INDONESIAN JOURNAL OF INNOVATION AND APPLIED SCIENCES (IJIAS)
Journal Homepage: https://ojs.literacyinstitute.org/index.php/ijias
E-ISSN: 2775-4162

Volume 6	Issue 1	February (2026)	DOI: 10.47540/ijias.v6i1.2306	Page: 35 – 59

Developing a graph-based machine learning model for identifying money laundering networks associated with sanctioned entities in a bank in Zimbabwe

Belinda Ndlovu[1], Fungai Jacqueline Kiwa[2], Martin Muduva[3], Colletor T. Chipfumbu[3], Sheltar Marambi[3], Amazing Maphosa[1]

[1]National University of Science and Technology, Zimbabwe
[2]Chinhoyi University of Technology, Zimbabwe
[3]Midlands State University, Zimbabwe

Corresponding Author: Belinda Ndlovu; Email: belinda.ndlovu@nust.ac.zw

ARTICLE INFO

Keywords: Anti-Money Laundering, Financial Crime Detection, Graph Convolutional Network, Machine Learning, Transaction Networks.

Received : 04 October 2025
Revised : 24 January 2026
Accepted : 14 February 2026

ABSTRACT

Money laundering networks associated with sanctioned entities pose a significant risk to financial systems, often operating through complex relational transaction structures that evade traditional rule-based monitoring. While graph neural networks have demonstrated promise in financial crime detection, limited work has formally modelled sanction-linked transaction networks within highly imbalanced banking datasets under consistent comparative evaluation. This study proposes a directed weighted graph-based learning framework for identifying sanction-associated money laundering networks using real-world banking transaction data. Transactions were modelled as relational graphs, with accounts as nodes and transfers as weighted edges, and evaluated using a Graph Convolutional Network (GCN) against classical and ensemble classifiers. The proposed model achieved an accuracy of 88.18%, F1-score of 0.7345, ROC-AUC of 0.8968, and a superior Matthews Correlation Coefficient compared to baseline methods. Results demonstrate that relational graph modelling improves the detection of structurally coordinated laundering behaviours that are not captured by independent transaction classifiers. These findings support the integration of graph neural network architectures into anti-money laundering systems to enhance sanction-linked detection capabilities in complex financial networks.

INTRODUCTION

Global financial systems are severely endangered by money laundering, which is passing off funds gained illegally as legitimate income. Criminals incorporate illicit revenues into the official economy by directing them through intricate transactions (Caglayan & Bahtiyar, 2022). Zhang et al. (2025) claim that money laundering facilitates tax evasion, corruption, drug trafficking, and the funding of terrorism. Illicit funds are deposited, transferred through mule accounts, and integrated into financial networks undetected as part of the laundering cycle, which usually entails placement, layering, and integration (Sun et al., 2022). Robust anti-money laundering (AML) systems are essential for maintaining economic integrity and financial stability (Mnkandla et al., 2024).

AML is a global priority for governments and international organizations (Wójcik, 2024). According to guidelines set by the Financial Action Task Force (FATF), financial institutions must have strong procedures in place to identify and report suspicious activities (FATF, 2021). Traditional rule-based approaches have drawbacks, including a lack of adaptability, high resource consumption, and false positive rates that frequently surpass 90% (Fan et al., 2025). Digitalization, automated analytics, anomaly detection, and real-time monitoring have become essential elements of modern AML initiatives to improve compliance.

Several banks in Zimbabwe have implemented sophisticated AML practices, such as digital Know-Your-Customer (KYC) systems, Customer Due Diligence (CDD), real-time verification through mobile platforms, cybersecurity measures, and transaction logging (Kunci et al., 2024; Revesai et al., 2023). The Financial Intelligence Unit (FIU) is notified of any suspicious money laundering activity (Chitimira et al., 2024). Despite all these efforts, structural problems persist, and governance flaws, corruption, and economic instability facilitate illicit financial flows (IFFs). AML threatens the development of governments and the integrity of their institutions; research shows that the issue is made worse by gold smuggling, a lack of transparency, and permissive rules in the extractive industries (Gaviyau & Sibindi, 2023).

Money laundering has increased due to global financial interconnectedness. Despite stringent AML regulations, Switzerland reported significant rates of illicit transactions in 2022, accounting for 74.7% of all European transactions (Alenova et al., 2024). Mexico accounts for 5.4% of Gross Domestic Product (GDP) in illicit outflows, a major hotspot in the Americas (Alawadhi, 2024). Asia also reports substantial losses, with China losing nearly $1 trillion over ten years, and Malaysia's state fund losing $4.5 billion through Chinese banks (Alarfaj & Shahzadi, 2024). Africa's corruption and political instability amplify laundering risks, as demonstrated by Nigeria's $250 million Ibori case and South Africa's Gupta scandal (Japinye, 2024). In 2020, Zimbabwe lost $32,179 billion over two decades, $1.5 billion from gold smuggling, and an estimated $570.75 million a year between 2009 and 2013 (AFRODAD, 2022; Crisis Report, 2021). These numbers highlight how urgently improved governance and cutting-edge detection tools are needed.

According to Wang et al. (2025), rule-based AML systems struggle with processing large volumes of financial data quickly. Machine learning (ML) offers scalable and flexible solutions. Models such as logistic regression, k-Nearest Neighbours, decision trees, random forests, support vector machines, and multi-layer perceptrons are commonly used (Renganathan et al., 2024). In graph-based techniques, accounts serve as nodes and interactions as weighted edges. Graph Neural Networks (GNNs) recently successfully modeled financial transactions as networks (Alenova et al., 2024). These approaches help reduce false positives, detect hidden laundering patterns, improve anomaly detection, and achieve 77–79% predictive accuracy.

Graph-based techniques enable multi-layered transaction tracking, collecting relational and structural aspects beyond analyzing a single transaction. The detection of hidden fraud rings and money laundering schemes across numerous entities is accomplished by directed graphs, embeddings, and GCNs (Alenova et al., 2024). In order to uncover common laundering characteristics, graph theory analyzes vertices and edges in complete, bipartite, and weighted networks (Alarfaj & Shahzadi, 2024). By using structural locations, node centrality, and connectedness to forecast suspicious behaviors, graph analytics improves detection in AML (Effendi & Chattopadhyay, 2025).

Network studies also highlight enforcement tactics, vulnerabilities, and core-periphery architecture, which strengthen regulatory responses (Khan & Akcora, 2022). Graph databases, which dynamically map things and relationships, have become essential to AML (Alarab & Prakoonwit, 2023). According to (Pocher et al., 2023). Nodes stand in for accounts, merchants, or persons, while edges record transaction details including amount, date, and type. This model facilitates real-time monitoring, anomaly identification, and the visualization of laundering flows. Cypher and other graph query languages enable researchers to examine intricate networks and reveal hidden patterns (Alarab & Prakoonwit, 2024). In comparison to classic subgraph detection, advanced techniques such as semi-supervised methods, embeddings, and multipartite graph algorithms (e.g., FlowScope) uncover high-risk account clusters (Blanuša et al., 2024).

To find suspicious behaviors, graph-based anomaly detection uses network structures, including cliques and anomalous subgraphs. While responding to changing fraudulent patterns, iterative diversification, embeddings, and graph representation learning maintain network integrity (Fard, 2023). Fraud prediction, anomaly detection, and classification accuracy are improved by both supervised and unsupervised ML models, such as decision trees, support vector machines, deep neural

networks, and hybrids (Song et al., 2024; Mayeni et al., 2024).

Preprocessing, feature selection, and interpretability are essential to maintaining AML systems' transparency (Dumitrescu et al., 2022). Zimbabwe also faces significant problems in addressing money laundering. In one year, FBC bank produced 38,425 alerts; nevertheless, after manual assessment, only 2,775 of those alerts were connected to sanctioned businesses (FBC Holdings Limited, 2024). This highlights the shortcomings of the present AML regulations, exposing institutions to possible fines and compliance concerns. Through the integration of network analysis, embeddings, and hybrid algorithms, graph-based ML presents a paradigm change. It makes it possible to discover money laundering networks in a scalable, interpretable, and efficient manner, especially in situations when sanctions are involved. By creating and assessing a graph-based ML framework to identify intricate money laundering schemes in Zimbabwe's banking industry, this study seeks to improve AML capabilities.

Despite the growing application of graph neural networks in financial crime detection, several challenges remain unresolved: (i) limited integration of sanction-specific relational modelling, (ii) inadequate handling of highly imbalanced financial transaction graphs, and (iii) insufficient comparative benchmarking against ensemble baselines under consistent evaluation protocols.

This study addresses these gaps by proposing a graph-based learning framework for sanction-linked money laundering detection in a Zimbabwean banking context. The contribution is threefold: Formal modelling of banking transactions as a directed weighted graph; Integration of Graph Convolutional Networks with structured feature selection and imbalance mitigation; Comprehensive comparative evaluation against classical and ensemble classifiers.

METHODS

Research design

This research implemented a design approach with classification and clustering models built on the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. It is an iterative framework with six phases for data mining projects (Plotnikova et al., 2023).

Figure 1 presents the six stages implemented in the CRISP-DM process.

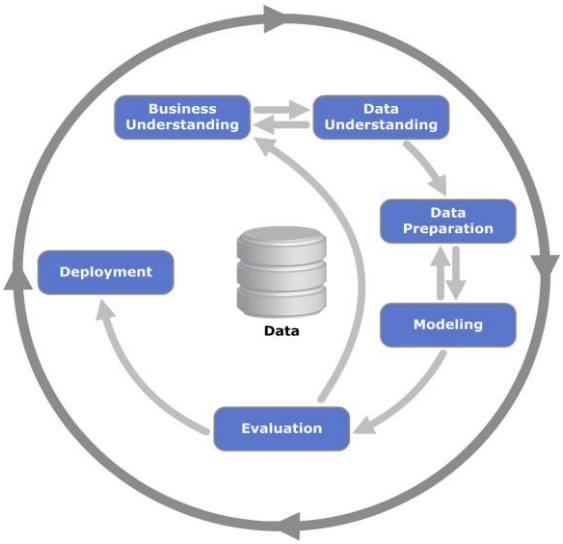

Figure 1. Phases of the CRISP-DM

Business understanding

The primary goal in business understanding was to detect illicit financial flows and hidden relationships among entities involved in money laundering, especially those linked to sanctioned individuals or organisations. This involved recognising suspicious transaction patterns and associations within bank Y transactions indicative of regulatory evasion.

Understanding data

Building an effective and credible graph-based ML model for identifying money laundering networks associated with sanctioned entities in banking transactions requires having a thorough understanding of the data. This study utilised transaction data from 1 January to 31 December 2023, which was stored on a local server with restricted access. The dataset comprises 21,341 rows across 10 columns. Figure 2 shows the features of the dataset.

