetic data supports controlled experimentation, it often reflects predefined laundering typologies and lacks the complexity, subtlety, and unpredictability of real-world money laundering behavior. This reliance on fixed templates, as noted by Lute [65], can cause models to overfit to dataset-specific patterns, thereby reducing generalizability.

Compounding this challenge is the fragmented nature of the financial ecosystem: banks, payment processors, and regulators typically operate in silos, each with only a partial view of the broader financial network. This hampers the detection of cross-institution or transnational laundering patterns. AMLworld, for instance, assumes global visibility across all banks, a highly unrealistic scenario. In practice, constrained observability remains a significant challenge, although emerging techniques in federated and privacy-preserving machine learning offer potential solutions [25, 26, 58, 61].

Severe class imbalance Money laundering is an inherently rare phenomenon, leading to highly imbalanced datasets where illicit transactions make up a tiny fraction of the total. This extreme skew challenges standard machine learning methods, which tend to overfit to the majority class, resulting in high apparent accuracy but poor performance on the minority class. Addressing this imbalance is critical for effective detection, particularly when annotation resources are limited.

Evolving criminal tactics Money laundering methods are constantly evolving in response to changes in detection systems, regulatory environments, and enforcement strategies. Launderers adapt by varying transaction patterns, exploiting new technologies, or shifting across jurisdictions. This creates a moving target for AML systems, which must detect not only known patterns of money laundering but also novel and subtle behaviors. Developing models that can generalize across different patterns or adapt to new laundering strategies remains a significant research challenge.

Label scarcity and imperfection Supervised AML models rely heavily on labeled examples of legitimate and illicit transactions, yet obtaining such labels is a significant challenge. Labeling is labor-intensive, costly, and inherently subjective, requiring AML teams to investigate transaction histories, customer profiles, and external intelligence before assigning a label. As a result, labeled datasets tend to be small, imbalanced, and noisy. Compounding this issue, the experimental setups often assume a perfect labeling oracle that provides accurate and certain labels for every selected transaction. In reality, AML teams face uncertainty due to limited evidence, evolving typologies, and the potential for delayed or incorrect labels. This discrepancy between idealized labeling and practical constraints oversimplifies the problem and may lead to overly optimistic estimates of model performance.

Limited feedback Another structural challenge is the limited feedback loop between financial institutions and FIUs. FIUs do offer general feedback through blogs, industry conferences, and periodic reports, informing financial institutions about trends in financial crime, typologies of money laundering, and emerging threats. However, the extent to which they provide case-specific feedback is a double-edged sword. Informing financial institutions about which transactions were linked to illicit activities allows them to refine their detection models, but this comes at the cost of tipping-off risks. This occurs when the account and their accomplices are alerted when financial institutions close or block their accounts mentioned by FIUs. This enables suspects to relocate operations before law enforcement can complete their investigations.

Lack of explainability Although the recently adopted EU AI Act [72] clarifies that transaction monitoring systems used solely for administrative purposes (such as filing SARs) are not classified as high-risk, the issue of explainability remains central. The 2019 Ethics Guidelines for Trustworthy AI call for transparency and interpretability as foundational principles [72]. Yet, many machine learning models used in AML operate as black boxes, making it difficult to provide human-understandable justifications for alerts. In a regulated domain where trust, accountability, and oversight are essential, this lack of transparency hinders adoption of state-of-the-art machine learning models and techniques.

8

CONCLUSION

The conclusion starts with a summary of the research in Section 8.1. Possible future work directions are discussed in Section 8.2.

8.1 SUMMARY

This research introduced a cost-sensitive active learning framework for anti-money laundering (AML) detection, evaluating its performance under a limited labeling budget and investigating whether a query strategy guided by the explainable AI technique SHAP can effectively identify illicit transactions while providing meaningful interpretability. Across three research questions, the findings reveal a nuanced interplay between learning efficiency, interpretability, and institutional risk preferences in designing AML systems.

First, we demonstrated that active learning using only 2.08% of labeled data can closely approximate the recall, precision, and true negative rate of fully supervised learning. This efficiency is largely driven by the selection behavior of the strategies, as they some are selecting more illicit transactions than the overall proportion in the training data. An example is query by committee, which identify illicit transactions at rates far exceeding their minority class prevalence (14 time more), thereby accelerating model learning while maintaining relatively low investigation costs.

Second, we found that SHAP-guided profiling, despite its intuitive appeal for inherent interpretability, underperforms on recall and precision as it struggles to detect subtle laundering patterns. While it prioritizes highly anomalous transactions and offers built-in explanations, this comes at the cost of narrow coverage and reduced recall. The assumption that deviation from the average feature importance profile of legitimate transactions indicates illicitness does not hold up, and its apparent transparency lacks the diagnostic depth provided by techniques that apply explainability post hoc. These findings suggest SHAP is more effective as a post hoc explanation tool than as an explainability-guided query strategy.

Finally, by varying the benefit per true positive parameter in the cost-sensitive classification threshold, we revealed clear trade-offs between recall, precision, and true negative rate. Higher benefit values correspond to risk-averse behavior, favoring broad detection at the expense of false positives, whereas lower values indicate risk-seeking behavior, emphasizing precision and reduced false alarms. This tunable thresholding mechanism enables alignment with institutional risk tolerances but depends critically on domain-informed calibration of the net value function.

In summary, this work shows that active learning can substantially reduce labeling effort without sacrificing detection performance, but the choice of query strategy must balance detection coverage, interpretability, and operational risk. Effective AML systems require query strategies evaluated not only on accuracy but also on their practical fit within the constraints of real-world workflows.

8.2 FUTURE WORK

The practical deployment of active learning in AML contexts remains constrained by limited involvement of human analysts. Future research should integrate AML experts into the loop to evaluate SHAP explanations, assess the actionability of query selections, and benchmark strategies based on their operational impact. Such collaboration would yield valuable insights into how expert feedback influences label accuracy and model trustworthiness.

Our research assumed a perfect labeling oracle, which overlooks real-world labeling uncertainty. Future studies should simulate noisy labels by introducing probabilistic error models based on analyst uncertainty or disagreement. This would allow for a more realistic assessment of robustness in active learning under imperfect supervision.

Further exploration is needed into explainability-guided query strategies. Techniques such as counterfactual explanations, especially near decision boundaries, and contrastive explanations, where samples are selected based on dissimilarity to neighbors in the feature or explanation space, may enhance interpretability during transaction selection. Coupling these approaches with human-in-the-loop evaluation and noisy labels will clarify the true value added by explainability in guiding data acquisition.

While SHAP-guided profiling offers a way to select anomalous instances based on model explanations, it may benefit from integration with informativeness-based strategies. As Kirsch, Amersfoort, and Gal [52] and Du et al. [23] suggest, combining selection criteria such as informativeness and representativeness within a single query strategy can outperform methods focusing on only one aspect. Following this idea, future work could explore hybrid SHAP-guided strategies that combine profiling with informativeness scores. Alternatively, as Cunha et al. [19] propose, anomaly detection methods like SHAP-based profiling could be used at early training stages, followed by informativeness-driven querying later.

Another promising direction is to extend cost-sensitive threshold optimization beyond its current linear formulation in terms of true positives (TP), false positives (FP), and false negatives (FN). A non-linear objective, such as a concave reward function TP^{α} with $0 < \alpha < 1$, could better reflect the diminishing marginal utility of detecting additional true positives. This adjustment would capture real-world considerations, including regulatory expectations that penalize low TP rates, where the incremental value of each detected true positive decreases as the model improves.

Finally, varying the size of the labeling budget instead of fixing it at 2.08% of the training data would provide insight into the trade-offs between annotation effort and performance. This would be particularly valuable to operations teams in banks seeking to optimize resource allocation and capacity.

References

[1] Financial Action Task Force (FATF). *Money Laundering through Money Remittance and Currency Exchange Providers*. Tech. rep. Financial Action Task Force and Groupe d'action financière, 2010.

[2] Erik Altman et al. "Realistic synthetic financial transactions for anti-money laundering models". In: *Advances in Neural Information Processing Systems* 36 (2024).

[3] Alejandro Correa Bahnsen et al. "Cost sensitive credit card fraud detection using Bayes minimum risk". In: *2013 12th international conference on machine learning and applications*. Vol. 1. IEEE. 2013, pp. 333–338.

[4] Nazanin Bakhshinejad et al. "A Graph-Based Deep Learning Model for the Anti-Money Laundering Task of Transaction Monitoring". In: *IJCCI 2024* 16 (2024), pp. 496–507.

[5] Richard Bellman. "Dynamic programming". In: *science* 153.3731 (1966), pp. 34–37.

