MARMARA UNIVERSITY
FACULTY OF BUSINESS ADMINISTRATION

TRANSACTION-BASED MONEY LAUNDERING DETECTION USING MACHINE LEARNING

Hilal Çalışkan
Nefise Hatun Demir
Ravza Nur Şişik

CONTENT

1. Definition of The Problem

FinkoBank is a leading global financial institution trusted by millions. Every day, it processes millions of transactions across borders — handling everything from personal remittances to corporate transfers in dozens of currencies.

But recently, something has gone wrong: A regulatory audit flagged multiple transactions as suspicious — transactions that had already been cleared by the bank's current anti-money laundering (AML) system. As headlines broke out and legal departments scrambled, one thing became clear:

FinkoBank had been used to launder money, right under its nose.

The Challenge: Modern Money Laundering Is Invisible
Money launderers are no longer reckless or easily caught. Instead, they:

- Send small amounts that fly under traditional rule-based thresholds,
- Disguise their transfers with currency mismatches or cross-country hops,
- Exploit quiet accounts that haven't raised suspicion before.

The Mission: Build an Intelligence-Led Detection System
We were granted access to FinkoBank's transaction history, comprising millions of entries. Some were already labeled as laundering cases after long investigations. Most were clean. But buried inside this massive dataset were subtle patterns only machine learning could catch.

Our task was to:

- Predict if a transaction is likely to be part of a laundering attempt,
- Learn from complex patterns like currency mismatches, sender/receiver geography, or unusual transaction volumes,
- And do so in a way that balances precision and recall, minimizing false alarms without missing real threats.

Why This Is Difficult
This wasn't just another classification problem. It was:

- A needle in a haystack: less than **0.1%** of transactions were confirmed laundering.
- **Imbalanced** data that could fool any model into predicting "clean" every time.

- Categorical and textual fields like currencies and locations, which needed meaningful encoding.

- A moving target: laundering methods **change** constantly.

The Impact of Getting It Right

If successful, the model would:

- Help FinkoBank avoid regulatory fines worth millions,
- Reduce the workload of human analysts by prioritizing the riskiest transactions,
- Restore customer trust and comply with global AML standards,
- And make the bank a leader in AI-powered compliance.

This was no longer a data project.
It was a **battle** against organized financial crime — fought with intelligence, algorithms, and responsibility.

1. Data Collection & Examination of Data

2.1 Data Source and Collection

The dataset utilized in this project is the SAML-D (Synthetic Anti-Money Laundering Dataset), published as part of the 2023 IEEE International Conference on e-Business Engineering (ICEBE)[1]. This synthetic dataset was developed to address critical limitations in the field of Anti-Money Laundering (AML), where access to real transactional data is heavily restricted due to legal, privacy, and ethical concerns.

The authors, designed the SAML-D dataset to replicate the complexity and diversity of real-world financial transactions. It includes both legitimate and suspicious transactions, with carefully constructed 28 typologies, consisting of 11 normal and 17 suspicious types. These were selected based on academic literature, real-world transaction behavior, and expert consultations with AML professionals.

The dataset is publicly accessible via Kaggle and comprises 9,504,852 transactions, of which approximately 0.1039% are labeled as suspicious. The highly imbalanced nature of the dataset mirrors the real-world distribution of money laundering cases and presents a relevant challenge for model development.

2.2. Dataset Overview and Features

2.2.1 Features (Columns) Description

#	Column	Dtype	Label
1	Time	object	Transaction time.

2	Date	object	Transaction date.
3	Sender_account	int64	Sender’s account ID.
4	Receiver_account	int64	Receiver’s account ID.
5	Amount	float64	Transaction amount.
6	Payment_currency	object	Currency sent.
7	Received_currency	object	Currency received.
8	Sender_bank_location	object	Country of sender’s bank.
9	Receiver_bank_location	object	Country of receiver’s bank.
10	Payment_type	object	Type of payment (e.g., cash, credit).
11	Is_laundering	int64	Label: 1 if suspicious, 0 if normal.
12	Laundering_type	object	Toll free service

2.2.2 Descriptive Statistics

The following table has some basic summary statistics for numerical features:

Column	count	mean	std	min	max	range
Sender_account	9,504,852	5.01E+09	2.89E+09	9,018	9.999987E+09	9.999978E+09
Receiver_account	9,504,852	5.01E+09	2.88E+09	9,018	9.999971E+09	9.999962E+09
Amount	9,504,852	8,763.0	25,615.0	3.73	12,618,500	12,618,497
Is_laundering	9,504,852	0.00104	0.0322	0	1	1
Sender_account	9,504,852	5.01E+09	2.89E+09	9,018	9.999987E+09	9.999978E+09

2.3 Preliminary Data Examination

- **Class Imbalance:**

Out of ~9.5 million transactions, only ~9,880 are labeled as laundering (Is_laundering = 1). This results in a class distribution of approximately 0.10% positive class, confirming the need for balancing techniques such as under-sampling, SMOTE, or anomaly detection strategies.

- **No Missing Data**

An initial analysis of the dataset revealed no missing or null values across any of the features. Therefore, no imputation or removal of incomplete records was necessary.

- **Data Quality:**

The dataset is synthetically generated, meaning there are no missing values in key features. However, potential data leakage and overfitting risks must be carefully managed, especially during feature engineering.

- **High Cardinality in Categorical Fields:**

Sender_account and Receiver_account have extremely high cardinality and are not directly informative without graph-based modeling.

Payment_type, Sender_bank_location, and Currency_Pair are more manageable and were selected for encoding.

- **Currency Mismatch Feature:**

A significant number of transactions involve different Payment_currency and Received_currency, which may indicate laundering attempts via currency conversion.

- **Behavioral Patterns:**

High transaction amounts and use of specific countries (like UAE, Turkey, Mexico) appear more frequently in suspicious transactions.

3. Preparation of the Data for Analysis

Data pre-processing is a fundamental step in building any machine learning pipeline, particularly in domains like Anti-Money Laundering (AML), where the datasets are large, imbalanced, and complex. In this study, we applied a sequence of systematic preprocessing techniques to ensure that the data is clean, informative, and suitable for robust model training.