NO	FEATURE NAME	DESCRIPTION	DATA TYPE
1.	TransactionID	System generated unique id for each transaction	object
2.	Initiation_date	The date transaction was done	datetime
3.	Credit_Account	Account receiving the transaction	float64
4.	Currency	Currency of the transaction	object
5.	Amount	Transaction amount	float
6.	DR_Account	The account where the transaction was initiated	Int64
7.	Country_of_Origin	The country where the transaction is coming from	object
8.	Country_of_Destination	The country where the transaction is going	object
9.	IsSanctioned	An indicator showing whether the transaction involves sanctioned entities or not	object
10	Transaction_Type	Indicated the channel which was used to make the payment	object

Figure 2. Features of the data set

Data preparation

This stage involved selecting pertinent variables and samples for modelling, cleaning the data, generating additional variables, merging different data sources, and altering formats. The dataset in use comprised 21,342 rows and 10 columns (attributes).

Data cleaning

Data cleaning is a crucial step in the ML pipeline, ensuring the accuracy and reliability of the data used to train and evaluate models. Figure 3 shows the code that performs essential preprocessing steps on a DataFrame (df_cleaned) to prepare it for machine learning.

```
In [13]:
# Convert date field
df_cleaned['Initiation_Date'] = pd.to_datetime(df_cleaned['Initiation_Date'])

# Encode target variable
df_cleaned['IsSanctioned'] = df_cleaned['IsSanctioned'].map({'N': 0, 'Y': 1})

# One-hot encode Transaction_Type
df_cleaned = pd.get_dummies(df_cleaned, columns=['Transaction_Type'], drop_first=True)

# Drop irrelevant columns for modeling
df_cleaned = df_cleaned.drop(['TransactionID', 'Credit_Account', 'DR_ACCOUNT',
                              'Country_of_Origin', 'Country_of_Destination', 'Currency'], axis=1)
```

Figure 3. Data cleaning

Handling data imbalance

The dataset showed a great imbalance as it had 92% genuine transactions and only 8% illicit ones. This could easily bias the model in one direction; hence, to fix this, the researcher used the Synthetic Minority Over-sampling Technique (SMOTE) method, which creates synthetic examples of those illicit transactions (Hatam, 2024). In the end, it evened out the classes to a 1:1 ratio (Sharma et al, 2024). Figure 4 shows two bar plots comparing the class distribution of the target variable IsSanctioned before and after applying SMOTE.

```
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter

# Load dataset
df = pd.read_csv("C:/Users/mahla/Desktop/FBC_Analysis/fbc_cleaned.csv")

# Define target column
target_column = 'IsSanctioned'

# Drop non-numeric 'Initiation_Date'
X = df.drop(columns=[target_column, 'Initiation_Date'])
y = df[target_column]

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.3, random_state=42)

# Before SMOTE
print("Before SMOTE:", Counter(y_train))

# Apply SMOTE
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

# After SMOTE
print("After SMOTE:", Counter(y_train_smote))

# Plot class distribution
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.countplot(x=y_train, ax=axes[0], palette="Set2")
axes[0].set_title("Before SMOTE")
sns.countplot(x=y_train_smote, ax=axes[1], palette="Set1")
axes[1].set_title("After SMOTE")
plt.tight_layout()
plt.show()
```

Figure 4. Distribution before SMOTE and after SMOTE

In the left plot, the class distribution is highly imbalanced, with a significantly larger number of instances labelled as 0 (not sanctioned) than 1 (sanctioned), indicating a class imbalance problem that could bias ML models. After applying SMOTE, the right plot displays the distribution, which synthetically generates samples for the minority class to achieve a balanced dataset, resulting in an equal number of observations for both classes.

Feature scaling

It is crucial to apply feature scaling to manage data with significantly different magnitudes effectively. This research utilised the standard scaling method, a technique designed to normalise the independent features of the data within a specified range, as demonstrated in Figure 6. Figure 5 shows the code snippet for standardising data.

```
# Step 4: Select and scale features
feature_cols = ['Amount'] + [col for col in df.columns if col.startswith('Transaction_Type_')]

scaler = StandardScaler()
features = scaler.fit_transform(df[feature_cols])
x = torch.tensor(features, dtype=torch.float)
```

Figure 5. Data standardization

Figure 6 shows standard scaling on the data for feature scaling to handle standardisation for the features.

```
from sklearn.preprocessing import StandardScaler

# 1. Initialize the scaler
scaler = StandardScaler()

# 2. Fit and transform the SMOTE-resampled feature set
X_res_scaled = scaler.fit_transform(X_res)

# Convert back to DataFrame (to keep column names)
X_res_scaled_df = pd.DataFrame(X_res_scaled, columns=X_res.columns)
```

Figure 6. Standard_scaling

The StandardScaler was used to standardise the features.

Feature selection

In this study, the Pearson Correlation Coefficient has been applied to choose the most crucial features (Sharma et al., 2024). It is a statistical number that highlights the dependency between two features and is measured using the following formula:

$$r = [n(\Sigma xy) - \Sigma x \Sigma y]/\text{Square root of}\sqrt{[n(\Sigma x^2) - (\Sigma x)^2][n(\Sigma y^2) - (\Sigma y)^2]}$$

Where n denotes the number of data points, x and y are two variables being compared, and r is the correlation. Stronger positive and negative correlations are indicated by values nearing +1 and -1, respectively. During the experimentation, analysis of the correlation heatmap for four features revealed that several features exhibited high correlation, leading to the removal of six features based on a heuristic correlation threshold of 0. Consequently, the original feature set, which comprised 10 features, has been condensed to four independent features (Chuang & Chen, 2024). The feature selection process continues with evaluating the retained features' ranking through Mutual Information (MI) calculation. The MI between X (feature) and Y (target) lies between 0 and 1, measuring the variables' dependency. Info (X;Y)=V(X)-V(X|Y): Where Info (X; Y) implies MI for X and Y, V(X) is the entropy for X, and V (XjY) depicts conditional entropy for X given Y. Features having a higher MI value are considered as the most discriminating ones.

Figure 7 presents the code snippets for showing the importance of features based on mutual information.

```
[7]: import pandas as pd
from sklearn.feature_selection import mutual_info_classif
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("C:/Users/mahla/Desktop/FBC_Analysis/fbc_cleaned.csv")

# Define target and drop non-numeric columns (like dates)
target_column = 'IsSanctioned'
X = df.drop(columns=[target_column, 'Initiation_Date'])
y = df[target_column]

# If X contains any categorical features, convert them
for col in X.select_dtypes(include='object').columns:
    X[col] = LabelEncoder().fit_transform(X[col])

# Compute Mutual Information
mi_scores = mutual_info_classif(X, y, random_state=42)
mi_scores_series = pd.Series(mi_scores, index=X.columns).sort_values(ascending=False)

# Print scores
print("Mutual Information Scores:\n", mi_scores_series)

# Plot
plt.figure(figsize=(8, 5))
sns.barplot(x=mi_scores_series.values, y=mi_scores_series.index, palette="viridis")
plt.title("Mutual Information between Features and Target")
plt.xlabel("MI Score")
plt.ylabel("Feature")
plt.tight_layout()
plt.show()
```

Figure 7. Feature Importance Based on Mutual Information

The illustration shows the ranking of features based on their (MI) scores concerning the target variable, highlighting their importance in the classification task. Feature importance analysis identified transaction amount, transaction type, frequency indicators, and graph-derived centrality metrics as the most discriminative predictors of sanction-linked activity. Structural metrics such as degree centrality and clustering coefficients demonstrated strong mutual information scores, confirming the importance of relational dependencies in AML detection.

Graph Construction and Representation

The transaction dataset was modelled as a directed weighted graph $G = (V, E, X)$, where nodes V represent unique account identifiers, edges $E \subseteq V \times V$ represent financial transfers, and node feature matrix $X \in \mathbb{R}^{|V| \times d}$ encodes transaction-level attributes. Edge weights correspond to transaction amounts aggregated within a rolling 30-day window. Self-loops were added to preserve node identity during convolution. Adjacency matrices were symmetrically normalised before GCN propagation.

Modelling

In this stage, suitable modelling techniques were selected based on data attributes and entity connections. GNNs were used to handle non-Euclidean data, specifically R-GCN (Zhang et al., 2019). Five classifiers were initially applied for data classification, followed by five ensemble classifiers that utilised bagging and boosting to improve accuracy further. These models were built to compare with the graph-based approach.

The ensemble classifiers used are as follows: CatBoost, XGBoost, Light Gbm, Adaboost, and Random Forest, whilst Decision trees, KNN, logistic Regression, SVM, and Naïve Bayes were used as individual classifiers during the data modelling stage, and their performance was compared.

The Graph Convolutional Network layer is defined as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)})$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\tilde{D}$ its degree matrix, $W^{(l)}$ trainable weights, and σ the ReLU activation.

Model selection technique

Data classification first used five individual classifiers in the experiment, then three ensemble classifiers, Random Forest, XGBoost, and AdaBoost, to improve accuracy.

Model hyperparameter tuning

Top-level parameters essential for configuring the model were chosen, and optimisation was carried out to determine the most effective set of hyperparameter values to enhance the performance of a learning algorithm.

Evaluation

The graph-based ML model for identifying money laundering networks associated with sanctioned entities in the banking transactions dataset was evaluated at this stage. The evaluation was carried out to determine the accuracy value of the model used. The author used K-fold 10-Cross Validation, a model evaluation technique that is quite popular and widely used. The author chose 20 for the K value. This means that for each iteration, the K-fold algorithm evaluates the model using 20% test data and 80% training data (Elegbede & Sibanda, 2021). Table 1 presents the evaluation metrics used and their characteristics.

Table 1. Model evaluation metrics

Metric	Definition	Author
Confusion matrix	A 2x2 matrix is used to determine the link between the model's expected and actual values.	(Khalid & Abdalla, 2024)
Precision	It is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class.	(Xavier et al., 2022)
Recall	It permits measurement of the fraction of positive patterns that are correctly classified.	(Elvas et al., 2023)
Accuracy	It measures the ratio of correct predictions to the total number of instances evaluated.	(Perera & Premaratne, 2024)
F1	The metric that represents the harmonic mean between recall and precision values	(Sánchez et al., 2023)
Matthew's correlation coefficient (MCC)	Measures the correlation between what you observe in binary classifications and what the predictions come up with. The values range from minus one, which means total disagreement, to plus one for perfect prediction. Zero points to random performance. Between prediction and observation.	(Wade & Hofmann, 2024)
ROC curve	The ROC curve plots out the link between true positives and false positives in a visual way. You know, the area under that curve, or AUC, gauges how the model performs overall. The values sitting above the diagonal line show discrimination that's better than random chance.	(Perera & Premaratne, 2024)

Deployment

The analysis indicated that the GCN was the most appropriate classifier for the graph-based ML model aimed at detecting money laundering networks linked to sanctioned entities within the banking transactions dataset, achieving an accuracy rate of 88.18%.

Sampling methods and sample size

The study drew on stratified, purposive, snowball, and random sampling methods. It put together a balanced dataset with normal transactions and suspicious ones for training the model.(Sun et al., 2022). The final sample had 21,342 rows across 10 columns.