[6] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13 (2012), pp. 281–305. URL: https://jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[7] James Bergstra et al. "Algorithms for hyper-parameter optimization". In: *Advances in Neural Information Processing Systems*. Vol. 24. 2011.

[8] Richard J Bolton and David J Hand. "Statistical fraud detection: A review". In: *Statistical science* 17.3 (2002), pp. 235–255.

[9] Anthony Bonato, Juan Sebastian Chavez Palan, and Adam Szava. "Enhancing anti-money laundering efforts with network-based algorithms". In: *International Conference on Complex Networks and Their Applications*. Springer. 2024, pp. 115–124.

[10] Bernardo Branco et al. "Interleaved sequence RNNs for fraud detection". In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 3101–3109.

[11] E. M. Carneiro et al. "High-Cardinality Categorical Attributes and Credit Card Fraud Detection". In: *Mathematics* 10.20 (2022), p. 3808.

[12] Miguel Carvalho, Armando J Pinho, and Susana Brás. "Resampling approaches to handle class imbalance: a review from a data perspective". In: *Journal of Big Data* 12.1 (2025), p. 71.

[13] Chainalysis. *The 2021 Crypto Crime Report*. https://go.chainalysis.com/2021-Crypto-Crime-Report.html. 2021.

[14] Derek Chau and Maarten van Dijck Nemcsik. *Anti-money laundering transaction monitoring systems implementation: Finding anomalies*. John Wiley & Sons, 2020.

[15] Aml world check. *Understanding Layering in Money Laundering*. 2024. URL: https://amlworldcheck.com/layering-money-laundering/.

[16] Zhiyuan Chen et al. "Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review". In: *Knowledge and Information Systems* 57 (2018), pp. 245–285.

[17] Zhiyuan Chen et al. "Variational autoencoders and wasserstein generative adversarial networks for improving the anti-money laundering process". In: *Ieee Access* 9 (2021), pp. 83762–83785.

[18] Cosive. *Fraud Detection in Banking: The Ultimate Guide*. 2024. URL: https://www.cosive.com/fraud-detection-in-banking-guide.

[19] Leandro L Cunha et al. "Active learning in the detection of anomalies in cryptocurrency transactions". In: *Machine Learning and Knowledge Extraction* 5.4 (2023), pp. 1717–1745.

[20] DataRobot. *Money Laundering Detection: Business Accelerator*. 2024. URL: https://docs.datarobot.com/en/docs/get-started/gs-dr5/biz-accelerators/money-launder.html.

[21] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[22] Jenny Domashova and Natalia Mikhailina. "Usage of machine learning methods for early detection of money laundering schemes". In: *Procedia Computer Science* 190 (2021), pp. 184–192.

[23] Bo Du et al. "Exploring representativeness and informativeness for active learning". In: *IEEE transactions on cybernetics* 47.1 (2015), pp. 14–26.

[24] Ahmad Naser Eddin et al. "Anti-money laundering alert optimization using machine learning with graphs". In: *arXiv preprint arXiv:2112.07508* (2021).

[25] Fabrianne Effendi and Anupam Chattopadhyay. "Privacy-Preserving Graph-Based Machine Learning with Fully Homomorphic Encryption for Collaborative Anti-money Laundering". In: *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer. 2024, pp. 80–105.

[26] Marie Beth van Egmond, Thomas Rooijakkers, and Alex Sangers. “Privacy-preserving collaborative money laundering detection”. In: *ERCIM NEWS* 27 (2021).

[27] Béni Egressy et al. “Provably powerful graph neural networks for directed multigraphs”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 10. 2024, pp. 11838–11846.

[28] European Banking Authority. *Study of the Cost of Compliance with Supervisory Reporting Requirements*. Report EBA/Rep/2021/15. Dec. 2021. URL: https://www.eba.europa.eu/sites/default/documents/files/document_library/Publications/Reports/2021/1019022/Study%20of%20the%20cost%20of%20compliance%20with%20supervisory%20reporting%20requirements.pdf.

[29] European Banking Federation. *Banking in Europe: EBF Facts & Figures 2024*. Report. Brussels, Belgium, Dec. 2024. URL: https://www.ebf.eu/wp-content/uploads/2024/12/EBF-Banking-in-Europe-Facts-Figures-2024-2023-banking-statistics-December-2024.pdf.

[30] European Central Bank. *Payments statistics: 2022*. 2022. URL: https://www.ecb.europa.eu/press/stats/paysec/html/ecb.pis2022~8bb6cc08f4.en.html.

[31] Europol. *Cryptocurrencies: Tracing the Evolution of Criminal Finances*. Tech. rep. Europol Spotlight Report. European Union Agency for Law Enforcement Cooperation (Europol), 2021. URL: https://www.europol.europa.eu/cms/sites/default/files/documents/Europol%20Spotlight%20-%20Cryptocurrencies%20-%20Tracing%20the%20evolution%20of%20criminal%20finances.pdf.

[32] Europol. *Global anti-money laundering framework – Europol report reveals poor success rate and offers ways to improve*. 2023. URL: https://www.europol.europa.eu/media-press/newsroom/news/global-anti-money-laundering-framework-%E2%80%93-europol-report-reveals-poor-success-rate-and-offers-ways-to-improve.

[33] Jiani Fan et al. “Deep Learning Approaches for Anti-Money Laundering on Mobile Transactions: Review, Framework, and Directions”. In: *arXiv preprint arXiv:2503.10058* (2025).

[34] Financial Action Task Force (FATF). *Money Laundering and Terrorist Financing in the Art and Antiquities Market*. FATF, 2023. URL: https://www.fatf-gafi.org/content/dam/fatf-gafi/reports/Money-Laundering-Terrorist-Financing-Art-Antiquities-Market.pdf.coredownload.pdf.

[35] Financial Action Task Force (FATF). *Professional Money Laundering*. Tech. rep. Financial Action Task Force, 2023. URL: https://www.fatf-gafi.org/content/dam/fatf-gafi/reports/Professional-Money-Laundering.pdf.

[36] FIU-Nederland. *Annual Review 2023*. June 2024. URL: https://www.fiu-nederland.nl/wp-content/uploads/sites/2/2024/06/Jaarverslag-online-2023-EN.pdf.

[37] FIU-Nederland. *Banks - Reporting group*. URL: https://www.fiu-nederland.nl/en/reporting_group/banks/.

[38] Chunjiang Fu, Liang Gong, and Yupu Yang. “An improved active learning method based on feature selection”. In: *2015 International conference on Applied Science and Engineering Innovation*. Atlantis Press. 2015, pp. 170–174.

[39] Elshan Gadimov and Ermiyas Birihanu. “Real-time suspicious detection framework for financial data streams”. In: *International Journal of Information Technology* (2025), pp. 1–17.

[40] Bhavya Ghai et al. “Explainable active learning (xal) toward ai explanations as interfaces for machine teachers”. In: *Proceedings of the ACM on human-computer interaction* 4.CSCW3 (2021), pp. 1–28.

[41] KK Girish and Biswajit Bhowmik. “Money Laundering Detection in Banking Transactions using RNNs and Hybrid Ensemble”. In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE. 2024, pp. 1–7.

[42] Mario Gjoni, Albana Karameta Gjoni, and Holta Bako Kora. *Money laundering effects*. 2015.

[43] Board of Governors of the Federal Reserve System. *Changes in U.S. Payments Fraud from 2012 to 2016: Evidence from the Federal Reserve Payments Study*. Tech. rep. Federal Reserve Board, 2018. URL: https://www.federalreserve.gov/publications/files/changes-in-us-payments-fraud-from-2012-to-2016-20181016.pdf.

[44] Waleed Hilal, S Andrew Gadsden, and John Yawney. “Financial fraud: a review of anomaly detection techniques and recent advances”. In: *Expert systems With applications* 193 (2022), p. 116429.

[45] International Consortium of Investigative Journalists (ICIJ). *Offshore Leaks Database*. 2025. URL: https://offshoreleaks.icij.org/.

[46] Investopedia. *Night Cycle: What it Means, How it Works, Examples*. 2024. URL: https://www.investopedia.com/terms/n/night-cycle.asp.

[47] Paul Jaccard. “The distribution of the flora in the alpine zone. 1”. In: *New phytologist* 11.2 (1912), pp. 37–50.

[48] Rasmus Ingemann Tuffveson Jensen and Alexandros Iosifidis. “Fighting money laundering with statistics and machine learning”. In: *IEEE Access* 11 (2023), pp. 8889–8903.

[49] Stamatis Karlos et al. “Using active learning methods for predicting fraudulent financial statements”. In: *Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings*. Springer. 2017, pp. 351–362.