3.1 Libraries and Tools

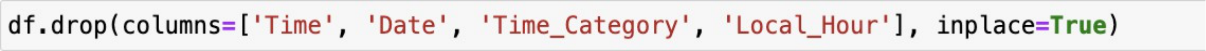
Python libraries and tools were employed throughout the project for data preprocessing, model development, and evaluation**.**

#	Library/Tool	Purpose
1	pandas	Data loading and manipulation
2	numpy	Numerical operations
3	matplotlib.pyplot	Visualization of performance metrics
4	StandardScaler	Feature normalization
5	train_test_split	Splitting the dataset
6	SMOTE	Balancing class distribution

7	LogisticRegression	Linear classification model
8	RandomForestClassifier	Tree-based ensemble model
9	XGBClassifier	Gradient boosting classifier
10	accuracy_score, f1_score, etc.	Model performance evaluation
11	confusion_matrix	Classification result breakdown

3.2 Data Cleaning and Reduction

After importing the dataset, the following columns were removed, as they were either redundant or not directly useful for predictive modeling in their raw form. Although potentially informative, time-related features may be explored in future iterations for temporal pattern analysis.



```
df.drop(columns=['Time', 'Date', 'Time_Category', 'Local_Hour'], inplace=True)
```

3.3 Feature Engineering

To improve the model's ability to detect complex and subtle patterns in transactional behavior, several new features were engineered based on domain knowledge. These derived features aim to reveal underlying risk indicators that may not be apparent from the raw data.

Currency Mismatch

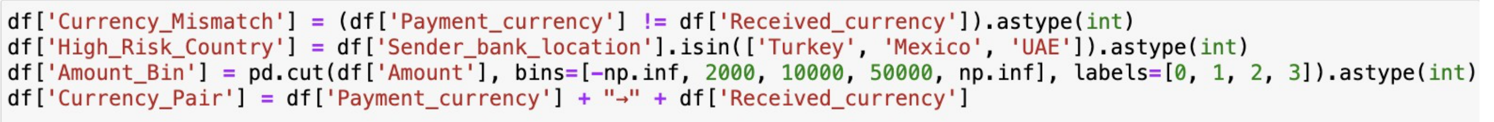
This binary feature flags transactions where the payment and received currencies differ, which may indicate suspicious currency conversion practices commonly associated with money laundering.

High-Risk Country Flag

This feature identifies transactions originating from countries that are frequently referenced in AML risk assessments as high-risk jurisdictions.

Amount Binning

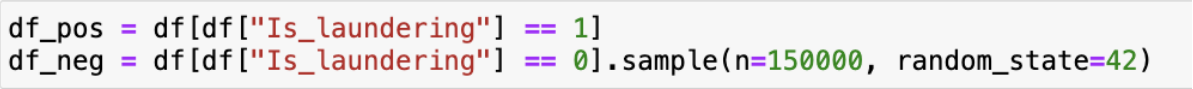
The transaction amount was grouped into bins to reduce skewness and improve the model's ability to detect outliers and anomalous payment behaviors.



```
df['Currency_Mismatch'] = (df['Payment_currency'] != df['Received_currency']).astype(int)
df['High_Risk_Country'] = df['Sender_bank_location'].isin(['Turkey', 'Mexico', 'UAE']).astype(int)
df['Amount_Bin'] = pd.cut(df['Amount'], bins=[-np.inf, 2000, 10000, 50000, np.inf], labels=[0, 1, 2, 3]).astype(int)
df['Currency_Pair'] = df['Payment_currency'] + "→" + df['Received_currency']
```

3.4 Class Imbalance Handling

As only ~0.1% of all transactions are labeled as laundering, a sub-sample was created for computational efficiency and to mitigate extreme imbalance. All positive samples were retained, and 150,000 negative samples were randomly selected.

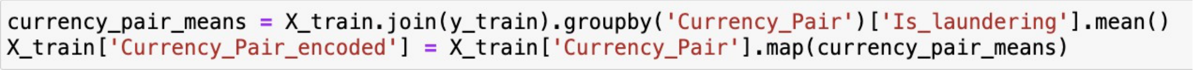


```
df_pos = df[df["Is_laundering"] == 1]
df_neg = df[df["Is_laundering"] == 0].sample(n=150000, random_state=42)
```

3.5 Encodıng Categorical Variables

- **Target Encoding for Currency_Pair**

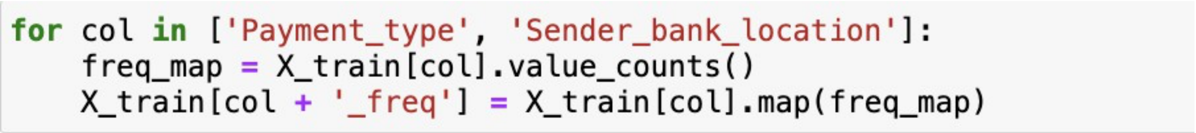
Each unique currency pair was encoded based on the average Is_laundering rate observed in the training set. This method captures the likelihood of laundering associated with specific currency flows.



```
currency_pair_means = X_train.join(y_train).groupby('Currency_Pair')['Is_laundering'].mean()
X_train['Currency_Pair_encoded'] = X_train['Currency_Pair'].map(currency_pair_means)
```

- **Frequency Encoding for Payment_type and Sender_bank_location**

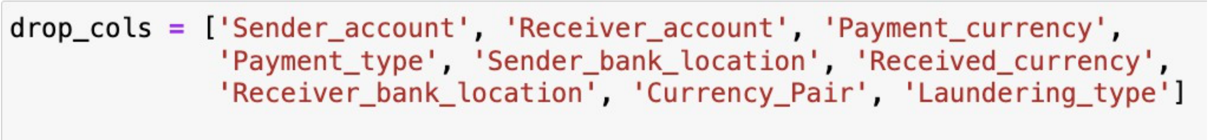
Frequency encoding was applied to the Payment_type and Sender_bank_location features by replacing each category with its occurrence count in the training set. This approach retains distributional information while avoiding the high dimensionality of one-hot encoding, making it especially suitable for features with many unique categories.



```
for col in ['Payment_type', 'Sender_bank_location']:
    freq_map = X_train[col].value_counts()
    X_train[col + '_freq'] = X_train[col].map(freq_map)
```

3.6 Feature Pruning

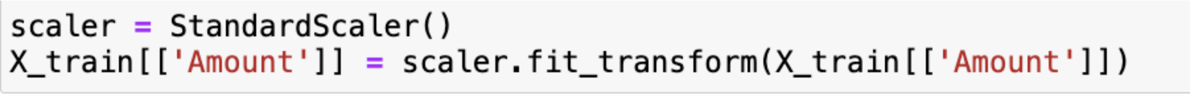
To reduce complexity and improve model performance, feature pruning was applied to eliminate columns that do not contribute meaningful information to the classification process.



```
drop_cols = ['Sender_account', 'Receiver_account', 'Payment_currency',
             'Payment_type', 'Sender_bank_location', 'Received_currency',
             'Receiver_bank_location', 'Currency_Pair', 'Laundering_type']
```