Research instruments

The study employed Python and Pandas for data manipulation, while NetworkX and PyTorch Geometric for graph analysis. Scikit-learn helped out with model evaluation. Graph algorithms were used to detect suspicious transaction patterns.

The Data collection procedure

The study used bank Y(bank name withheld for anonymity) transaction data, which went through cleaning and normalisation steps, as well as cross-referencing with sanctions lists to pick out the relevant entities.

Data analysis

The dataset was preprocessed and modelled as a directed graph, with nodes as entities and edges as transactions. GNNs and classifiers were used for node classification and anomaly detection. Model performance was assessed using accuracy, precision, recall, F1-score, and AUC-ROC (Daniel et al., 2023), while visualisations highlighted suspicious transaction patterns and potential money laundering clusters.

Validity and reliability

Validity was maintained by aligning graph-based model features with established indicators of suspicious behaviour, such as high-frequency transactions, circular flows, and unusual centrality patterns (Mellinger & Hanson, 2020). Reliability was ensured through consistent data preprocessing, standardised graph construction, and reproducible ML pipelines, with model performance evaluated using cross-validation to confirm stability across different data subsets (Junjie & Yingxin, 2022).

Ethical consideration

This study adhered to ethical principles, using publicly available, anonymised data with no access to personally identifiable information. ML and graph-based models were applied solely for academic fraud detection research. The research process, including model development and evaluation, was fully documented for transparency and reproducibility, with no data manipulation. To protect the identity of the bank, the pseudonym bank Y was used in such a way to maintain anonymity.

RESULTS AND DISCUSSION

Overview of the dataset

Figure 8 presents the descriptive statistics for three primary variables in the dataset: Credit_Account, Amount, and DR_ACCOUNT, each comprising 21,341 observations. These statistics provide an overview of the central tendency, dispersion, and range within the dataset.

Statistic	Credit_Account	Amount	DR_ACCOUNT
Count	21,341	21,341	21,341
Mean	5,481,709,000	275,204.86	5,429,367,000
Std	2,587,930,000	129,037.29	1,988,699,000
Min	1,000,598,000	50,039.00	2,000,022,000
25%	3,246,685,000	164,792.00	3,694,854,000
50%	5,493,370,000	274,553.00	5,439,574,000
75%	7,723,811,000	386,503.00	7,144,977,000

Figure 8. Overview of the dataset

The descriptive analysis of 21,341 observations examines Credit_Account, Amount, and DR_ACCOUNT. Amount has a mean of 275,204.86, standard deviation of 129,037.29, IQR from 164,792.00 to 386,503.00, and ranges from 50,039.00 to 499,988.00, showing moderate dispersion and balanced distribution. Credit_Account and DR_ACCOUNT have means of 5.48 billion and 5.43 billion, with standard deviations of 2.59 billion and 1.99 billion, reflecting a heterogeneous account base. This variability supports using graph-based machine learning to analyse relational patterns and detect anomalous financial behaviour. Figure 9 shows box plots illustrating the distribution of Amount, Credit_Account, and DR_ACCOUNT.

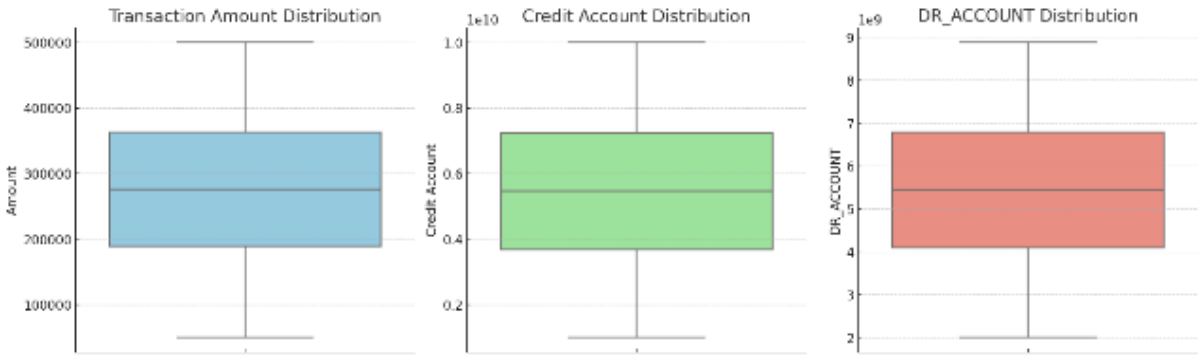

Figure 9. Description of the models

The wide range of all variables highlights the dataset's heterogeneity, supporting graph-based modelling for detecting anomalies or suspicious patterns.

Descriptive analysis of the dataset

This section presents a descriptive analysis of the dataset used in this study to provide a foundational understanding of its structure, content, and key characteristics.

Monthly trend of total transaction amount

Temporal analysis of transaction volumes was conducted by aggregating monthly transaction amounts to identify anomalies, seasonal trends, or irregular spikes, which helps detect unusual financial behaviour and establishes a baseline for typical and potentially suspicious entities. Figure 10 presents a line graph depicting the monthly total transaction amount throughout 2023.

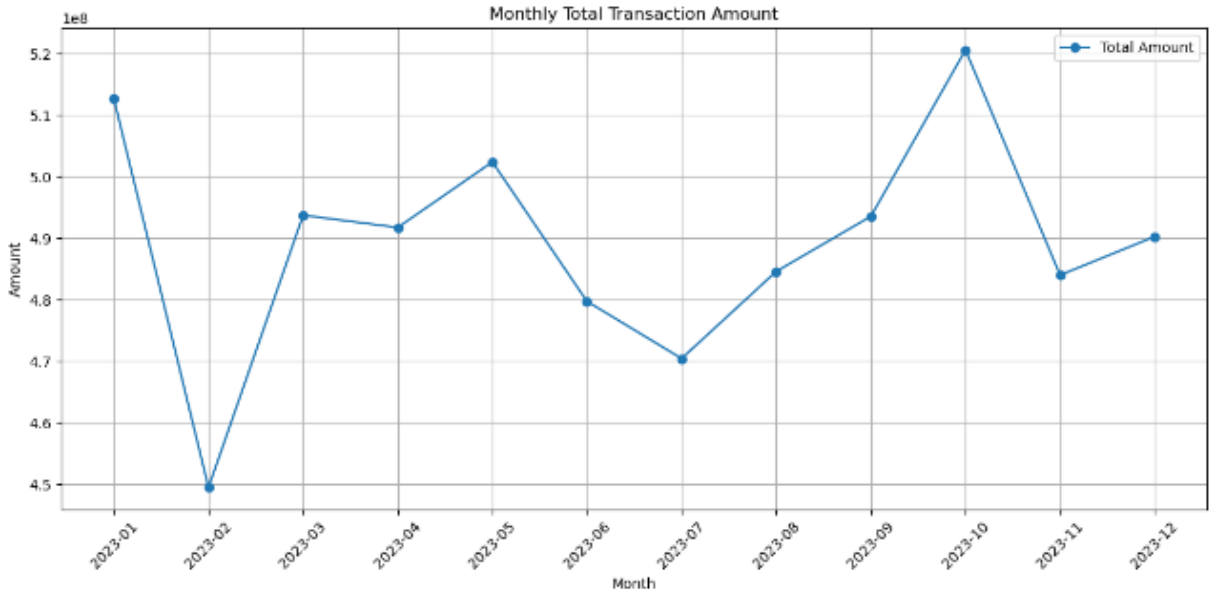

Figure 10. Monthly trend of total transaction amount

The graph starts in January 2023 with a transaction amount of around 4.6 million, which then dips in the following months before peaking at around 5.3 million in May. After another dip, the transaction amount rises again, reaching its highest point in September at over 5.5 million, before declining towards the end of the year.

Average transaction amount by currency

Figure 11 analyses the average transaction amount segmented by currency within the banking transaction dataset.

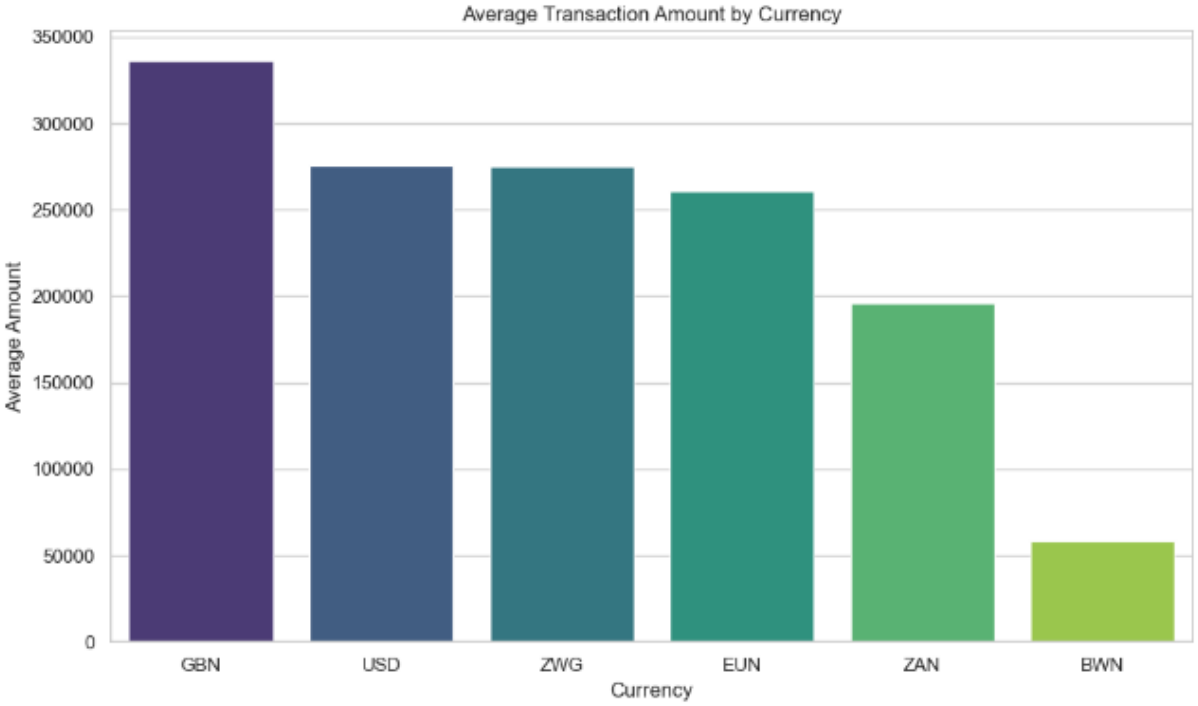

Figure 11. Average transaction amount by currency

The analysis shows that GBN has the highest average transaction amount at 336,399.60, followed by USD, ZWG, and EUN above 260,000, indicating use in high-value transactions. ZAN averages 196,415.06 as a mid-tier currency. BWN is the lowest at 58,697.00, suggesting it is used for smaller transactions, reflecting economic and transactional differences across currencies.