[50] Farzin Kazemi et al. “Active learning on stacked machine learning techniques for predicting compressive strength of alkali-activated ultra-high-performance concrete”. In: *Archives of Civil and Mechanical Engineering* 25.1 (2024), p. 24.

[51] Claire Greene Kevin Foster and Joanna Stavins. *2019 Survey of Consumer Payment Choice*. Tech. rep. Federal Reserve Bank of Atlanta, 2020. URL: https://www.atlantafed.org/banking-and-payments/consumer-payments/survey-of-consumer-payment-choice/2019-survey.

[52] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. *BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning*. OATML Blog, University of Oxford. 2019.

[53] Karel Križnar et al. *Explainable artificial intelligence meets active learning: A novel gradcam-based active learning strategy*. 2023.

[54] Dattatray Vishnu Kute et al. “Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review”. In: *IEEE access* 9 (2021), pp. 82300–82317.

[55] Danilo Labanca et al. “Amaretto: An active learning framework for money laundering detection”. In: *IEEE Access* 10 (2022), pp. 41720–41739.

[56] LexisNexis Risk Solutions. *True Cost of Financial Crime Compliance 2024*. https://risk.lexisnexis.com/global/en/insights-resources/research/true-cost-of-financial-crime-compliance-study. 2024.

[57] Na Li et al. “Active learning for data quality control: A survey”. In: *ACM Journal of Data and Information Quality* 16.2 (2024), pp. 1–45.

[58] Yang Li, Thilina Ranbaduge, and Kee Siong Ng. “Privacy technologies for financial intelligence”. In: *arXiv preprint arXiv:2408.09935* (2024).

[59] Junhong Lin et al. “FraudGT: A Simple, Effective, and Efficient Graph Transformer for Financial Fraud Detection”. In: *Proceedings of the 5th ACM International Conference on AI in Finance*. 2024, pp. 292–300.

[60] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[61] Wenzheng Liu et al. “A Federated Anti-money Laundering Detection Model with Bidirectional Graph Attention Network”. In: *International Conference on Intelligent Computing*. Springer. 2024, pp. 254–262.

[62] Joana Lorenz et al. “Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity (2020)”. In: *Preprint arXiv* (2020).

[63] Hanbin Lu and Haosen Wang. “Graph Contrastive Pre-training for Anti-money Laundering”. In: *International Journal of Computational Intelligence Systems* 17.1 (2024), p. 307.

[64] Yun Luo et al. “Xal: Explainable active learning makes classifiers better low-resource learners”. In: *arXiv preprint arXiv:2310.05502* (2023).

[65] Sara Lute. “What If we Cannot See the Full Picture? Anti-Money Laundering in Transaction Monitoring”. PhD thesis. Vrije Universiteit Amsterdam, 2024.

[66] Julie Moeyersoms and David Martens. “Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector”. In: *Decision support systems* 72 (2015), pp. 72–81.

[67] Nigel Morris-Cotterill. “Money laundering”. In: *Foreign Policy* 125 (2001), pp. 16–22.

[68] Viacheslav Moskalenko et al. “Model and training method of the resilient image classifier considering faults, concept drift, and adversarial attacks”. In: *Algorithms* 15.10 (2022), p. 384.

[69] Nasdaq and Verafin. *2024 Global Financial Crime Report*. Jan. 2024. URL: https://nd.nasdaq.com/rs/303-QKM-463/images/2024-Global-Financial-Crime-Report-Nasdaq-Verafin-20240119.pdf.

[70] James O’Donovan, Hannes F Wagner, and Stefan Zeume. “The value of offshore secrets: Evidence from the Panama Papers”. In: *The Review of Financial Studies* 32.11 (2019), pp. 4117–4155.

[71] Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. “The difficulty of passive learning in deep reinforcement learning”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 23283–23295.

[72] The European Parliament and The Council Of The European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 12 July 2024 laying down harmonised rules on artificial intelligence and amending certain Union legislative acts (Artificial Intelligence Act)*. 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

[73] Javier Pastor-Galindo et al. “The not yet exploited goldmine of OSINT: Opportunities, open challenges and future trends”. In: *IEEE access* 8 (2020), pp. 10282–10304.

[74] PBS Frontline. *Black Market Peso Exchange*. 1998. URL: https://www.pbs.org/wgbh/frontline/wgbh/pages/frontline/shows/drugs/special/blackpeso.html.

[75] Richard Phillips, Kyu Hyun Chang, and Sorelle A Friedler. “Interpretable active learning”. In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 49–61.

[76] Pradheepan Raghavan and Neamat El Gayar. “Fraud detection using machine learning and deep learning”. In: *2019 international conference on computational intelligence and knowledge economy (ICCIKE)*. IEEE. 2019, pp. 334–339.

[77] Pragati Raj. *Comparison of Random Forest and Neural Network for Classification on Structured Tabular Data*. https://medium.com/@praggrt/comparison-of-random-forest-and-neural-network-for-classification-on-structured-tabular-data-d5842ad044e3. Accessed: 2025-06-19. 2021.

[78] Matthew R. Redhead. *The Future of Transaction Monitoring: Better Ways to Detect and Disrupt Financial Crime*. Tech. rep. SWIFT Institute Working Paper. SWIFT Institute, May 2021. URL: https://www.swift.com/swift-resource/252235/download.

[79] Roam Analytics. *Categorical Variables in Tree Models*. 2016. URL: https://github.com/roamanalytics/roamresearch/blob/master/BlogPosts/Categorical_variables_in_tree_models/categorical_variables_post.ipynb.

[80] Peter J Rousseeuw and Katrien Van Driessen. “A fast algorithm for the minimum covariance determinant estimator”. In: *Technometrics* 41.3 (1999), pp. 212–223.

[81] Alexey Ruchay et al. “The imbalanced classification of fraudulent Bank transactions using machine learning”. In: *Mathematics* 11.13 (2023), p. 2862.

[82] Sumit Saha. *Decision Boundary for Classifiers – An Introduction*. https://medium.com/analytics-vidhya/decision-boundary-for-classifiers-an-introduction-cc67c6d3da0e. 2020.

[83] Yusuf Sahin, Serol Bulkan, and Ekrem Duman. “A cost-sensitive decision tree approach for fraud detection”. In: *Expert Systems with Applications* 40.15 (2013), pp. 5916–5923.

[84] Sterling Seagrave. *Lords of the Rim*. Bantam Press, 1995. URL: https://cir.nii.ac.jp/crid/1130000794287803776.

[85] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. “Query by committee”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 287–294.

[86] Isabella Claudia IC van der Steenhoven. “Stakeholder Involvement during the Development of a Machine Learning Based Anomaly Detection Model for the Anti-Money Laundering Context”. MA thesis. Eindhoven University of Technology, 2024.

[87] Xingzhe Sun et al. “Counterfactual Based Probabilistic Graphs for Explainable Money Laundering Detection”. In: *AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques*. 2025.

[88] Toyotaro Suzumura and Hiroki Kanezashi. *Anti-Money Laundering Datasets*. http://github.com/IBM/AMLSim/. 2021.

[89] Pavlo Tertychnyi et al. “Scalable and imbalance-resistant machine learning models for anti-money laundering: A two-layered approach”. In: *Enterprise Applications, Markets and Services in the Finance Industry: 10th International Workshop, FinanceCom 2020, Helsinki, Finland, August 18, 2020, Revised Selected Papers 10*. Springer. 2020, pp. 43–58.

[90] M Tiwari, A Gepp, and K Kumar. *A review of money laundering literature: the state of research in key areas. Pacific Accounting Review, 32 (2), 271–303*. 2020.

[91] *United States v. Sullivan*. 1927. URL: https://supreme.justia.com/cases/federal/us/274/259/.

[92] Gabriel Vedrenne. *STRs Rise in Western Europe, Fall Sharply in the East*. 2021. URL: https://www.moneylaundering.com/news/strs-rise-in-western-europe-fall-sharply-in-the-east/.

[93] Jarrod West and Maumita Bhattacharya. “Intelligent financial fraud detection: a comprehensive review”. In: *Computers & security* 57 (2016), pp. 47–66.

[94] John G. Goldsworth Wouter H. Muller Christian H. Kalin. “Anti-money laundering–a short history”. In: *Anti-Money laundering: International law and practice*. Vol. 1. John Wiley & Sons, 2007.

[95] Jilei Yang. “Fast treeshap: Accelerating shap value computation for trees”. In: *arXiv preprint arXiv:2109.09847* (2021).

[96] Yan Zhang and Peter Trubey. “Machine learning and sampling scheme: An empirical study of money laundering detection”. In: *Computational Economics* 54.3 (2019), pp. 1043–1063.

[97] Zhi-Hua Zhou and Xu-Ying Liu. “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *IEEE Transactions on knowledge and data engineering* 18.1 (2005), pp. 63–77.