3.7 Feature Scaling

Scaling was essential because some machine learning algorithms, such as Logistic Regression, are sensitive to differences in feature magnitude. Normalization ensures that the Amount feature contributes proportionally to the model and prevents it from dominating the learning process due to its larger scale.
The Amount feature, which varies widely in scale, was normalized using StandardScaler:



```
scaler = StandardScaler()
X_train[['Amount']] = scaler.fit_transform(X_train[['Amount']])
```

3.8 Synthetic Minority Oversampling (SMOTE)

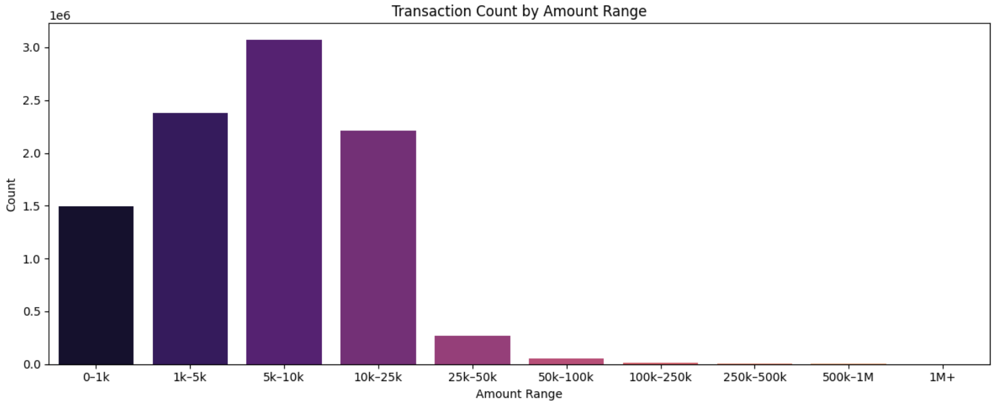
Despite sub-sampling, the dataset remained slightly imbalanced. To further balance the classes and improve recall performance, SMOTE was applied.

4. Exploratory Analysis

4.1 Transaction Count by Amount Range

This bar chart illustrates the distribution of transaction volumes across predefined amount intervals. The majority of transactions are concentrated within the **5k–10k**, **1k–5k**, and **10k–25k** ranges, reflecting a high frequency of mid-value transfers. In contrast, transactions above **50k** are relatively rare, and those exceeding **250k** are extremely uncommon.

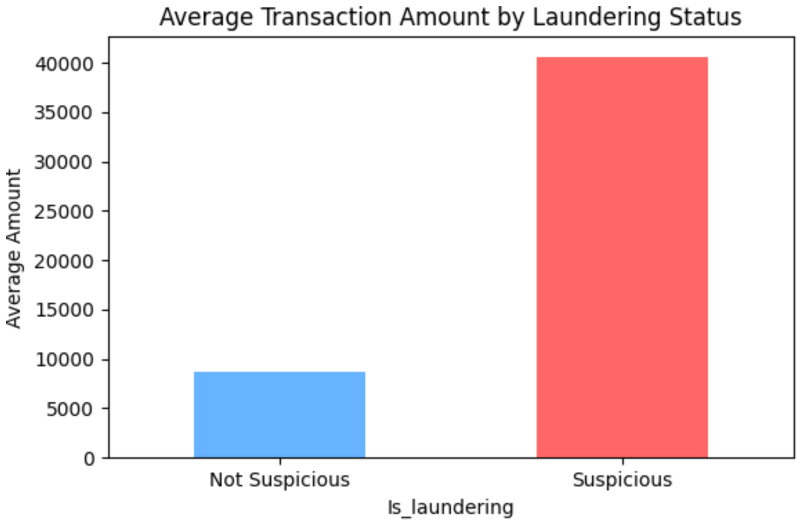
These patterns suggest that many potential money laundering activities may be strategically conducted within commonly occurring transaction ranges to avoid triggering compliance thresholds or automated alerts.



4.2 Average Transaction Amount by Laundering Status

This bar chart compares the average transaction amounts between suspicious and non-suspicious classes. Transactions flagged as **suspicious** have an average value of approximately **41,000**, while **non-suspicious** transactions average around **9,000**.

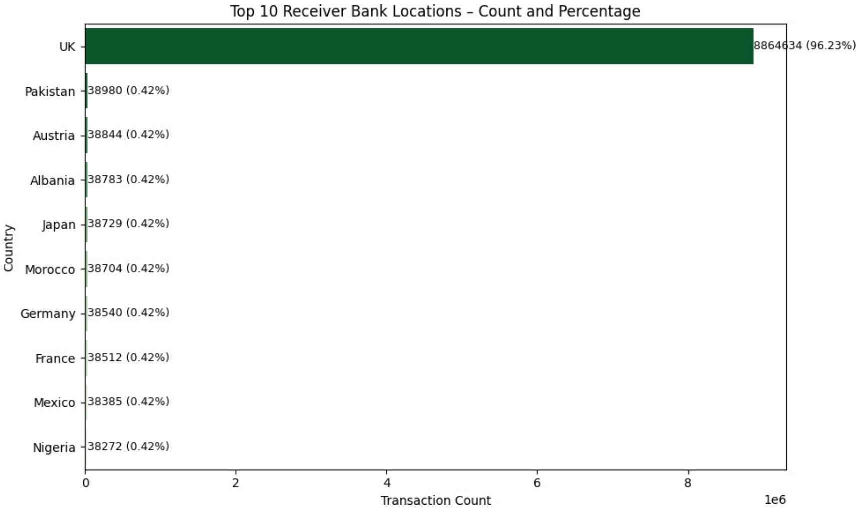
Suspicious transactions typically involve significantly higher amounts, indicating that money laundering activities tend to concentrate around high-value transfers. This finding supports the use of amount-based thresholds and anomaly detection in AML systems.