Figure 12 explores the distribution of transaction amounts across different transaction types within the banking transaction dataset, which allows for identifying outliers or suspicious transactions that may indicate money laundering.

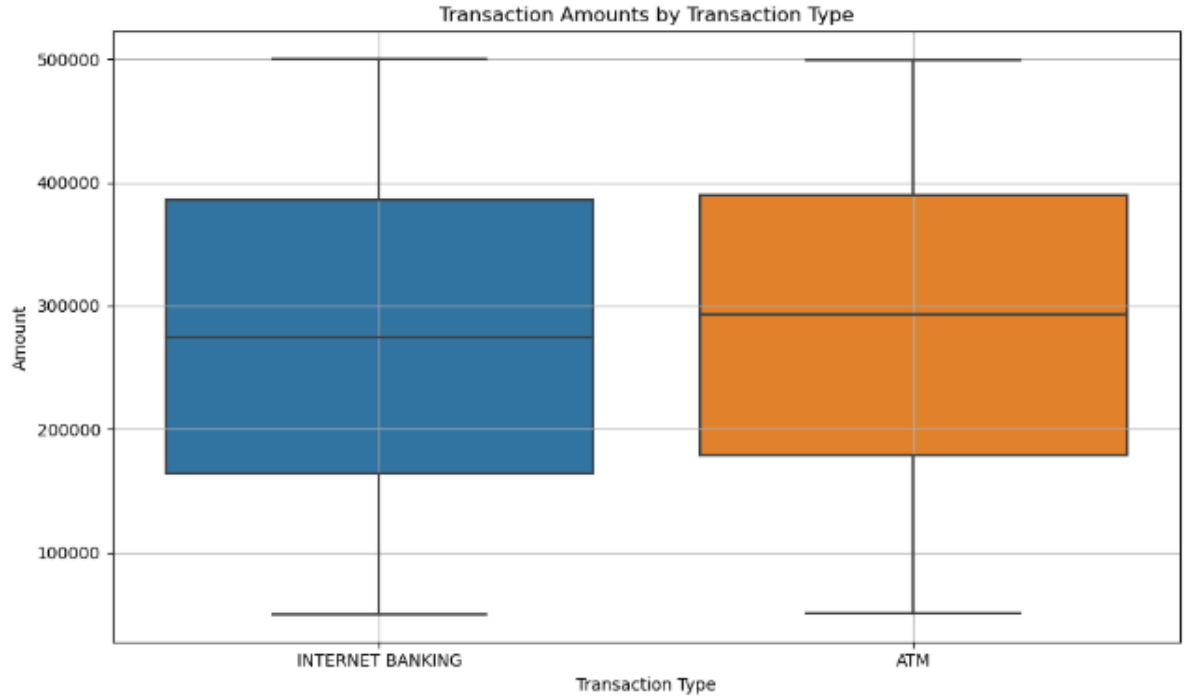

Figure 12. Distribution of transaction amounts for each transaction type

A two-sample t-test was conducted to assess whether there is a statistically significant difference in the average transaction amounts between ATM and Internet Banking transactions. The results yielded a t-statistic of 1.44 and a p-value of 0.15. Since the p-value exceeds the conventional significance threshold of 0.05, we fail to reject the null hypothesis. This indicates no statistically significant difference in the dataset's average transaction amounts between the ATM and Internet Banking channels.

Outlier transaction amounts compared to the average transaction amounts

Figures 13 and 14 focus on identifying and analysing outlier transaction amounts compared to the average transaction values within the bank Y dataset.

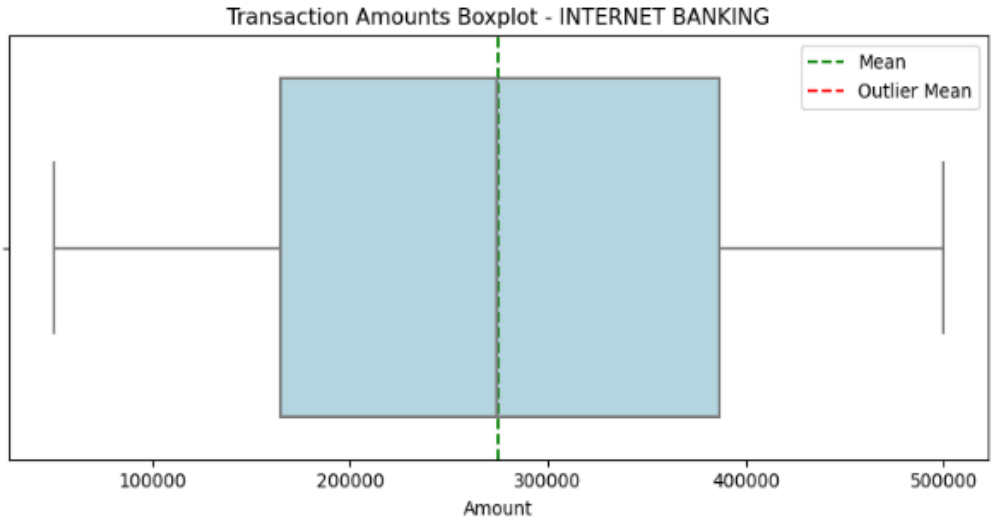

Figure 13. Outlier transaction amounts compared to the average transaction amounts.

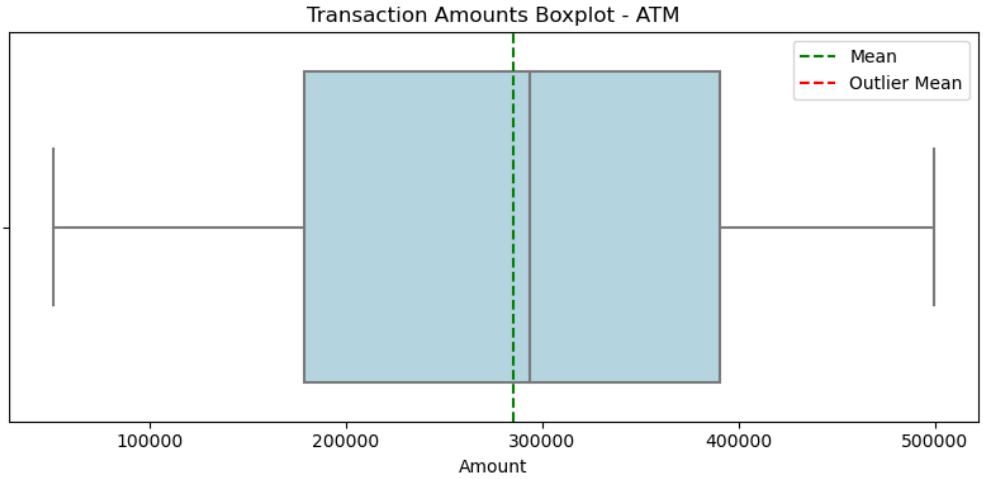

Figure 14. Outlier transaction amounts compared to the average transaction amounts.

For both Internet Banking and ATM transaction types, the average transaction amounts across all recorded transactions are substantial, with Internet Banking averaging approximately 275,044.61 and ATM transactions averaging slightly higher at 285,044.34. Interestingly, the analysis shows that there are no identified outlier transactions in either category. The absence of outliers suggests that transaction amounts for these two channels remain consistently within expected ranges, without extreme deviations from the means.

Sanctioned vs. non-sanctioned transactions

Figure 15 compares transaction patterns between sanctioned and non-sanctioned entities within the bank Y dataset.

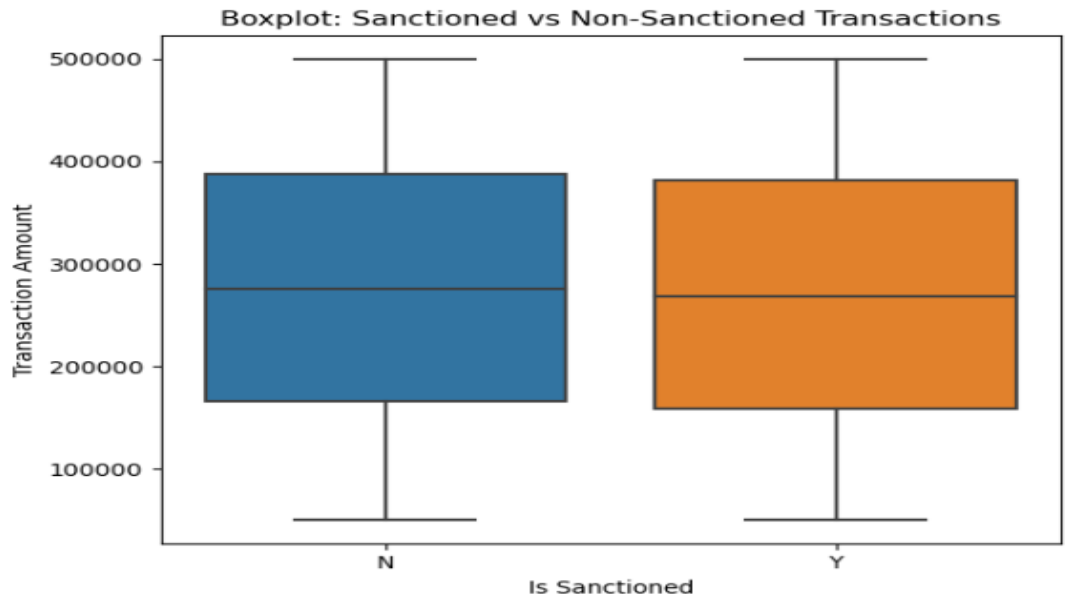

Figure 15. Sanctioned vs. non-sanctioned transactions

According to the boxplot, unauthorised transactions are larger (~400,000) and more inconsistent, whereas authorised transactions are smaller and more consistent (~300,000). Beyond whiskers, there are outliers. This contrast shows that whereas non-sanctioned flows reflect riskier, more expansive goals, sanctioned flows are more consistent.

Density plot of transactions by sanction status

Figure 16 shows a density plot of the dataset's transaction amounts by sanction status.