10

APPENDIX

10.1 HISTORY OF MONEY LAUNDERING

Historian Sterling Seagrave, in his book "Lords of the Rim" [84], describes how merchants in ancient China, over 3,000 years ago, took steps to conceal their wealth from rulers seeking to seize it. The merchants devised clever methods to convert their profits and evade regional trade restrictions [67]. While not exactly the same as modern money laundering, these early examples of wealth concealment because of restrictions set a historical precedent for the practices that would later emerge in the global economy. The practice of money laundering has evolved significantly over time. Its progression can be broken down into three key periods: the 1920s, the 1980s, and the present day.

10.1.1 1920s

The origins of money laundering as a recognized concept can be traced back to the rise of organized crime. Historically, it was not perceived as a distinct offense but rather as a secondary consequence of criminal activity. This perspective began shifting in the 1920s, linked to the rise of criminal organizations in America due to the Prohibition era. The Prohibition Era began in 1920 when the 18th Amendment to the U.S. Constitution went into effect. This amendment banned the manufacture, transportation, and sale of liquors with more than 0.5% alcohol. The law was also known as the Volstead Act, named after Congressman Andrew John Volstead of Minnesota. Organized crime boomed as the demand for alcohol rose. These organizations, led by figures such as Al Capone, amassed immense fortunes through bootlegging (the illegal production and sale of goods) [94]. However, depositing large amounts of illicit cash into banks without raising suspicion became increasingly difficult, necessitating money laundering strategies.

A landmark case illustrating this evolving legal stance is *United States v. Sullivan* [91] in 1927. The U.S. Supreme Court ruled that illicit income from bootlegging was subject to federal income tax. Manley Sullivan, a South Carolina bootlegger, argued that his earnings were not taxable since they were derived from illegal activities. The Court's decision established a precedent that all income, regardless of its source, must be reported for taxation. This ruling later played a pivotal role in the prosecution of Al Capone, whose initial conviction was not for bootlegging, but instead for evasion of federal taxes [94].

10.1.2 1980s

During the 1980s, Colombian drug cartels such as the Medellín and Cali cartels generated large profits from cocaine trafficking, necessitating increasingly sophisticated methods to launder billions of dollars annually. PBS Frontline [74] states in a report about the war of drugs that traffickers initially relied on direct and unsophisticated means. Large amounts of U.S. dollars were simply flown back to Colombia aboard the same planes used to smuggle cocaine into the United States. The cash was either converted into pesos through willing Colombian banks or buried on private estates. Some locals even claimed that rivers occasionally became clogged with U.S. dollars after heavy rain revealed hidden caches. Enforcement of the Bank Secrecy Act, which required reporting deposits over $10,000, was widely neglected. In some cases, individuals deposited over $250 million annually into accounts without raising any alarms.

When the law enforcement caught on to the problem and pressured banks to comply with the laws, traffickers conceived different strategies. Smurfing, also known as structuring, in which large sums are broken up into smaller deposits and distributing them across multiple accounts and financial institutions. Simultaneously, traffickers turned to trade-based money laundering. By manipulating invoices and using phantom

shipments (shipments happening only on paper), they could move value internationally under the guise of legitimate trade. Offshore banking was another method that gained popularity in the mid-1980s. Cash was flown to tax havens such as the Bahamas, Aruba, the Cayman Islands, and the British Virgin Islands, where it was deposited with limited oversight. However, as international enforcement intensified, traffickers sought more robust and covert systems.

The most sophisticated method was the Black Market Peso Exchange (BMPE), a laundering system that remains active today. The BMPE evolved as traffickers moved away from traditional banking and took advantage of an existing black market currency exchange used by Colombian businesses to circumvent strict currency controls. Under this system, U.S.-based traffickers hand over dirty drug money to money brokers who take full responsibility for laundering the U.S. dollars. In return, the traffickers receive payment in laundered Colombian pesos. On the Colombian side, legitimate businessmen seeking to purchase American goods pay these money brokers in pesos. The brokers then use the laundered U.S. dollars to pay for the goods, effectively completing the cycle [74].

10.1.3 Present

In the modern era, money laundering has evolved alongside financial and technological advancements. While traditional methods such as trade-based laundering and offshore accounts remain prevalent, new digital tools have made illicit financial flows more sophisticated and harder to trace.

Europol [31] report that cryptocurrencies, initially adopted by cybercriminals, have become a prominent tool for laundering illicit proceeds. However, blockchain analysis on Bitcoin quickly revealed that activity can be traced, prompting criminals to adopt specialized services that enhance anonymity, such as tumblers, which are crypto mixing services to obscure the origin of funds. These services have significantly lowered the technical barriers to entry, enabling broader use among various criminal networks. Although cryptocurrency offers pseudonymity, speed, and borderless transactions, the volume of illicit activity conducted via crypto remains smaller compared to traditional financial crime [13].

Another major revelation in recent years was the Panama Papers scandal of 2016, which exposed how politicians, business leaders, and criminals used offshore accounts and shell companies to hide wealth and evade taxes. The Panamanian law firm Mossack Fonseca, one of the world's leading creators of hard-to-trace companies, trusts, and foundations, provided an unprecedented look into the global scale of financial secrecy. This leak revealed how legal structures were exploited to launder money [45]. While this scandal led to increased regulatory scrutiny and reforms, offshore laundering continues to be a significant issue, as demonstrated by the subsequent Paradise Papers (2017) and Pandora Papers (2021) [45, 70].

The art and antiquities market presents a unique and persistent vulnerability to money laundering and terrorist financing and will be the last driver of money laundering that will be discussed. Due to the high value, subjective pricing, and limited transparency of transactions, art objects can serve as effective vehicles for moving and concealing illicit funds. As the Financial Action Task Force (FATF) [34] highlights, criminals exploit this sector by purchasing high-value artwork with illicit proceeds and subsequently reselling them, often through private sales or intermediaries, to legitimize the gains.

10.2 Supplementary data exploration

Section 10.2.1 provides the data exploration on the bank-level. The subsequent Section 10.2.2 provides a similar exploration on account-level.

10.2.1 Banks

As banks facilitate the money laundering of criminals, there might be a relationship between a characteristic of the bank and the illicit activity inside it. This section is dedicated to understanding the behavior of different types of banks.

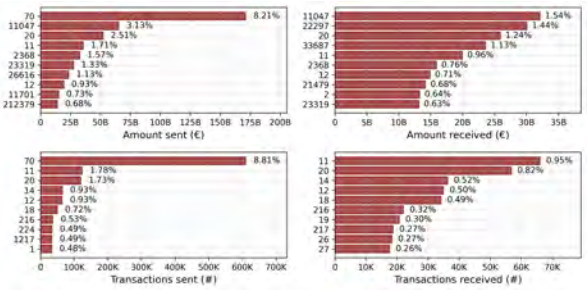
Some initial terminology is required to get a clear picture of the activity of different types of banks. Banks involved in non-zero transactions as the paying party are referred to as paying banks, while those on the receiving end with non-zero transactions are called receiving banks. Banks with exclusively internal transactions have no exchanges with other banks, whereas exclusively external banks conduct transactions only with other banks and none within themselves. Table 14 shows the number of banks and illicit activity of each of these types. Receiving banks, which there are about twice as few of as paying banks, exhibit a relatively

high proportion of illicit activity. Interestingly, any bank that is exclusive in one of the types (paying, receiving, internal, external) has no illicit activity, which is unexpected since they count for about 75% of the total number of banks. While the number of banks with illicit activity is the same for paying and receiving banks, it should be noted that there are 968 sending banks that have sent an illicit transaction and 1,187 receiving banks that have received an illicit transaction.

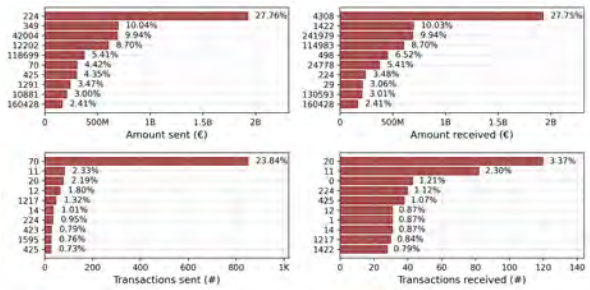
Bank type	Banks	Banks with illicit activity	Laundering share
Paying banks	41,814	1,474	3.525%
– Exclusively paying banks	20,227	0	0%
Receiving banks	21,588	1,474	6.828%
– Exclusively receiving banks	1	0	0%
All banks	41,815	1,474	3.525%
– Exclusively internal	9,642	0	0%
– Exclusively external	1,878	0	0%

Table 14: Unique number of banks and money laundering activity for different types of banks.