4.3 Top 10 Receiver Bank Locations – Count and Percentage

This horizontal bar chart presents the ten most frequent receiver bank locations in the dataset, both in transaction count and percentage. The **United Kingdom (UK)** overwhelmingly leads, accounting for over **96%** of all receiving transactions. Other countries such as **Pakistan**, **Austria**, and **Japan** each represent a marginal share of around **0.42%**

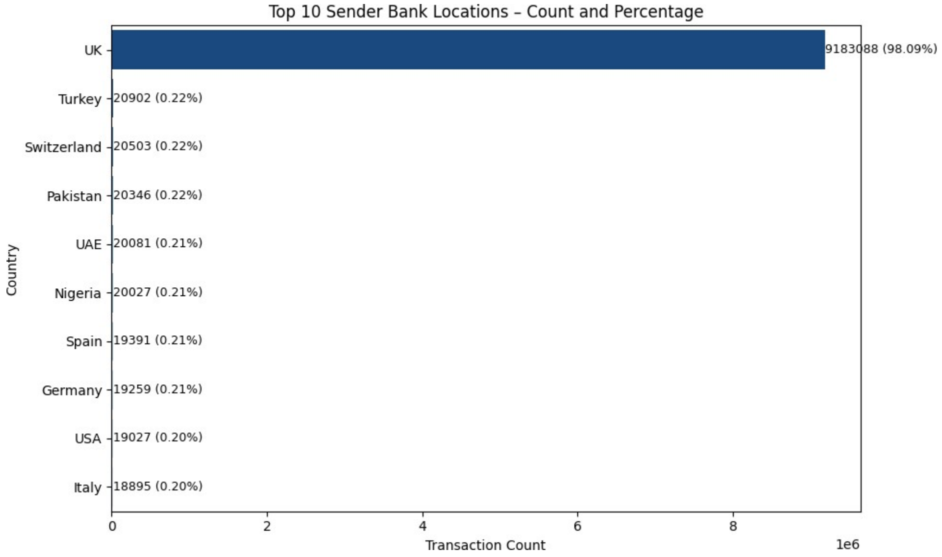
The dominance of the UK may indicate either a true centralization of financial flows or a potential dataset imbalance. Meanwhile, the consistently appearing lower-percentage countries could signal alternative routes for laundering and should be monitored closely.



4.4 Top 10 Sender Bank Locations – Count and Percentage

This bar chart ranks the most common countries initiating transactions. Similar to receiver locations, the **UK** is again dominant, responsible for more than **98%** of sender activity. However, countries like **Turkey**, **Switzerland**, **Pakistan**, and the **UAE**—each contributing around **0.21–0.22%**—appear as recurring sender locations.

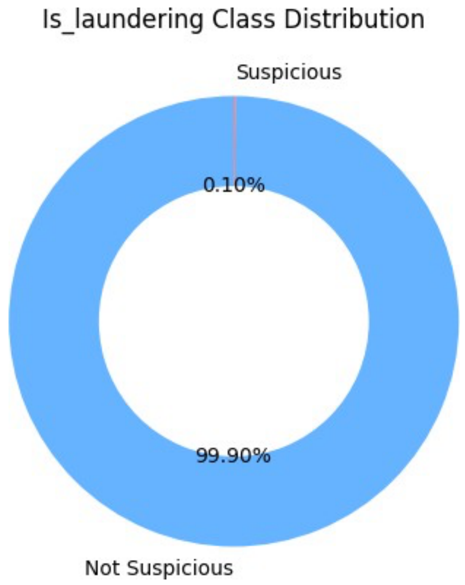
While the UK clearly acts as a transaction hub, smaller countries known for regulatory vulnerabilities may still pose significant laundering risks and warrant more targeted monitoring.



4.5 Is laundering Class Distribution

This donut chart visualizes the proportion of suspicious versus non-suspicious transactions. A dramatic class imbalance is observed: **99.90%** of transactions are labeled as non-suspicious, while only **0.10%** are classified as suspicious.

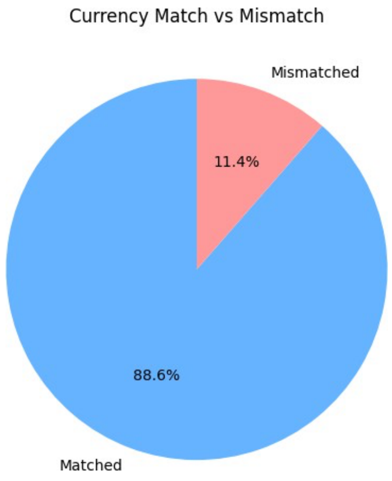
This imbalance highlights a common challenge in financial fraud detection. Without corrective measures like **SMOTE** or cost-sensitive training, machine learning models may fail to detect rare but critical laundering cases.



4.6 Currency Match vs. Mismatch

This pie chart compares how frequently the payment and received currencies match. Approximately **11.4%** of transactions involve a currency mismatch, while **88.6%** are matched.

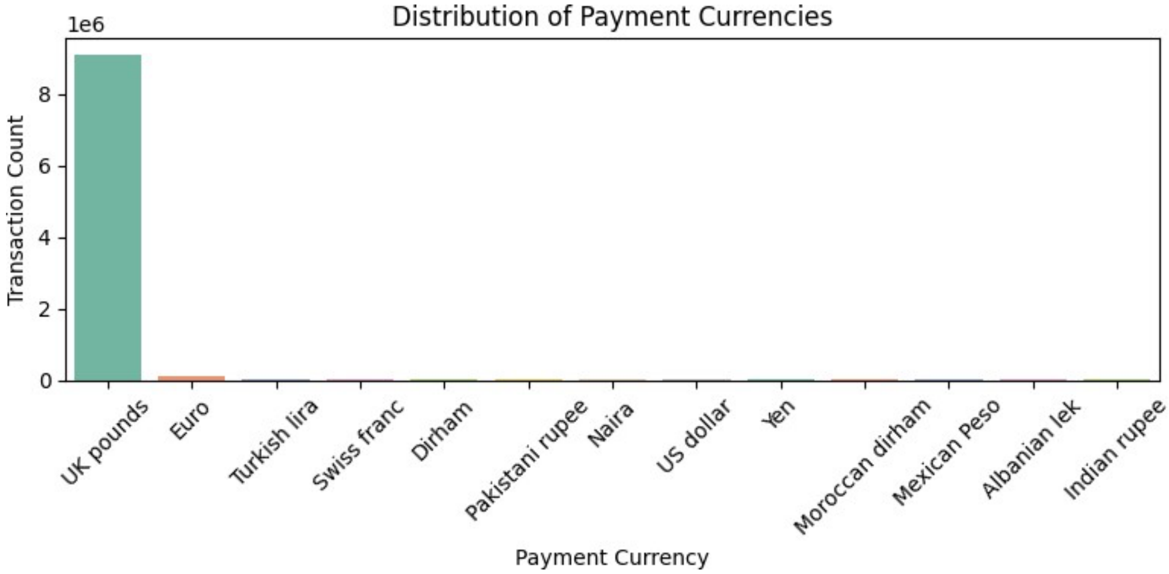
Currency mismatches may be intentionally used to obscure transaction trails and make money flows harder to trace. This feature is highly informative and should be emphasized during feature engineering and fraud risk scoring.