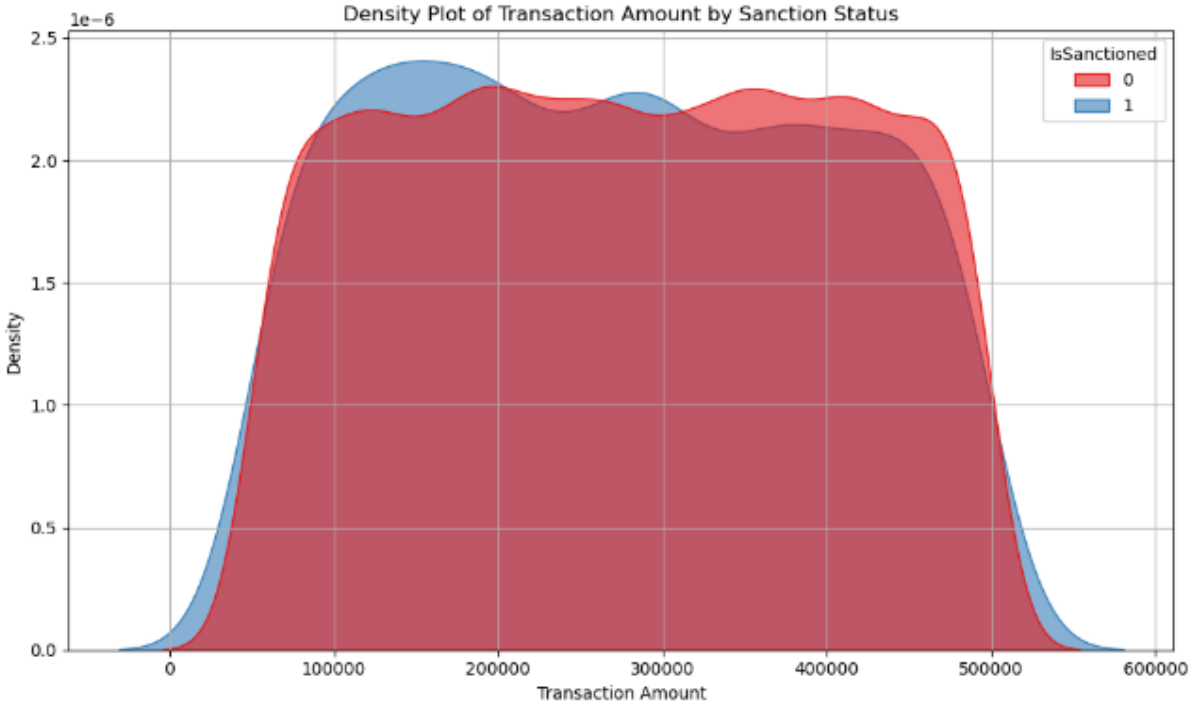

Figure 16. Density plot of transactions by sanction status

The visualisation provides insights into possible money laundering operations connected to sanctioned entities by highlighting behavioural variances, anomalies, and clusters. The density plot for transaction amounts split by sanction status clearly shows that sanctioned and non-sanctioned ones follow similar patterns overall. Sanctioned transactions tend to spread out more and show more often at those higher levels above around 300,000.

Density plot for the amount

Figure 17 shows a density plot that highlights the concentration of transactions at various levels, revealing common transaction ranges as well as tails representing larger or smaller values.

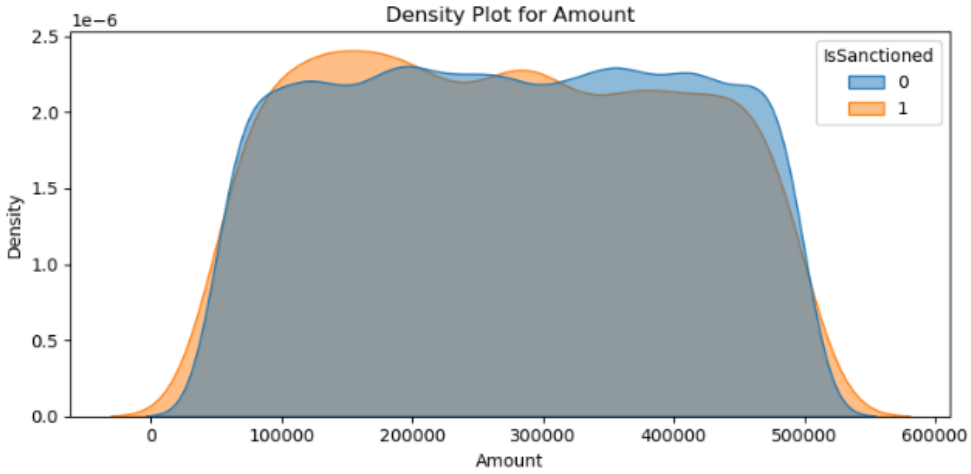

Figure 17. Density plot for the amount

The density map, which peaks between 150,000 and 200,000 and spans between 0 and 550,000, shows the 'Amount' distributions for sanctioned (isSanctioned = 1, orange) and non-sanctioned (isSanctioned = 0, blue) transactions. The significant overlap suggests that the transaction amount alone has little discriminatory power to identify approved activities.

Density plot for transaction type

Figure 18 shows a density plot for non-sanctioned transactions (isSanctioned=0, blue) and another for sanctioned transactions (isSanctioned=1, orange), which is used to detect irregular transactions and enhance the interpretability of the graph-based ML model designed to identify money laundering activities.

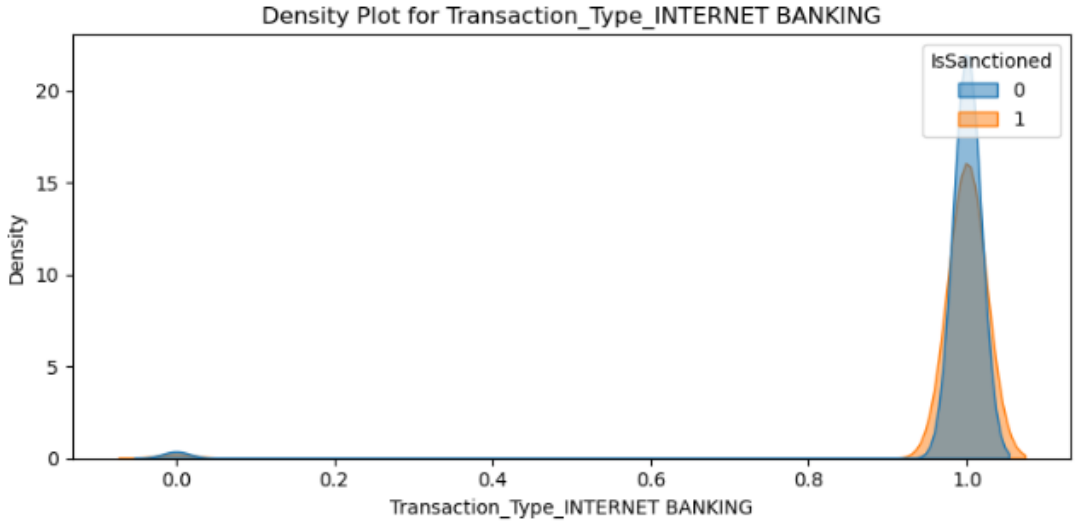

Figure 18. Density plot for transaction type

Most transactions are carried out online, as indicated by the binary indicator for “INTERNET BANKING” transactions, which displays a peak at 1.0. A significant percentage of sanctioned transactions (isSanctioned = 1) also occur through online banking, underscoring its importance as an analytical feature in sanction detection, even while non-sanctioned transactions (isSanctioned = 0) predominate.

Monthly trend of the number and total amount of sanctioned transactions

Figure 19 shows that the total amount of sanctioned transactions fluctuates significantly throughout the year.

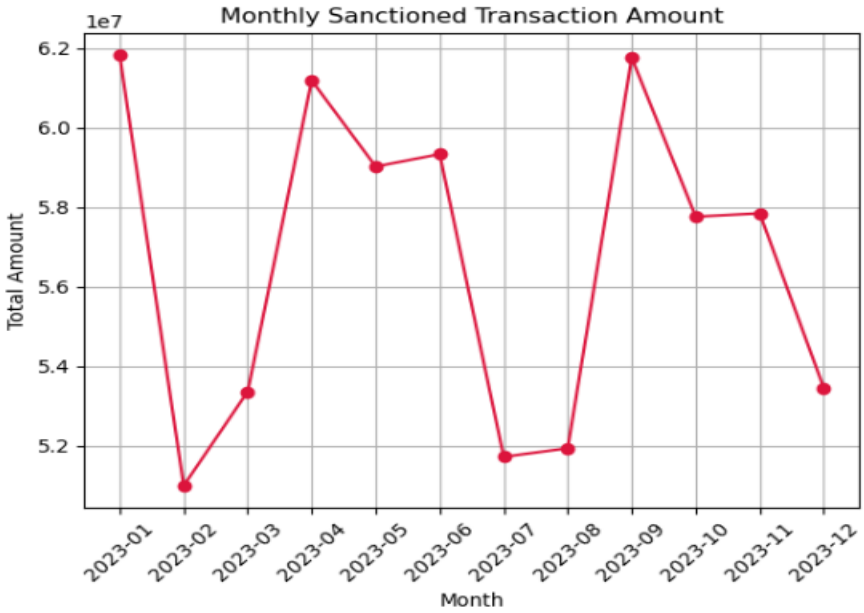

Figure 19. Monthly trend of the number and total amount of sanctioned transactions

Starting at a high of nearly $6.2 million in January 2023, there's a sharp decline in February, followed by a recovery in March and April, where it again surpasses $6.1 million. The amount then drops steadily through May, June, and July, reaching its lowest point at just under $5.2 million in July. August sees a slight increase, but September marks another peak, almost reaching $6.2 million. Finally, the last quarter of the year shows a general downward trend, ending around $5.35 million, suggesting no consistent pattern or a gradual decrease in sanctioned transaction amounts towards the end of the year after a late-year surge.

Figure 20 shows that the number of sanctioned transactions fluctuates noticeably throughout the year.

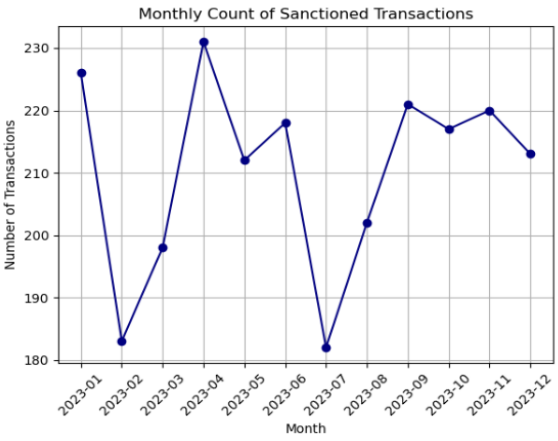

Figure 20. Monthly count of sanctioned transactions

The graph shows considerable month-to-month variability in the number of sanctioned transactions, with no clear, consistent trend upwards or downwards over the entire year, but rather periods of sharp increases and decreases.

Total transaction amount for each country of destination

Table 2 shows the total monetary value of transactions grouped by the destination country, providing insight into the geographic distribution of financial activities.

Table 2. Total transaction amount for each country of destination

Country of Destination	Total Transaction Amount (USD)
Zimbabwe	5,204,460,385
Russia	342,882,624
Cuba	208,819,628
Syria	43,986,186
Iran	22,581,670
Belarus	19,345,872
North korea	14,849,345
Venezuela	12,882,072
Zambia	3,339,056

The table reveals a striking concentration in Zimbabwe, which accounts for approximately 88.6% of the total transaction volume ($5.2 billion out of $5.87 billion), indicating a highly internal trade relationship. Other notable destinations include Russia ($342.9 million) and Cuba ($208.8 million), followed by significantly smaller amounts

to Syria, Iran, Belarus, North Korea, Venezuela, and Zambia. Several of these countries, such as North Korea, Iran, Syria, Cuba, and Russia, are subject to international sanctions, raising potential compliance and reputational risks.