To understand the major banks more, the 10 banks with the most amount sent, amount received, transactions sent, and transactions received are shown in Figure 27a. Bank '70' has a significant share in amount sent, however the amount is not shown in the amount received, which is reason to investigate if there is a relationship between money laundering and bank. Figure 27b illustrates the top 10 banks based on the money illicit activity. Also for money laundering it is the case that there exist dominant banks. Bank '224' sends more than a quarter of the total illicit funds, and Bank '4308' receives about the same amount. Bank '70' sends more than 800 illicit transactions, but on average they contain small transactions amounts, as this equates to about 4.5% of the amount sent.



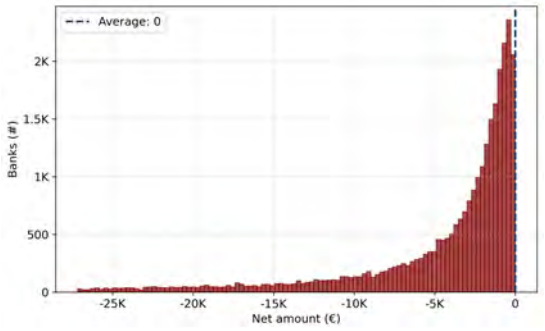
(a) Total transactions (legitimate and illicit).



(b) Illicit transactions.

Figure 27: Top 10 banks by total amount and total transactions (a), with total amount sent (top-left), amount received (top-right), transactions sent (bottom-left), and transactions received (bottom-right). Similarly, top 10 banks based on illicit amount and illicit transactions (b). Percentages relative to the total across all banks are shown at the end of the bars.

The discrepancy between the money going in and out of the banks is shown in Figure 28. Outliers are distorting the scale of Figure 28a, making it harder to visualize the data effectively, necessitating a trim of extreme values. This is illustrated by the large net amounts in Table 28b. These outliers could potentially raise concerns regarding the liquidity management of the banks in the virtual world. Interestingly, Figure 28a shows a smooth histogram for negative net amounts, while no such histogram exists for positive net amounts.



(a) Distribution of net amounts across banks over total time span. A cut between 5% and 95% is made to hide outliers; the mean of the lower 5% quantile is -9.45B and the upper 95% quantile is 4.53B. As expected, the average across all banks is 0 since all money sent is received in a closed economic system.

Top 10		Bottom 10	
Bank	**Net amount (€)**	**Bank**	**Net amount (€)**
22297	30.1B	41720	-171B
33687	23.6B	23	-33.3B
21479	14.1B	901	-26.4B
147006	11.9B	1606	-23.2B
25526	11.4B	1246	-16.8B
126631	11.3B	12	-15.7B
16	9.78B	1200	-14.5B
22435	8.36B	1339	-11.9B
119699	8.06B	1489	-10.7B
2	7.90B	842	-9.77B

(b) Top and bottom 10 net amounts by bank.

Figure 28: Visualization of net amounts across banks (a) and top/bottom 10 net amounts (b).

Banks exhibiting large net discrepancies (such as Bank '22297' and Bank '41720' in Table 28b) may process a higher absolute number of illicit transactions than smaller institutions. However, this does not necessarily imply a greater relative involvement in illicit activity. For meaningful comparison, it is essential to consider illicit activity in proportion to overall transaction volume or amount. Two banks with the same number or value of illicit transactions may differ substantially in risk profile if their total transaction volumes differ. To capture this, we define two normalized metrics: the Illicit Transaction Rate (ITR) and the Illicit Amount Rate (IAR). These quantify the proportion of a bank's transactions or transaction value that is illicit, and are calculated as follows:[9]

$$\text{Illicit transaction rate (ITR)} = \frac{\text{Illicit transactions}}{\text{Total transactions}}, \tag{22}$$

$$\text{Illicit amount rate (IAR)} = \frac{\text{Illicit amount}}{\text{Total amount}}. \tag{23}$$

Figure 29 plots the net amount against the ITR for banks sending and receiving illicit transactions, but shows no clear pattern based on the amount going in and out of the bank and the ITR. It does show that the net amounts for banks that are involved in money laundering are more symmetrically distributed than Figure 28a. Additionally, it is shown that most banks have an illicit transaction rate between 0.00% and 0.0025%.

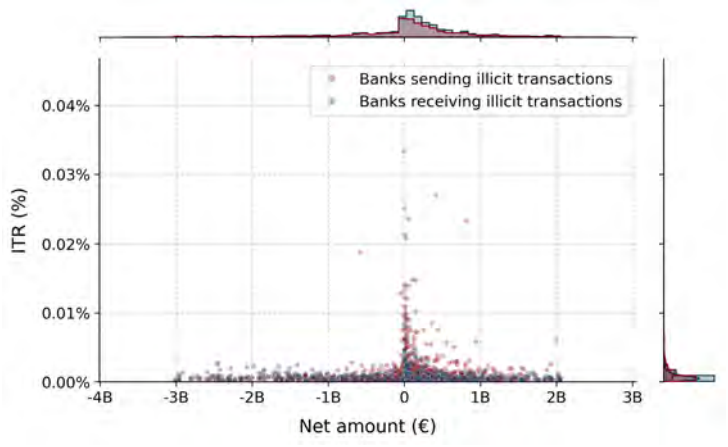
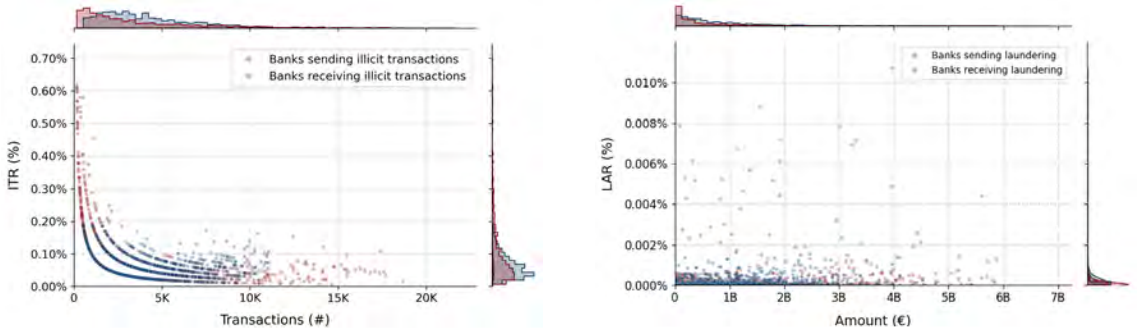


Figure 29: Distribution of net amounts and non-zero ITR across banks. A cut for the net amounts between the 5% and 95% quantiles is done and only banks with ITR between 0 and 5% is shown to improve clarity. The datapoints are categorized into the laundering rates of banks for both sending and receiving illicit transactions. Note that if bank received and send illicit transactions, a single bank is represented as two labels with the same net amount.

[9]Note that ITR and IAR differ from the "illicit share" in Table 14, which indicates the proportion of banks with any non-zero illicit activity.

Investigating the relationship between banks and illicit activity further, it is interesting to see how the number of transactions and amount impacts the laundering rate. Figure 30 indicates that there is a noticeable pattern in the number of transactions 30a, in contrast with the transaction amount 30b. The number of transactions exhibits an inverse relationship with the laundering rate, suggesting that banks with higher number of transactions tend to have lower rates. Moreover, there appears to be approximately five distinct categories, independent of whether the banks are paying or receiving banks, expressed by the discrete laundering rate levels corresponding to different transaction totals. This distinction becomes less apparent for low and high transactions totals, where the levels are tighter together and possess fewer data points. However, recalling Equation 22, one can see that Figure 30a shows distinct levels because the illicit transactions numerator takes integer values. Figure 30b does not show such a leveled pattern as transaction amounts can also take decimal values.



(a) Illicit activity based on total transactions.

(b) Illicit activity based on amount.

Figure 30: Scatter plot (a) shows the relationship between the transaction total and non-zero laundering rates. Scatter plot (b) shows the relationship between transaction amount and non-zero laundering rates. For both plots apply: each dot represents a bank, red for banks sending illicit transactions and blue for banks receiving illicit transactions and a 95%-quantile exclusion is applied to remove outliers (the outliers do not show unusual behavior, only change the scale).

10.2.2 Accounts

Criminals exploit banks during their activities, so it is interesting to understand the typical behavior of accounts, and even more to investigate their suspicious activities.