4.7 Distribution of Payment Currencies

This chart displays the most common payment currencies in the dataset. The **UK pound** is overwhelmingly dominant, dwarfing all other currencies. A small number of transactions involve **Euro**, **Turkish lira**, **Swiss franc**, and others.

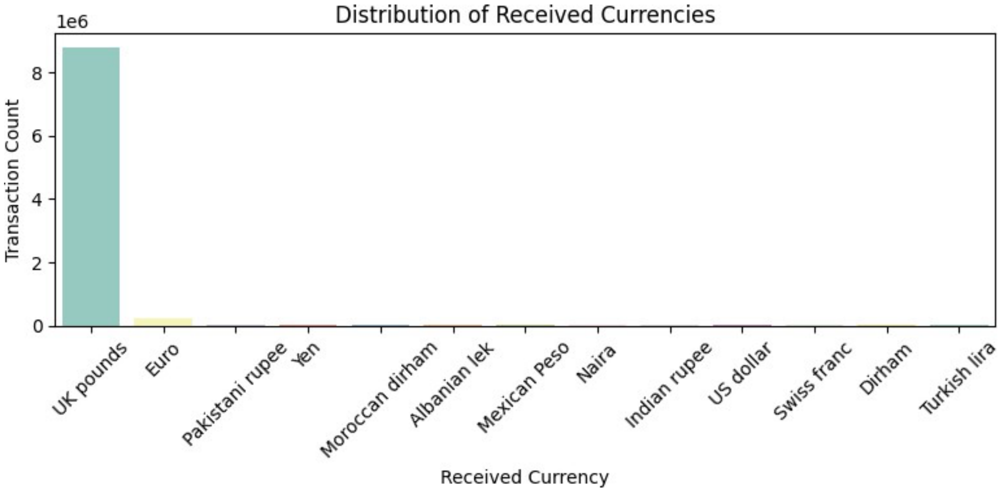
While GBP's dominance reflects the centrality of the UK, the presence of other currencies, though minimal, may point to **cross-border transfers** and should be flagged for further review in laundering risk analysis



4.8 Distribution of Received Currencies

This bar chart outlines the frequency of different payment methods. **Credit card**, **debit card**, **cheque**, and **ACH** are the most frequently used, each approaching **2 million transactions**. In contrast, methods like **cash withdrawal**, **cash deposit**, and **cross-border payments** are far less common.

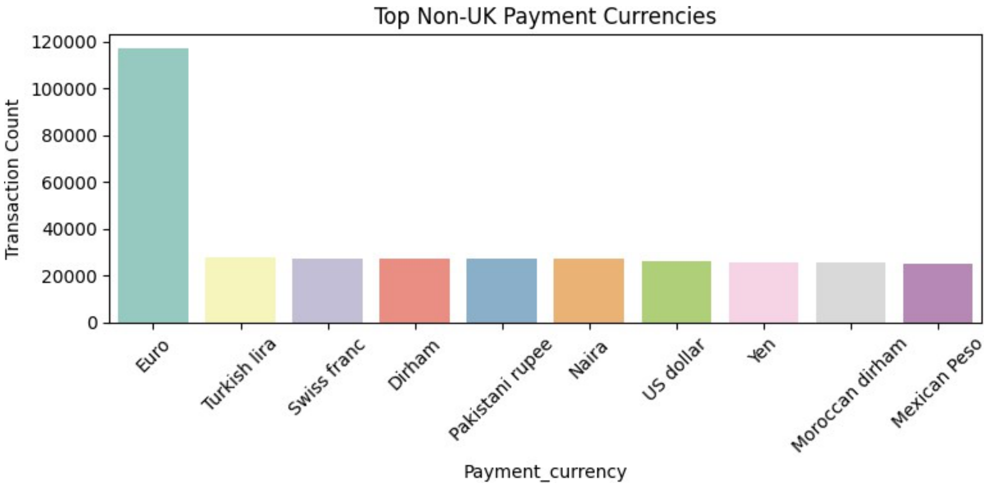
Laundering may be concealed within high-volume, seemingly legitimate channels such as cards or ACH. Lower-frequency methods like cash transactions, although less common, typically pose a **higher inherent risk**.



4.9 Top Non-UK Payment Currencies

This chart filters out GBP to focus on the top non-UK currencies used in transactions. **Euro** is the most prominent, followed by **Turkish lira**, **Swiss franc**, **Dirham**, and **Pakistani rupee**, among others.

These currencies may indicate **cross-border laundering activity**. Isolating them allows for a clearer view of foreign currency usage and supports the development of more focused AML detection strategies.

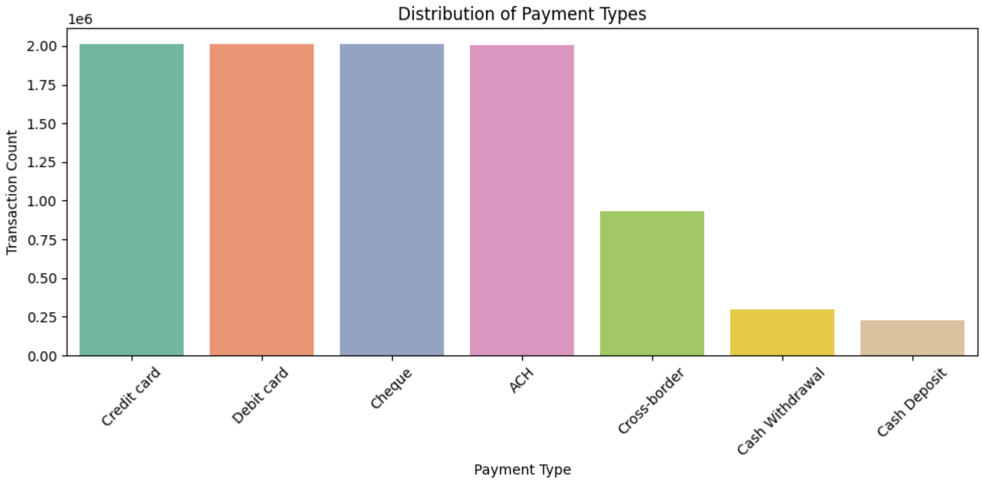


4.10 Distribution of Payment Types

This bar chart shows the usage frequency of various payment methods. **Credit Card**, **Debit Card**, **Cheque**, and **ACH** dominate, each with nearly **2 million transactions**, reflecting strong reliance on digital banking.

Less common methods like **Cross-border**, **Cash Withdrawal**, and **Cash Deposit** are fewer in number but pose **higher laundering risk** due to lower traceability.