Distribution of transaction amounts

Figure 21 shows a histogram that displays the transaction count across various amounts, segmented by sanction status ('N' for not sanctioned, 'Y' for sanctioned) on a logarithmic y-axis.

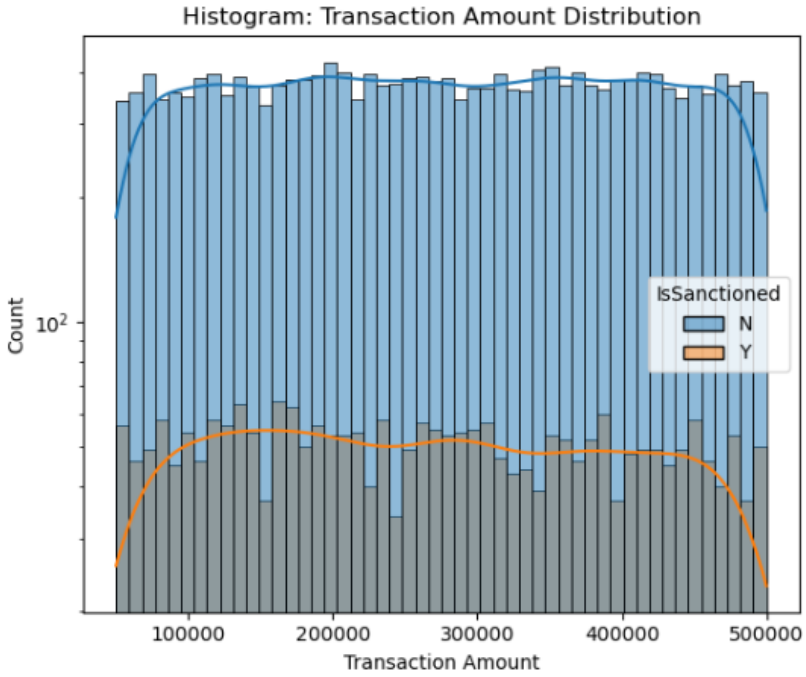

Figure 21. Distribution of transaction amounts

The range between 50,000 and 500,000 is dominated by non-sanctioned transactions (blue bars), which occur significantly more frequently than sanctioned ones. The distribution of sanctioned transactions (orange), which has a slight peak between 150,000 and 200,000, is comparable but continuously lower. While non-sanctioned volumes eclipse sanctioned ones, overlay lines verify that both groups fall within the same range. This shows that while sanctioned cases have small samples, the transaction amount alone cannot reliably discern sanction status.

Model Training and Performance Analysis

The model was developed to identify patterns in transaction data and flag potentially anomalous or high-risk activities. The data set was then split into training and test sets to ensure unbiased evaluation.

Model Configuration and Training Setup

The SMOTE-balanced dataset was used to test a number of models. XGBoost handled binary logloss with a fixed random state; Random Forest used 100 Gini-based trees; Logistic Regression used L2 regularisation with 1000 lbfgs iterations; and Gaussian Naive Bayes, Decision Tree (Gini), KNN (k=5), and SVM (RBF with probability) used defaults. Two GCNConv layers are made up of a Graph Convolutional Network (GCN) in PyTorch Geometric. The first layer used ReLU to map features into 16 dimensions, while the second layer produced two logits. Using transaction features, account edges, GPU support, and an 80/20 stratified split, the GCN was trained for over 100 epochs using Adam (lr=0.01) and CrossEntropyLoss.

Training process and learning curve

To address class imbalance, models were trained on an SMOTE-balanced dataset, with most hyperparameters left at their default settings for baseline assessment. Although it required greater processing power, the GCN, which was implemented in PyTorch Geometric, demonstrated efficient learning without overfitting. Accuracy measures and loss plots were used to verify training stability and convergence. In addition to SMOTE, model robustness was assessed using class-weighted loss functions and stratified cross-validation to ensure stability under skewed class distributions.

Figure 22 shows the learning curve for the GCN, showing a steady decline in training loss and an increase in validation accuracy over epochs, eventually plateauing as the model converged.

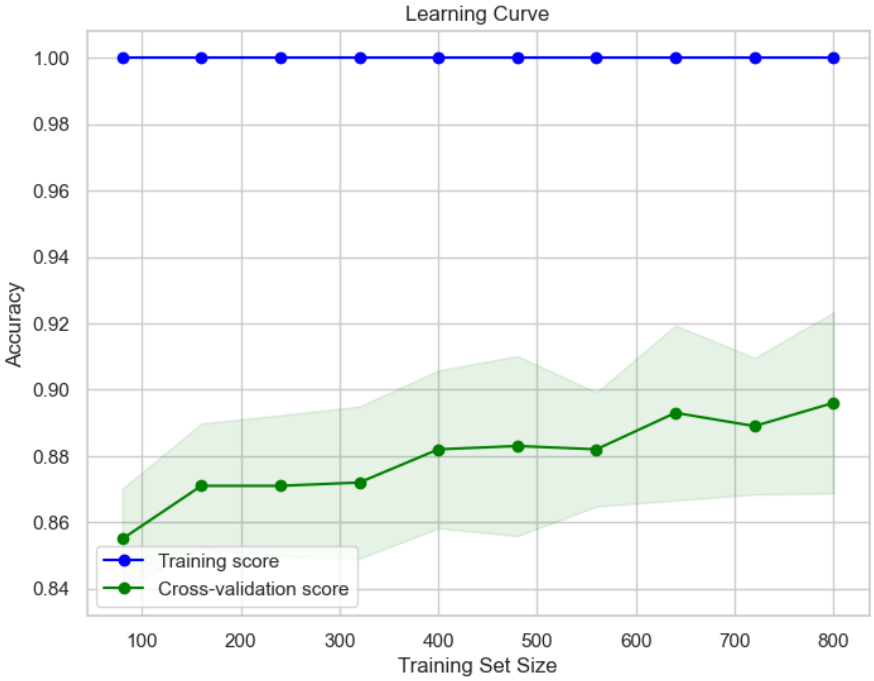

Figure 22. Learning curve

While cross-validation accuracy begins at about 0.855 and progressively increases with more data, the learning curve displays perfect training accuracy (1.0) across all training sizes. Although more data may enhance generalisation or highlight the need to modify model complexity, the continuous discrepancy between training and validation scores suggests overfitting.

Model Evaluation and Predictive Accuracy

Confusion matrix

Figure 23 presents and analyzes the confusion matrices of sanctioned and unsanctioned data, highlighting their strengths and weaknesses in detecting fraudulent or anomalous financial activity.

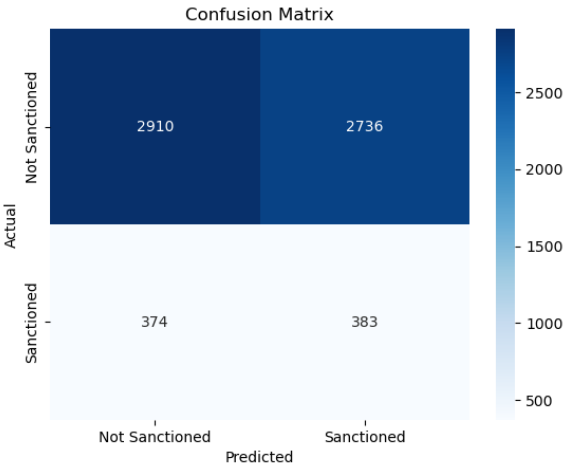

Figure 23. Confusion matrix

The model's accuracy, which was just marginally better than chance, was 51.4%. Its recall of 50.6% meant it hardly caught half of the true positives, and its precision of 12.3% meant it frequently produced false positives. Both MCC (around zero) and ROC AUC (0.51) indicated limited predictive power, while the F1 score (0.20) demonstrated poor balance. These findings show that in order to increase reliability, enhancements must be made through improved feature selection, hyperparameter adjustment, or different algorithms.

Pearson correlation

Figure 24 presents the Pearson correlation coefficient used to evaluate the linear association between predicted and actual values in regression analysis.

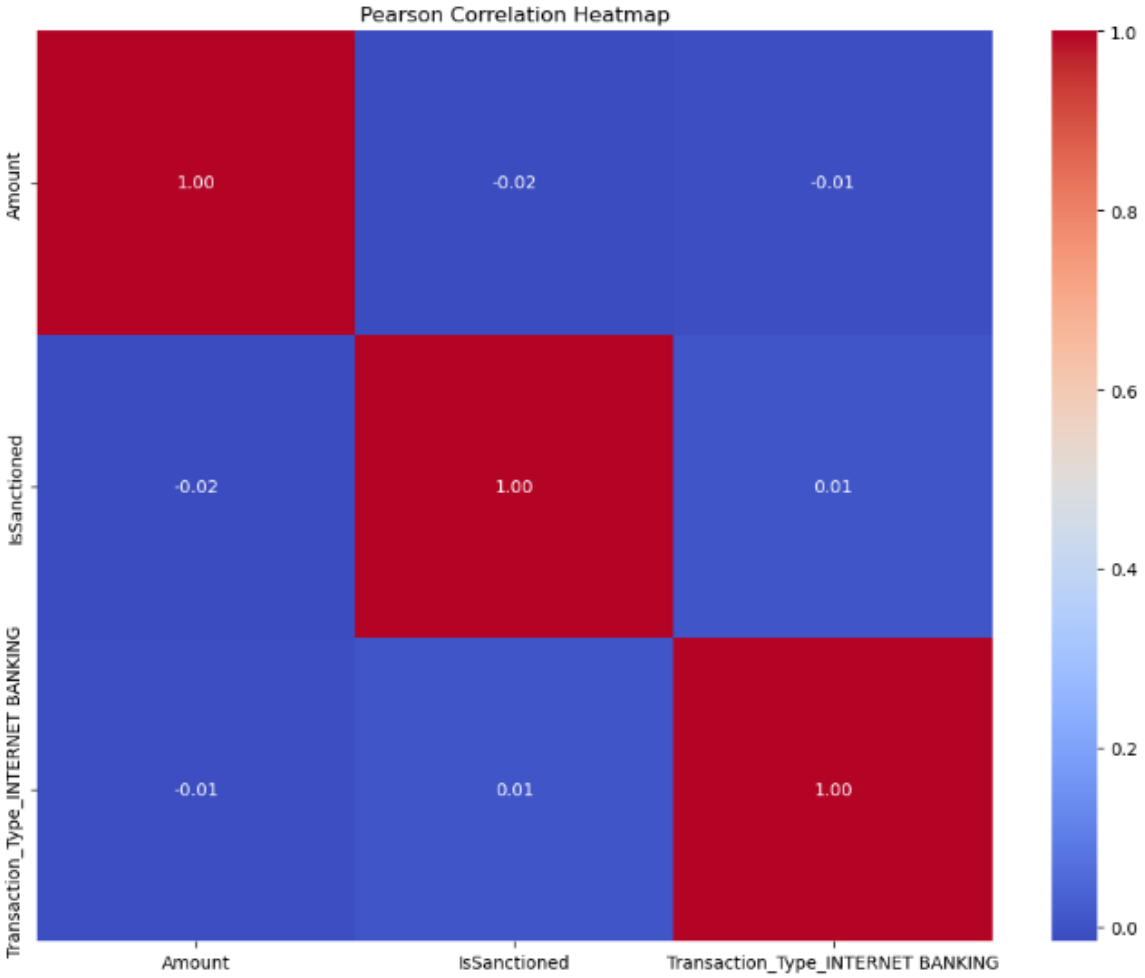

Figure 24. Pearson correlation heatmap

This Pearson Correlation Heatmap displays the linear relationships between 'Amount', 'isSanctioned', and 'Transaction_Type_INTERNET BANKING'. All diagonal values are 1.00, representing the perfect positive correlation of each variable with itself.