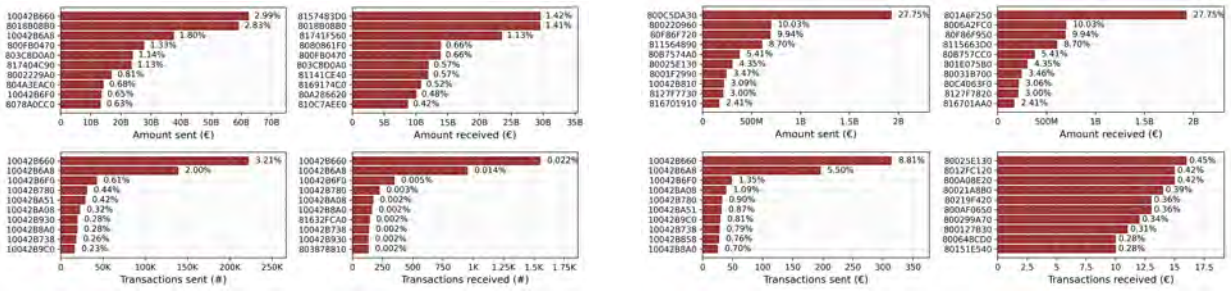
Table 15 shows the overall behavior of accounts based on their type. Similar to the banks, accounts involved in non-zero transactions as the paying party are referred to as paying accounts, while those on the receiving end with non-zero transactions are called receiving accounts. Accounts with only internal transactions have no exchanges with other accounts, whereas exclusively external accounts conduct transactions only with other accounts and none within themselves. In the synthetic dataset there exist also no transactions that debit the internal transactions, so it is as if money is generated out of nothing for these internal transactions.

Account type	Accounts	Accounts with illicit activity	Laundering share
Paying accounts	681,281	5,139	0.754%
– Exclusively paying accounts	129,727	90	0.069%
Receiving accounts	576,176	5,214	0.905%
– Exclusively receiving accounts	24,622	165	0.670%
All accounts	705,903	5,304	0.751%
– Exclusively internal	129,126	0	0%
– Exclusively external	44,878	62	0.138%

Table 15: Unique number of accounts and money illicit activity for different types of accounts.

Figure 31 presents the top 10 accounts by total and illicit transaction volume and amount, both sent and received. Notably, there is no clear correspondence between the rankings in total activity (Figure 31a) and illicit activity (Figure 31b). For example, while Account '10042B660' ranks first in total amount sent, it does not occupy such a position in illicit amount sent. In terms of total transactions sent, Accounts '10042B660' and '10042B6A8' stand out, sending a combined 350K transactions. These two accounts also rank highest in total transactions received, though with substantially lower volumes. When focusing on illicit amounts,

the dynamic shifts. Account '800C5DA30' is responsible for originating over 25% of the total illicit amount sent, while Account '801A6F250' receives more than 25% of the illicit funds, indicating a concentration of suspicious activity among a few key accounts.



(a) Total transactions (legitimate and illicit). (b) Illicit transactions.

Figure 31: Top 10 banks by total amount and total transactions (a), with total amount sent (top-left), amount received (top-right), total transactions sent (bottom-left), and total transactions received (bottom-right). Similarly, top 10 banks based on illicit amount and total transactions (b). Percentages relative to the total across all banks are shown at the end of the horizontal bars.

To visualize interactions between criminal accounts, a graph is constructed with paying accounts as sources, receiving accounts as sinks, and transactions as directed edges, shown in Graph 32. The two dominant Accounts '10042B660' and '10042B6A8' stand out based on the total number of illicit transactions send, accounting for over 10% of all illicit transactions. Interestingly, a large number of illicit transactions involve only two accounts and remain disconnected from the rest of the network.

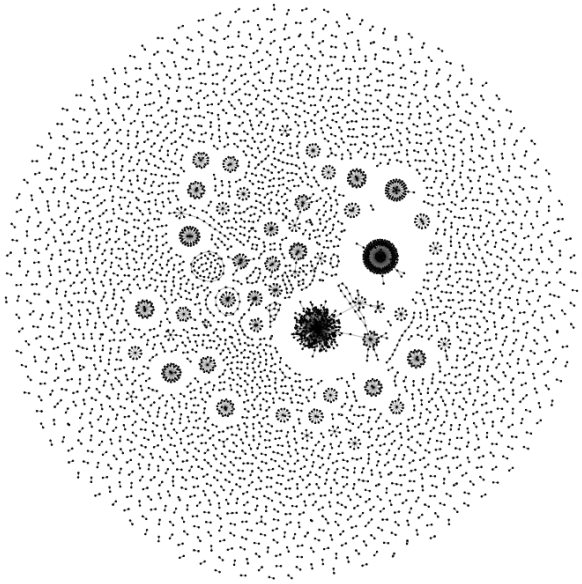
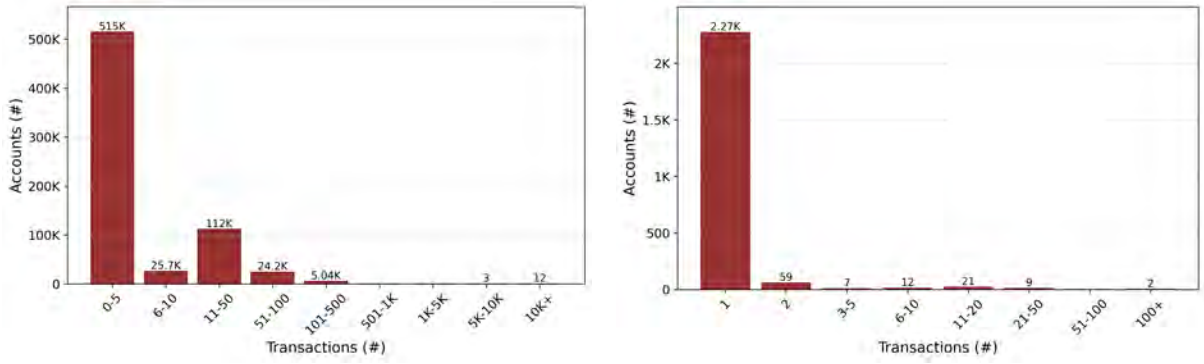


Figure 32: Network of illicit transactions. Nodes represent accounts, while edges depict the flow of illicit transactions between accounts. The dominant 2 Accounts '10042B660' and '10042B6A8' have 314 (8.81% of total) and 196 (5.50% of total) illicit transactions, respectively.

To have more insight in the distribution of illicit transactions, Figure 33a shows the number of accounts with a certain range of number of transactions. Most accounts (515K) have 0-5 transactions in the 16 days time span. A small number of accounts (12) have more than 10,000 transactions, suggesting that these accounts may be corporate accounts or payment processing accounts. Figure 33b demonstrates that most money laundering does not happen in a short time span and/or is done by multiple accounts. The bulk of transactions comes from the 2.27K accounts that only made a single illicit transaction in that time, which

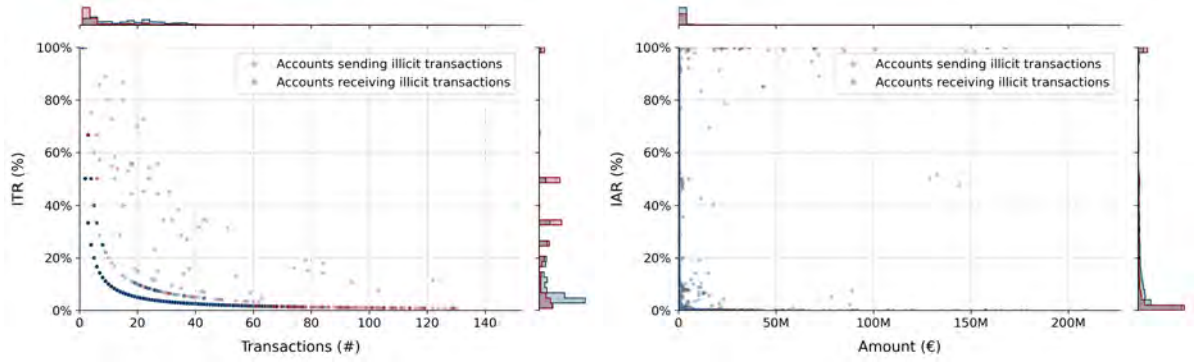
are also seen in the coupled transactions in Figure 32. As expected, the two accounts with over 100 illicit transactions are the same dominant entities mentioned earlier.



(a) All accounts. (b) Accounts involved in illicit transactions.

Figure 33: The number of accounts for different ranges of total transactions, based on all accounts (a) and criminal accounts (b).

Figure 34 shows similar behavior as the banks, where Figure 34a shows a leveled pattern due to the integer nature of total transactions. Figure 34b again shows no relationship between the amount send/received and laundering rate.



(a) Illicit activity based on total transactions. (b) Illicit activity based on amount.