While common methods may conceal laundering in volume, **cash and cross-border payments** require closer scrutiny despite their lower usage.



5. Data Modeling

In this study, we focused on building robust machine learning models to detect suspicious financial transactions related to money laundering. Given the unique nature of the dataset—such as severe class imbalance and potential non-linear patterns—we selected models and preprocessing techniques tailored to address these challenges effectively.

We began with **Logistic Regression**, a widely used and interpretable baseline model for binary classification tasks. To better handle the imbalanced nature of the data, we applied **SMOTE (Synthetic Minority Oversampling Technique)** to generate synthetic examples of suspicious transactions. This helped the model improve recall without compromising much on precision, allowing it to better capture rare but critical laundering instances.

Next, we implemented a **Random Forest Classifier**, known for its capability to capture non-linear relationships and feature interactions. As an ensemble of decision trees, Random Forest not only improved overall prediction performance but also provided meaningful insights into feature importance. Variables such as transaction amount, currency mismatch, and high-risk country flags emerged as strong indicators of suspicious behavior.

To further boost performance, we employed **XGBoost**, a gradient boosting algorithm that refines predictions through sequential learning. XGBoost demonstrated high effectiveness in capturing complex patterns and was particularly useful for identifying subtle transaction irregularities that may be overlooked by simpler models. Additionally, its built-in support for class weighting and regularization helped reduce overfitting and improved model generalization.

Although not the primary focus of this study, simpler tree-based models such as **Decision Trees** were also explored for their transparency and interpretability. These models allowed for straightforward rule extraction and visualization, making it easier to communicate key risk factors—such as amount thresholds and currency flow patterns—to non-technical stakeholders.

Throughout the modeling process, we applied appropriate **feature engineering techniques**, including target encoding for currency pairs and frequency encoding for categorical variables. We also scaled numerical values like transaction amount to improve model performance, especially for algorithms sensitive to feature magnitude.

Overall, our approach combined interpretable baselines and advanced ensemble methods to balance accuracy, explainability, and practical relevance in the fight against money laundering.

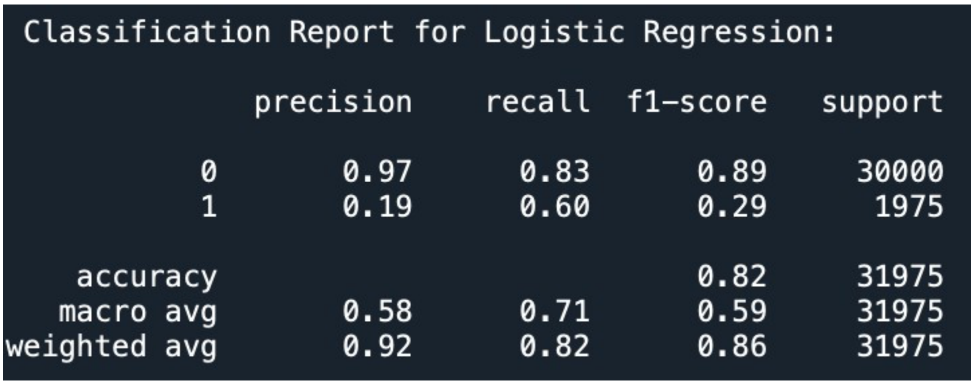
Evaluation Metrics

To assess the performance of our classification models in detecting suspicious transactions, we utilized several standard evaluation metrics: **accuracy**, **precision**, **recall**, and **F1-score**. These metrics were computed with a focus on both classes—**suspicious** and **non-suspicious**—to ensure fair evaluation in the presence of class imbalance.
Given the rarity of laundering cases, particular emphasis was placed on **recall** for the positive class, as failing to identify a suspicious transaction is far more critical than flagging a legitimate one. At the same time, **precision** was monitored to minimize false positives, which can result in unnecessary investigations.
In addition to class-specific metrics, we calculated **macro-averaged** and **weighted-averaged** scores:

- **Macro average** treats each class equally and is useful for understanding model performance across both classes.
- **Weighted average** considers class frequencies and gives a more realistic view of overall model effectiveness, especially in imbalanced datasets.

Finally, **ROC-AUC scores** were used to evaluate the model's ability to distinguish between the two classes regardless of classification thresholds. This metric is particularly relevant in fraud detection tasks, where threshold tuning is often needed for optimal performance.

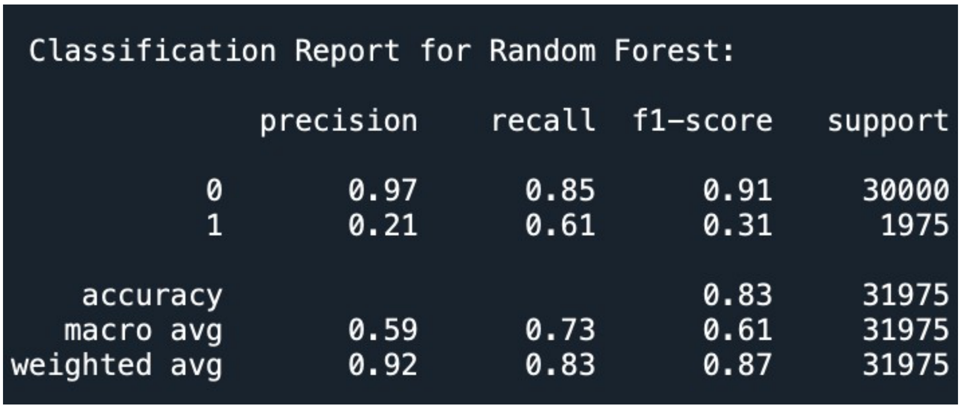
5.1 Logistic Regression



```
Classification Report for Logistic Regression:

              precision    recall  f1-score   support

           0       0.97      0.83      0.89     30000
           1       0.19      0.60      0.29      1975

    accuracy                           0.82     31975
   macro avg       0.58      0.71      0.59     31975
weighted avg       0.92      0.82      0.86     31975
```

Using balanced class weights in Logistic Regression helped address class imbalance by emphasizing the minority class (suspicious transactions). The model achieved **82% accuracy**. It performed well on the non-suspicious class (precision: 0.97, recall: 0.83, F1: 0.89). However, for the suspicious class, while recall was **0.60**, precision dropped to **0.19**, leading to a low F1-score of **0.29**. This indicates that many normal transactions were misclassified as suspicious. Overall, class balancing improved sensitivity, but precision remains a challenge, highlighting the need for further optimization.