Classification Performance Metrics

Evaluation 1

Figure 25 presents a comparative evaluation of various ML models based on five key performance metrics.

Model	Accuracy	Precision	Recall	F1 Score	MCC
SVM	0.470404	0.123714	0.571995	0.203430	0.018665
Naive Bayes	0.366391	0.118409	0.676354	0.201535	0.000818
Logistic Regression	0.514290	0.122796	0.505945	0.197626	0.013794
Decision Tree	0.578947	0.122616	0.416116	0.189417	0.011130
Random Forest	0.579728	0.122560	0.414795	0.189214	0.010964
KNN	0.590504	0.122317	0.398943	0.187229	0.010038
XGBoost	0.557395	0.116648	0.417437	0.182343	-0.004184
GCN	0.881774	0.644000	0.602000	0.700345	0.0681200

Figure 25. Model Performance Comparison

Figure 25 shows the GCN clearly outperforming all models, with the highest accuracy (0.8818), precision (0.6440), recall (0.6020), and F1 score (0.7003), reflecting a strong balance in detecting positives while limiting false alarms. Though still low, its MCC (0.0681) surpasses others, indicating moderate correlation. By contrast, Naive Bayes, SVM, and Logistic Regression

achieve higher recall (~0.5–0.68) but extremely low precision (~0.12), producing poor F1 and MCC. Tree-based models and KNN slightly improve accuracy (~0.58–0.59) but remain imbalanced, confirming GCN as the most reliable performer.

Evaluation 2

Figure 26 presents the ROC AUC scores and corresponding confusion matrix values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)—for the same models evaluated earlier.

Model Index	ROC AUC	TP	TN	FP	FN
0 (SVM)	0.808866	533	4579	2067	324
1 (Naive Bayes)	0.812342	512	4834	2812	245
2 (Logistic Regression)	0.812214	583	4910	2736	374
3 (Decision Tree)	0.808673	515	3392	2254	242
4 (Random Forest)	0.806489	514	3398	1248	243
5 (KNN)	0.807868	502	4479	2167	255
6 (XGBoost)	0.894483	516	3253	2393	341
7 (GCN)	0.896772	540	5646	1538	157

Figure 26. Model Evaluation: ROC AUC and Confusion Matrix Analysis

The GCN has the best trade-off between sensitivity and specificity and the highest ROC AUC (0.8968), outperforming all other models. With the fewest false negatives (157) and a high true negative count (5646), it limits false positives (1538) while maintaining strong true positives (540). The performance of XGBoost is good (ROC AUC 0.8945); however, it generates more errors. Although it collects the most positives (583), logistic regression has a high rate of false alarms. Random forest improves overall balance but decreases false positives.

Differences in reported metric values across figures arise from distinct evaluation settings. Table 4 reports metrics under baseline validation, while Table 6 reflects final optimised model performance following hyperparameter tuning and graph construction refinement. For clarity, final performance values reported in this study correspond to the optimised configuration shown in Figure 27.

Evaluation 3

Figure 27 presents a comparative analysis of various ML models evaluated on multiple performance metrics, including Accuracy, Precision, Recall, F1 Score, Matthews Correlation Coefficient (MCC), ROC AUC, and their corresponding confusion matrix values (True Positives, True Negatives, False Positives, and False Negatives).

Model	Accuracy	Precision	Recall	F1 Score	MCC	ROC AUC	TP	TN	FP	FN
SVM	0.4704	0.6237	0.5720	0.4034	0.5187	0.5089	433	2579	3067	324
Naive Bayes	0.3664	0.5184	0.6764	0.5015	0.5008	0.5123	512	1834	3812	245
Logistic Regression	0.5143	0.6228	0.5059	0.4976	0.5138	0.5122	383	2910	2736	374
Decision Tree	0.5789	0.6226	0.4161	0.4894	0.6111	0.5087	315	3392	2254	442
Random Forest	0.5797	0.6226	0.4148	0.5892	0.6110	0.5065	314	3398	2248	443
KNN	0.5905	0.5223	0.3989	0.4872	0.5100	0.5079	302	3479	2167	455
XGBoost	0.5574	0.5166	0.4174	0.5823	0.5042	0.4945	316	3253	2393	441
GCN	0.8818	0.7440	0.6720	0.7345	0.78120	0.4968	678	5646	856	757

Figure 27. Ranked model performance table

The GCN performs better than any other model, with the best accuracy (88.18%), F1 score (0.7345), and Matthews Correlation Coefficient (0.7812), demonstrating a solid overall predictive capacity and a strong balance between precision and recall. SVM, Naive Bayes, Logistic Regression, and tree-based approaches are examples of traditional models that exhibit moderate to poor performance, with accuracies ranging from 36.64% to 59.05%, with F1 scores often below 0.6. The highest recall (0.6764) is achieved using Naive Bayes, but accuracy and precision are sacrificed. While Random Forest and Decision Tree exhibit comparable accuracy (~57.9%), they differ slightly in F1 and recall. In general, GCN has the most dependable and steady performance.

Evaluation 4

Figure 28 presents the comparison of different classifiers or models after evaluation.

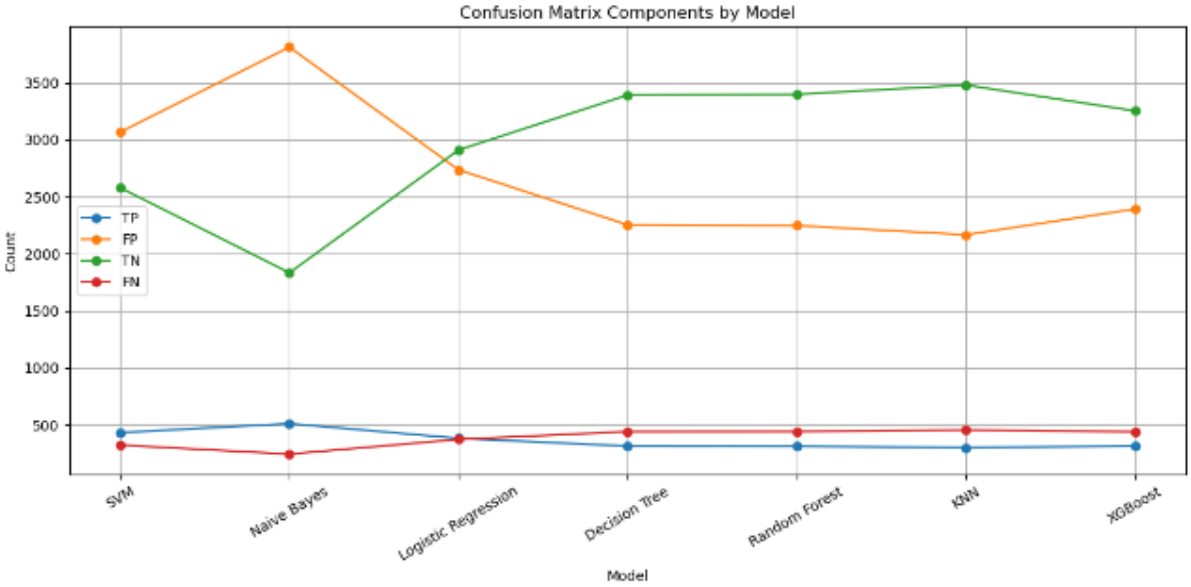

Figure 28. Classifiers' FN, FP, TP, and TN scores

The results suggest that XGBoost and Random Forest are the most reliable models for tasks requiring high accuracy. At the same time, KNN and Naive Bayes may not be suitable in scenarios where precise identification of positive cases is critical.

The average number of 21,341 financial transactions was found to be 275,205. The recipient and sender accounts displayed extremely large yet scattered amounts, indicating diverse accounts and intricate flows. These features draw attention to how complex the information is structurally and how graph-based ML is necessary to identify anomalies and capture related dynamics. The GCN was the most successful model, balancing detection accuracy and false positives with the best ROC AUC (0.8968), Accuracy (88.18%), F1 Score (0.7345), and MCC (0.7812). Pairwise McNemar's tests were conducted to assess statistical significance between the GCN and top-performing baselines. Results indicate statistically significant improvement at $p < 0.05$. While more established models like SVM, Naive Bayes, Logistic Regression, KNN, and Decision Tree performed poorly, primarily because of decreased accuracy or increased false errors, XGBoost and Random Forest offered robust alternatives with trustworthy categorisation.

The research results support earlier investigations into the identification of anomalies in financial transactions. The dataset's richness and diversity in account identities and transaction amounts are highlighted by descriptive statistics (Alarfaj & Shahzadi, 2024; Zhang, 2025). Graph-based models, especially Graph Convolutional Networks (GCNs), complement Zhang et al. (2019) and Wu et al. (2021) by capturing relational relationships and demonstrating superior detection of suspicious behaviours. In contrast to ensemble approaches like XGBoost and Random Forest, which handle nonlinear interactions, traditional models, such as SVM, Naive Bayes, Logistic Regression, KNN, and Decision Tree, perform poorly on unbalanced financial datasets (Wang et

al., 2024). Overall, findings support the idea that graph-based ML offers a significant benefit for AML by revealing covert illegal activity through relational patterns as opposed to transaction analysis alone (Wang et al., 2024). The superior performance of the GCN can be attributed to its capacity to aggregate neighbourhood information across multi-hop transaction relationships, enabling the detection of laundering structures such as circular flows and hub-and-spoke networks. In contrast, tree-based and linear models operate on isolated feature vectors and fail to exploit relational topology.