Figure 34: Scatter plot (a) showing the relationship between total transactions and non-zero laundering rates. Scatter plot (b) showing the relationship between transaction amount and non-zero laundering rates. For both plots apply: each dot represents an account, red for paying accounts and blue for receiving accounts and a 99%-quantile exclusion is applied to remove outliers (the outliers do not show unusual behavior, only change the scale).

The marginal distribution of accounts based on the ITR from Figure 34 shows distinct behavior, as the laundering rates seem to cluster around specific values. Figure 35 zooms into that distribution, and indeed some laundering rates are disproportionately common. Interestingly, both accounts sending and receiving illicit transactions exhibit this behavior. However, due to the undefined bulk between 0% and 10% ITR, this is not an exploitable pattern for feature engineering.

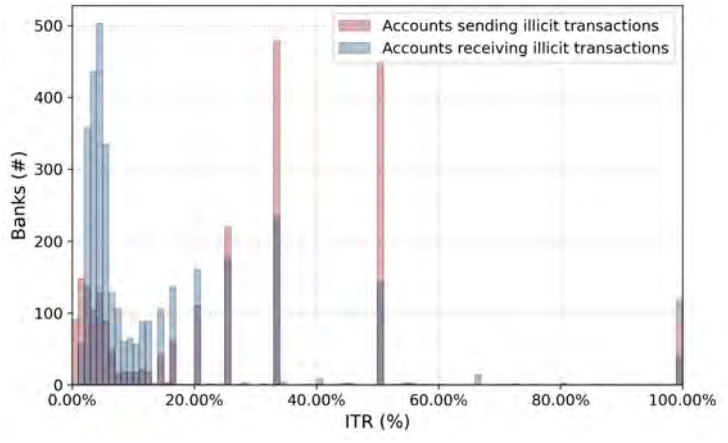


Figure 35: Distribution of illicit transaction rates for the number of accounts sending and receiving illicit transactions.

10.3 FEATURES

Feature Name	Description
IS LAUNDERING	Whether the transaction is money laundering (1) or not (0)
PAYING BANK	Bank initiating the payment
PAYING ACCOUNT	Account sending the money
RECEIVING BANK	Bank receiving the payment
RECEIVING ACCOUNT	Account receiving the money
CYCLE DETECTED	A cycle back to account within 15 transactions is detected
DAYS ACTIVE PAYING ACCOUNT	Active days for the paying account
DAYS ACTIVE RECEIVING ACCOUNT	Active days for the receiving account
UNIQUE DESTINATIONS 4D	Unique recipients from the sender in the last 4 days
UNIQUE DESTINATIONS 8D	Unique recipients from the sender in the last 8 days
UNIQUE ORIGINS 4D	Unique senders to the receiver in the last 4 days
UNIQUE ORIGINS 8D	Unique senders to the receiver in the last 8 days
CRIMINALS IN NETWORK	Number of unique accounts that the paying and receiving account have had illicit transactions with
PAYMENT FORMAT ACH	Transaction used the automated clearing house
PAYMENT FORMAT BITCOIN	Transaction used Bitcoin
PAYMENT FORMAT CASH	Transaction used cash
PAYMENT FORMAT CHEQUE	Transaction used a cheque
PAYMENT FORMAT CREDIT CARD	Transaction used a credit card
PAYMENT FORMAT WIRE	Transaction used a wire transfer

Feature	Description
HOUR	Hour of the day the transaction occurred
DAY	Day of the month
MONTH	Month of the year
YEAR	Year
WEEKDAY	Day of the week
IS WEEKEND	Whether the transaction happened on a weekend
DAWN	Transaction occurred during dawn hours (hour 0–5)
MORNING	Transaction occurred during morning hours (6–11)
MIDDAY	Transaction occurred during midday (12–17)
EVENING	Transaction occurred during evening/night (18–23)
AMOUNT (EUR)	Transaction amount in Euros
UK POUND	Binary variable for currency indicating if sender and/or receiver used British Pounds
MEXICAN PESO	Indicator for Mexican Pesos
SAUDI RIYAL	Indicator for Saudi Riyals
RUPEE	Indicator for Indian Rupees
EURO	Indicator for Euros
YUAN	Indicator for Chinese Yuan
CANADIAN DOLLAR	Indicator for Canadian Dollars
BRAZIL REAL	Indicator for Brazilian Real
US DOLLAR	Indicator for US Dollars
SHEKEL	Indicator for Israeli Shekel
SWISS FRANC	Indicator for Swiss Francs
YEN	Indicator for Japanese Yen
RUBLE	Indicator for Russian Ruble
AUSTRALIAN DOLLAR	Indicator for Australian Dollars
BITCOIN	Indicator for Bitcoin
INTRA-CURRENCY	Binary variable indicating if sender and receiver used same currencies

Table 16: Features and their descriptions. The IS LAUNDERING feature is the target variable.

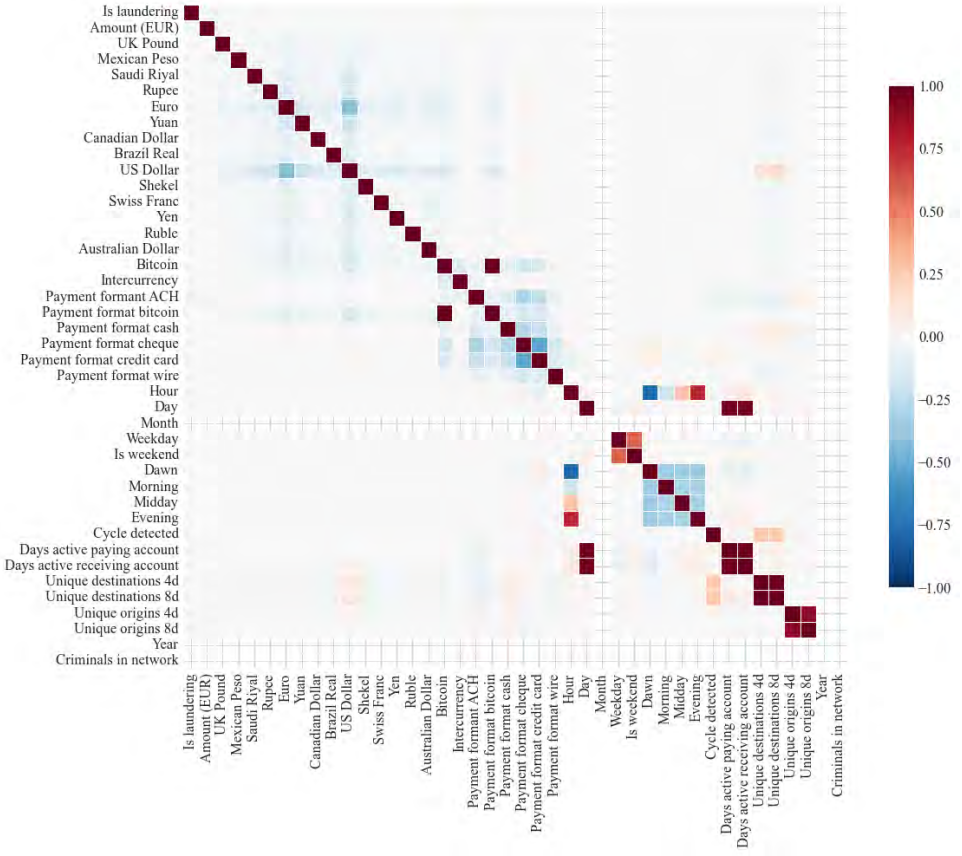


Figure 36: Correlation matrix of all features before training. The CRIMINALS IN NETWORK feature updates its values during training.

10.4 COST OF ALERT INVESTIGATION

Investigating an alert costs a certain amount due to the time analysts are investigating. The cost can be estimated by first understanding the operational profile of an average bank. This includes the distribution of alerts, their classification outcomes.

10.4.1 Profile of average bank

The calculation of the cost starts with estimating the distribution of confusion matrix elements is considered the profile of the bank, as they describe the cost and (implicit) benefit gained from transaction monitoring.

The number of transactions handled by an average bank is estimated first. According to the European Banking Federation [29], there were 5,075 credit institutions in the EU-27 in 2022, including commercial and savings banks. In the same year, European Central Bank [30] reports 29.4 billion retail payment transactions worth €23.5 trillion across the EU-27. Assuming that the transactions are done by these 5,075 institutions, this yields an average of approximately 5.8 million transactions per bank annually.

Vedrenne [92] shows that in 2021 there were about 1,590,000 SARs filed. Since an individual illicit transaction rarely signals a complete laundering scheme, SARs typically bundle contextually related transactions. FIU-Nederland [36] reports an average of ∼4.5 transactions per SAR (80,578 transactions from 17,848 SARs in 2023). For the sake of estimation, we assume each SAR represents 10 distinct illicit transactions, although in reality not all bundled transactions may be independently illicit. This leads to an estimated 15.9 million reported illicit transactions. Spread across 5,075 banks, this yields an average of 3,133 transactions flagged as illicit per bank annually.