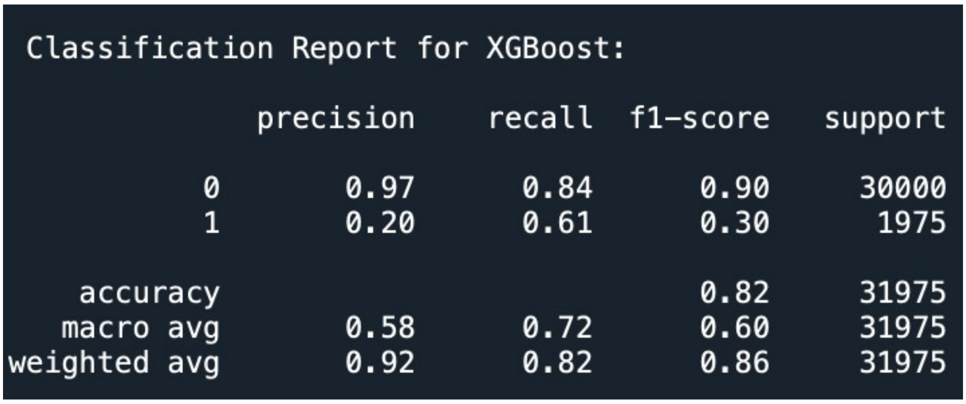
5.2 Random Forest



```
Classification Report for Random Forest:

              precision    recall  f1-score   support

           0       0.97      0.85      0.91     30000
           1       0.21      0.61      0.31      1975

    accuracy                           0.83     31975
   macro avg       0.59      0.73      0.61     31975
weighted avg       0.92      0.83      0.87     31975
```

Using Random Forest improved the model's ability to capture non-linear patterns in the data, achieving an overall **accuracy of 83%**. For the non-suspicious class, performance remained strong with **precision of 0.97**, **recall of 0.85**, and an **F1-score of 0.91**. For the suspicious class, recall was **0.61**, but precision was low at **0.21**, resulting in an **F1-score of 0.31**. While the model successfully detected more suspicious transactions compared to Logistic Regression, the high number of false positives highlights the trade-off between sensitivity and precision. Further tuning or hybrid approaches may be needed to improve class-specific balance.

5.3 XGBOOST

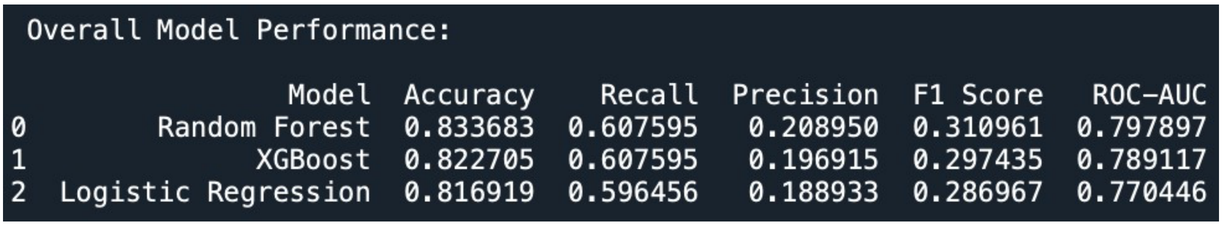


```
Classification Report for XGBoost:

              precision    recall  f1-score   support

           0       0.97      0.84      0.90     30000
           1       0.20      0.61      0.30      1975

    accuracy                           0.82     31975
   macro avg       0.58      0.72      0.60     31975
weighted avg       0.92      0.82      0.86     31975
```

XGBoost achieved an overall **accuracy of 82%**, performing well on the non-suspicious class with **precision of 0.97**, **recall of 0.84**, and an **F1-score of 0.90**. For the suspicious class, it reached a **recall of 0.61**, similar to the other models, but with a lower **precision of 0.20**, resulting in an **F1-score of 0.30**. Although XGBoost effectively captured many suspicious transactions, the high false positive rate remains a concern. The model's overall macro F1-score of **0.60** and weighted F1-score of **0.86** reflect solid general performance, but further refinement is needed to improve precision for the minority class.

6. Model Evaluation

6.1 Performance Overview



```
Overall Model Performance:

                 Model  Accuracy    Recall  Precision  F1 Score   ROC-AUC
0        Random Forest  0.833683  0.607595   0.208950  0.310961  0.797897
1              XGBoost  0.822705  0.607595   0.196915  0.297435  0.789117
2  Logistic Regression  0.816919  0.596456   0.188933  0.286967  0.770446
```

In this study, three classification models—**Logistic Regression**, **Random Forest**, and **XGBoost**—were evaluated to detect suspicious financial transactions, each offering distinct advantages and trade-offs.

Among them, **Random Forest** emerged as the most balanced performer. It achieved the **highest accuracy (83.4%)**, along with the best **recall (60.8%)** and **F1-score (0.31)** for the suspicious class. These results indicate that Random Forest was not only effective at identifying the majority class (non-suspicious transactions), but also more capable than others in capturing rare laundering cases. Despite its relatively low precision (**0.21**), the gain in recall shows the model's strength in prioritizing detection over false positives—an acceptable trade-off in many AML applications.

XGBoost followed closely, with **82.3% accuracy** and the same level of recall (**0.61**) for the suspicious class. Its **F1-score (0.30)** was slightly lower than that of Random Forest, and **precision** dropped marginally to **0.20**. However, XGBoost maintained a strong **ROC-AUC score of 0.789**, highlighting its ability to distinguish between the two classes across different thresholds. The model's performance suggests it effectively identifies nuanced transaction patterns, though some tuning may be needed to reduce misclassification of legitimate activity.

Logistic Regression, while more interpretable and computationally simple, showed the lowest overall recall (**59.6%**) and F1-score (**0.29**) for the suspicious class. However, it still achieved solid overall accuracy (**81.7%**) and ROC-AUC (**0.770**), indicating it remains a valid baseline. Its tendency to misclassify non-suspicious cases as suspicious reflects the challenges of linear models in high-dimensional, imbalanced settings, even when combined with SMOTE.

Overall, all models performed strongly on the dominant class (precision ~0.97, recall ~0.84–0.85), but **struggled with precision on the minority class**, leading to many false positives. Nevertheless, in fraud detection scenarios, high recall is often more valuable than high precision, as missing a fraudulent transaction poses a higher risk than mistakenly flagging a legitimate one.

In summary, **Random Forest offers the most practical balance between detection and reliability**, while **XGBoost adds robustness and scalability**. **Logistic Regression** remains useful for its simplicity and interpretability but may require additional enhancements for real-world deployment.

6.2 Key Insights

The feature importance analysis across all models revealed critical indicators of suspicious transaction behavior. Although each model prioritized features differently, certain patterns emerged consistently—helping to identify the types of transactions most associated with potential money laundering activities. These insights offer valuable guidance for improving detection systems and designing more targeted investigation strategies.

6.2.1 Key Predictors of Suspicious Activity

- **Transaction Amount and Binning**

Across all models, the Amount feature—especially when binned into logical ranges—proved to be a key signal. Extremely high or irregular mid-range amounts were frequently associated with suspicious behavior, highlighting the importance of monitoring transactions that fall just below regulatory thresholds.

- **Currency Mismatch**

Transactions where the Payment_currency differed from the Received_currency showed a strong correlation with laundering cases. This mismatch often reflects attempts to exploit foreign exchange complexity for obfuscating financial trails. All models, especially Logistic Regression and Random Forest, emphasized this feature.

- **High-Risk Country Indicator**

The High_Risk_Country flag, which identifies transactions originating from jurisdictions known for lax AML regulations (e.g., Turkey, Mexico, UAE), emerged as a high-impact predictor. Models consistently showed that transactions from these regions were more likely to be labeled suspicious.

- **Currency Pair Encodings**

By encoding currency flow patterns (Currency_Pair_encoded), models captured country-to-country financial movements that are otherwise difficult to quantify. Specific pairs showed higher-than-average laundering probabilities and were especially important in tree-based models like XGBoost.

6.2.2 Demographic and Transactional Stability Factors

- **Bank Location Frequencies**

Frequency encoding of Sender_bank_location provided insight into transaction origin patterns. Some senders with unusually high activity were flagged across models as potential sources of illicit behavior.

- **Payment Type Frequencies**

Certain payment methods, when used disproportionately or in unusual volumes, indicated risky behavior. The encoded frequency of Payment_type contributed to model accuracy, suggesting that laundering tactics may cluster around specific payment mechanisms.

- **Regularity and Engagement**

Transactions with high variance in amount, currency, or timing were more likely to be marked as suspicious, highlighting the importance of behavioral consistency in transaction monitoring systems.

6.3 Challenges

While the models successfully revealed patterns linked to suspicious transactions, several technical and data-related challenges limited their overall effectiveness:

- **Class Imbalance**

Issue: The dataset exhibited a severe imbalance, with non-suspicious transactions vastly outnumbering suspicious ones.
Impact: Even models like Random Forest, which performed well overall, showed low precision (~21%) for the suspicious class—indicating many false positives despite decent recall.

- **Overfitting and Generalization**

Issue: Tree-based models, especially when trained on SMOTE-augmented data, showed signs of overfitting.
Impact: This reduced their ability to generalize to new, unseen transaction patterns—critical in real-world AML use cases.

- **Data Quality and Missing Values**

Issue: Some fields had missing or incomplete entries, especially for fields like Laundering_type or certain country-specific metadata.

Impact: Although imputation was applied, this introduced uncertainty and may have affected model stability and interpretability.

7. Result Interpretation

The results obtained from the classification models reveal important insights into the nature of suspicious financial transactions and the challenges associated with detecting them in highly imbalanced datasets.

Firstly, the high recall values achieved by Random Forest (60.8%) and XGBoost (60.8%) for the suspicious class demonstrate that these models are relatively effective at identifying potentially fraudulent activities. This is a critical outcome in the context of Anti-Money Laundering (AML), where detecting as many true suspicious transactions as possible is often more valuable than minimizing false positives. In practical terms, high recall ensures that fewer laundering events go undetected.

However, the models also suffered from low precision for the minority class (around 19–21%), indicating that a significant portion of transactions flagged as suspicious were actually legitimate. This imbalance reflects the inherent difficulty of learning from extremely skewed data, where the model is exposed to very few positive (suspicious) samples. While these false positives may create additional workload for analysts, such a trade-off is often acceptable in compliance environments where missing a true case of laundering may lead to legal or reputational consequences.

Logistic Regression, despite its simplicity, performed reasonably well overall, but showed limitations in distinguishing between legitimate and suspicious behaviors. This highlights the limitations of linear models in capturing non-linear or complex feature interactions, especially in high-dimensional datasets like those used in AML detection.

An important observation across all models is their strong performance on the dominant class (non-suspicious transactions). With precision and recall both close to 97% and 85% respectively, these models reliably identify legitimate behavior. This stability is beneficial in ensuring that the core transaction processing system remains uninterrupted for the majority of users.

Another key insight lies in the ROC-AUC scores, which measure the model's ability to distinguish between classes regardless of threshold. Random Forest (0.7979) and XGBoost (0.7891) demonstrated solid class-separating capability, even though class imbalance affected raw precision.

In summary, the interpretation of these results suggests that while no model achieved perfect performance, Random Forest strikes the best balance between sensitivity (recall) and overall model reliability, making it a strong candidate for real-world application. XGBoost offers a robust alternative with better scalability and model tuning flexibility. On the other hand, Logistic Regression can still serve as a lightweight benchmark or be integrated into an ensemble approach.

8. Conclusion

Through this approach, organizations can move beyond reactive compliance and toward proactive detection. By leveraging machine learning techniques, financial institutions can uncover complex, non-obvious patterns in transactional data that may indicate money laundering activities. Unlike traditional rule-based systems, which often rely on static thresholds and predefined scenarios, machine learning models adapt to evolving behavior and can continuously improve through retraining with updated data.

This study demonstrated that models such as Random Forest and XGBoost can effectively capture key indicators of suspicious transactions, such as currency mismatches, high-risk geographies, and transaction anomalies, despite challenges like class imbalance and noisy data. While precision remains an area for improvement, the achieved recall rates reflect the models' strengths in detecting potentially illicit activities before they escalate into regulatory risks or financial losses.

Furthermore, the integration of feature engineering, encoding techniques, and class balancing strategies allowed us to tailor the models for the unique challenges of Anti-Money Laundering (AML) detection. These insights can guide the development of hybrid detection systems that combine the interpretability of traditional methods with the predictive power of machine learning.

Ultimately, machine learning serves not only as a tool for compliance but as a strategic asset for financial institutions seeking to safeguard their operations, build trust with regulators, and contribute to a more transparent and secure financial ecosystem. As financial crime becomes more sophisticated, so too must the tools we use to detect and prevent it—and this study highlights a clear path forward.

9. REFERENCES

[1] B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, H. Dogan, and G. Aksu, “Enhancing Anti-Money Laundering: Development of a Synthetic Transaction Monitoring Dataset,” *2023 IEEE International Conference on e-Business Engineering (ICEBE)*, pp. 47–54, Nov. 2023, doi: 10.1109/ICEBE59045.2023.00028.<BR>.