Chau and Dijck Nemcsik [14] notes that the false positive rates often exceed 90%, implying 28,197 false positives. From these values an alert rate of 0.54% can be computed. This substantiates the claim from Redhead [78] who explain that the conversion rate of transactions to SAR drops below 0.005%, or approximately 0.5% with 10 transactions per SAR and a true positive rate of 10%.

The number of missed illicit transactions is inherently unknown due to the concealed nature of money laundering schemes. But based on an interview with a model auditor of a large Dutch bank, where it is stated that the bank aims to catch about 8% of the illicit transactions of the test dataset. With the assumption that this goal is the same for an average bank, the total number of illicit transactions computes to 39,163 and the number of missed illicit transactions to 36,030.

Europol [32] reports that only about 10% of SARs are further investigated, a trend that has persisted since 2006. Moreover, they state that only 1% of criminal proceeds are successfully confiscated. Assuming that the 1% of criminal proceeds confiscated corresponds to 1% of all money laundering activity, and that these are discovered through the 10% of SARs that are actually investigated, we infer that only 1% of true illicit transactions are caught. This implies a 99% miss rate, and for 313 detected true positives, this yields an estimated total of 31,300 illicit transactions, meaning 30,987 go undetected.

A confusion matrix can be constructed for an average bank from the perspective of the bank and also from the perspective of a FIU, which renders only 10% of the SARs illicit/useful.

5,732,640 (98.84%)	28,197 (0.49%)
36,030 (0.62%)	3,133 (0.05%)

(a) Confusion matrix of an average bank from the perspective of the bank.

5,737,683 (98.93%)	31,017 (0.53%)
30,987 (0.53%)	313 (0.005%)

(b) Confusion matrix of an average bank from the perspective of the FIU.

Table 17: Confusion matrices based on different perspectives of a true positive. TP as seen by banks are transactions reported in a SAR, TP as seen by the FIU are transactions that are further investigated by the FIU.

The similar values for FN of both matrices indicates that the recall goal of 8% of the test set makes sense with the 1% actually detected illicit transactions by the FIU. This can be checked with the approximation that 8% of transactions is caught by the bank, and 10% of these caught transactions are further investigated as seen by the FIU, resulting in 0.8% of the illicit activity being caught by banks. The banks perspective will be used throughout this research, as the FIU does not give feedback to the bank about which SARs are further used and which are not.

10.4.2 Cost of alert investigation

A well defined indicator of the average cost of alert investigation is not well-established in the literature. DataRobot [20] states that about \$30 ∼ \$70 to investigate one alert, however the source of this range is not clear. Rather than relying on potentially opaque estimates, a more defensible approach is to derive the cost per alert by dividing the total financial crime compliance (FCC) expenditures by the total number of alerts an average bank handles annually. This approach hinges on two main components: the number of alerts a bank receives annually and the overall costs for investigating these alerts. Following the logic to determine the profile of an average bank, the former is estimated to 28,197 FP and 3,133 TP (Table 17). To estimate the cost of alert investigation, the average FCC expenditure per bank in the EEA is considered and comes down to approximately €14 million annually, as reported by European Banking Authority [28]. A 2024 study by LexisNexis Risk Solutions [56], which surveyed 254 compliance professionals, found that 7.1% of FCC budgets are allocated to alert investigation and decision-making. This translates to around €994K spent per bank each year specifically on alert investigation. The cost of handling a single alert is therefore approximately €31.73. This approximation coincides with the \$30 ∼ \$70 range DataRobot [20] estimated.

10.5 Optimal hyperparameters

Query strategy	Parameter	€50	€1K	€5K	€10K	€20K	€30K	€40K	€50K	€100K	€1M
Random	τ	1.00	0.17	0.14	0.06	0.05	0.04	0.04	0.02	0.02	0.00
	T	250	100	50	250	150	100	100	200	50	200
	D	8	10	8	6	6	6	6	6	6	2
Uncertainty	τ	1.00	0.22	0.15	0.14	0.03	0.02	0.05	0.13	0.00	0.00
	T	100	200	250	100	200	100	200	50	200	200
	D	16	8	6	6	6	14	6	4	4	4
Query by committee	τ	1.00	0.23	0.15	0.11	0.11	0.10	0.11	0.09	0.10	0.00
	T	300	200	300	150	200	50	100	250	300	100
	D	2	6	6	6	6	6	6	8	2	16
Isolation forest	τ	1.00	0.22	0.15	0.15	0.07	0.06	0.05	0.07	0.04	0.00
	T	150	150	300	150	50	50	100	250	50	50
	D	8	8	6	6	6	6	6	2	6	16
Elliptic envelope	τ	0.26	1.00	0.12	0.18	0.08	0.09	0.12	0.07	0.03	0.00
	T	150	200	200	200	50	200	200	150	200	100
	D	10	2	8	4	8	6	2	6	6	12
SHAP-guided	τ	1.00	1.00	0.10	0.10	0.01	0.02	0.05	0.01	0.01	0.00
	T	50	150	250	300	50	300	150	250	250	100
	D	10	8	6	6	8	6	6	6	6	16
Passive	τ	1.00	1.00	0.22	0.20	0.15	0.15	0.15	0.15	0.11	0.04
	T	50	100	100	150	50	250	50	250	150	300
	D	2	8	16	6	6	4	6	6	4	8

Table 18: Optimal hyperparameters for each query strategy and several b values. Not all b values are shown for clarity.

10.6 Performance comparison

Figure 37 presents the changes in recall, precision, and true negative rate (Δ metrics) due to model optimization, evaluated across the query strategies and various benefit values. Optimization was performed using the validation set and includes both feature selection and hyperparameter tuning. The Δ metrics are computed as the difference between the performance of the optimized model and that of the non-optimized model. All reported values are based on the test set to ensure unbiased performance estimation.

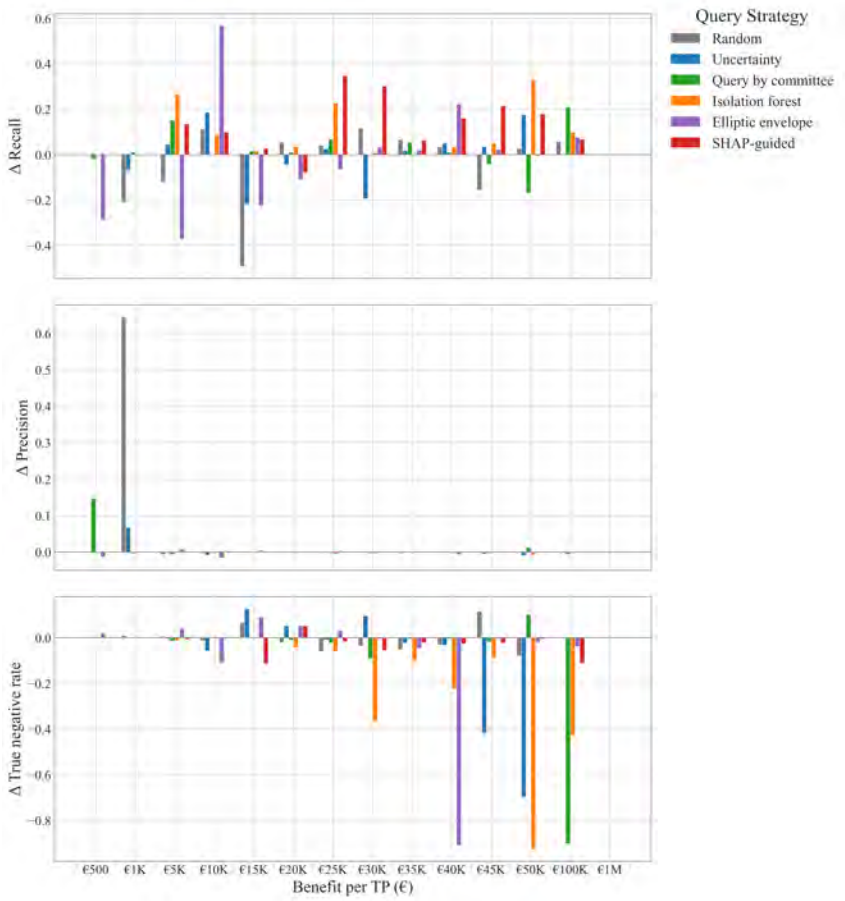


Figure 37: Improvement in recall, precision, and true negative rate for each query strategy and selected values of benefit per TP.