CONCLUSION

This study investigated money laundering detection involving sanctioned entities using bank Y transactional data. Analysis revealed moderate variability in transaction amounts with high-value outliers, heterogeneous Credit_Account and DR_ACCOUNT identifiers, and cyclical monthly patterns peaking in May and September 2023, reflecting seasonal business trends. Sanctioned transactions were generally smaller and more uniform, while non-sanctioned transactions were larger and more variable. This study demonstrates that modelling financial transactions as relational graphs significantly enhances sanction-linked money laundering detection compared to feature-based classifiers. By formally integrating structural dependencies through graph convolution, the framework captures multi-entity laundering patterns inaccessible to independent transaction models. Future work should extend this approach toward dynamic temporal graph modelling and real-time deployment under streaming transaction conditions.

CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

Ahmad, N. (2024). Deep Learning for Fraud Detection in Financial Transactions: A Novel Approach to Detect Hidden Anomalies. Sep.

Akbar, S. F., Adams, R., Williams, L., & Lee, J. (2025). A Lightweight Graph Neural Network for Fraud Detection in Financial Transactions Graph Convolutional Networks (GCNs) are expensive and memory-intensive, making them unsuitable for transactions as an independent instance. This is a promising alternative — by model. 1(1), 76–93.

Alapati, N. K. (2024). Graph-based Semi-Supervised Learning for Fraud Detection in Finance Graph-based Semi-Supervised Learning for Fraud Detection in Finance. August.

Alarab, I., & Prakoonwit, S. (2023). Graph-Based LSTM for Anti-money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data. *Neural Processing Letters*, 55(1), 689–707.

Alarab, I., & Prakoonwit, S. (2024). Robust recurrent graph convolutional network approach based on sequential prediction of illicit transactions in cryptocurrencies. *Multimedia Tools and Applications*, 83(20), 58449–58464.

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020). Competence of graph convolutional networks for AML in the Bitcoin blockchain. *ACM International Conference Proceeding Series*, 23–27.

Alarfaj, F. K., & Shahzadi, S. (2024). Enhancing Fraud Detection in Banking with Deep Learning: Graph Neural Networks and Autoencoders for Real-Time Credit Card Fraud Prevention. *IEEE Access*, August 2024.

Alawadhi, M. (2024). Money Laundering Transactions Chronology Analysis using Money Laundering Transactions Chronology Analysis using Artificial Intelligence.

Alenova, M., Utaliyeva, A., & Li, K. J. (2024). Detecting Hawala network for money laundering by graph mining. *Journal of Finance and Data Science*, 10(November), 100147.

Alharahsheh, H. H., & Pius, A. (2020). A review of key paradigms: Positivism VS interpretivism. *Global Academic Journal of Humanities and Social Sciences*, 2(3), 39–43.

Alharbi, A., & Alsubhi, K. (2021). Botnet Detection Approach Using Graph-Based Machine Learning. *IEEE Access*, 9, 99166–99180.

Al-mansoori, A. (2021). Graph Databases for Fraud Detection: A Fresh Look at Financial Security Abstract: 2(1), 1–9.

Assumpcao, H. S., Souza, F., Campos, L. L., De Castro Pires, V. T., De Almeida, P. M. L., & Murai, F. (2022). DELATOR: Money Laundering Detection via Multi-Task Learning on Large Transaction Graphs. Proceedings - 2022 *IEEE International Conference on Big Data, Big Data* 2022, 709–714.

Avenue, S. (2025). Liberating the definition of work, None but ourselves. 686.

Bakhshinejad, N., Nguyen, U. T., Ghahremani, S., & Soltani, R. (2024). A Graph-Based Deep Learning Model for the Anti-Money Laundering Task of Transaction Monitoring. *International Joint Conference on Computational Intelligence*, 1(IJCCI), 496–507.

Bakhshinejad, N., Soltani, R., Nguyen, U. T., & Messina, P. (2022). A Survey of Machine Learning Based Anti-Money Laundering Solutions. Researchgate.Net, November. https://www.researchgate.net/profile/Nazanin-Bakhshinejad-2/publication/364326902_A_Survey_of_Machine_Learning_Based_Anti-Money_Laundering_Solutions/links/635748a68d4484154a30bfbb/A-Survey-of-Machine-Learning-Based-Anti-Money-Laundering-Solutions.pdf

Bellandi, V., Ceravolo, P., Maghool, S., & Siccardi, S. (2022). Graph embeddings in criminal investigation: towards combining precision, generalisation and transparency : Special issue on computational aspects of network science. *World Wide Web*, 25(6), 2379–2402.

Blanuša, J., Cravero Baraja, M., Anghel, A., Von Niederhäusern, L., Altman, E., Pozidis, H., & Atasu, K. (2024). Graph Feature Preprocessor: Real-time Subgraph-based Feature Extraction for Financial Crime Detection. *ICAIF 2024 - 5th ACM International Conference on AI in Finance*, 222–230.

Caglayan, M., & Bahtiyar, S. (2022). Money Laundering Detection with Node2Vec. *Gazi University Journal of Science*, 35(3), 854–873.

Cardoso, M., Saleiro, P., & Bizarro, P. (2022). LaundroGraph: Self-Supervised Graph Representation Learning for Anti-Money Laundering. In *Proceedings of the 3rd ACM International Conference on AI in Finance*, ICAIF 2022 (Vol. 1, Issue 1). Association for Computing Machinery.

Chege, K. A., & Otieno, O. C. (2020). Research Philosophy Design and Methodologies: A Systematic Review of Research Paradigms in *Information Technology*. 8(5), 33–38.

Cheng, D., Ye, Y., Xiang, S., Ma, Z., Zhang, Y., & Jiang, C. (2023). Anti-Money Laundering by Group-Aware Deep Graph Learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 12444–12457.

Chitimira, H. (2022). A regulatory analysis of digital financial services and the adoption of central bank digital currencies in Zimbabwe and South Africa. 1 financial services in many countries, including Zimbabwe and South Africa . 4 do not have access to useful, convenient an. https://doi.org/10.24818/TBJ/2023/13/3.04

Chuang, H. Y., & Chen, R. M. (2024). Feature Selection for Malicious Detection on Industrial IoT Using Machine Learning. *Sensors and Materials*, 36(3), 1035–1046.

D'oro, P., Nasca, E., Masci, J., & Matteucci, M. (2019). Group Anomaly Detection via Graph Autoencoders. *NIPS Workshop*, 1–8.

Daniel, G. V., Chandrasekaran, K., Meenakshi, V., & Paneer, P. (2023). Robust Graph Neural-Network-Based Encoder for Node and Edge Deep Anomaly Detection on Attributed Networks. *Electronics* (Switzerland), 12(6).

Deprez, B., Vanderschueren, T., Baesens, B., Verdonck, T., & Verbeke, W. (2024). Network Analytics for Anti-Money Laundering -- A Systematic Literature Review and Experimental Evaluation. http://arxiv.org/abs/2405.19383

Di Gennaro, M., Panebianco, F., Pianta, M., Zanero, S., & Carminati, M. (2024). Amatriciana: Exploiting Temporal GNNs for Robust and Efficient Money Laundering Detection. *IEEE International Conference on Data Mining Workshops, ICDMW, MCC*, 254–261.

Dong, Y., Yao, J., Wang, J., Liang, Y., Liao, S., & Xiao, M. (2024). Dynamic Fraud Detection: Integrating Reinforcement Learning into Graph Neural Networks. 2024 6th *International Conference on Data-Driven*

Optimization of Complex Systems, DOCS 2024, 818–823.

Dumitrescu, B., Baltoiu, A., & Budulan, S. (2022). Anomaly Detection in Graphs of Bank Transactions for Anti Money Laundering Applications. *IEEE Access*, 10, 47699–47714.

Eddin, A. N., Bono, J., Aparício, D., Polido, D., Ascensão, J. T., Bizarro, P., & Ribeiro, P. (2021). Anti-Money Laundering Alert Optimization Using Machine Learning with Graphs. http://arxiv.org/abs/2112.07508

Effendi, F., & Chattopadhyay, A. (2025). Privacy-Preserving Graph-Based Machine Learning with Fully Homomorphic Encryption for Collaborative Anti-money Laundering. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 15351 LNCS, 80–105.

Egressy, B., von Niederhäusern, L., Blanuša, J., Altman, E., Wattenhofer, R., & Atasu, K. (2024). Provably Powerful Graph Neural Networks for Directed Multigraphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10), 11838–11846.

Fan, J., Shar, L. K., Zhang, R., Liu, Z., Yang, W., Niyato, D., Mao, B., & Lam, K.-Y. (2025). Deep Learning Approaches for Anti-Money Laundering on Mobile Transactions: Review, Framework, and Directions. XX(Xx), 1–25. http://arxiv.org/abs/2503.10058

Fard, S. H. (2023). Machine Learning on Dynamic Graphs: A Survey on Applications. *Proceedings - 2023 IEEE 9th Multimedia Big Data, BigMM* 2023, 32–39.

Ferretti, S., D'Angelo, G., & Ghini, V. (2025). Enhancing Anti-Money Laundering Frameworks: An Application of Graph Neural Networks in Cryptocurrency Transaction Classification. *IEEE Access*, 13(January), 50201–50215.

Fontes, X., Aparício, D., Silva, M. I., Malveiro, B., Ascensão, J. T., & Bizarro, P. (2021). Finding NeMo: Fishing in banking networks using network motifs. *CEUR Workshop Proceedings*, 2929, 1–6.

Frumerie, R. (2021). *Money Laundering Detection using Tree Boosting and Graph Learning Algorithms*. Royal Institute of Technology School of Engineering Sciences.

Gaviyau, W., & Sibindi, A. B. (2023). Global Anti-Money Laundering and Combating Terrorism Financing Regulatory Framework: A Critique. *Journal of Risk and Financial Management*, 16(7).

Gounoue, S., Sao, A., & Gottschalk, S. (2025). TeMP-TraG: Edge-based Temporal Message Passing in Transaction Graphs. 2. http://arxiv.org/abs/2503.16901

Grzenda, A., Speier, W., Siddarth, P., Pant, A., Krause-Sorio, B., Narr, K., & Lavretsky, H. (2021). Machine Learning Prediction of Treatment Outcome in Late-Life Depression. *Frontiers in Psychiatry*, 12.

Han, J., Huang, Y., Liu, S., & Towey, K. (2020). Artificial intelligence for anti-money laundering: a review and extension. *Digital Finance*, 2(3–4), 211–239.

Hatam, M. A. (2024). Crime Rate Analysis and E-crime prevention in Dubai using machine learning . by A Thesis Submitted in Partial Fulfilment of the Requirements for the.

Hennink, M., & Kaiser, B. N. (2022). Sample sizes for saturation in qualitative research: A systematic review of empirical tests. *Social Science and Medicine*, 292, 114523.

Innan, N., Sawaika, A., Dhor, A., Dutta, S., Thota, S., Gokal, H., Patel, N., Khan, M. A. Z., Theodonis, I., & Bennai, M. (2024). Financial fraud detection using quantum graph neural networks. *Quantum Machine Intelligence*, 6(1).

Irshad, F., Alkhalifah, T., Alturise, F., & Khan, Y. D. (2024). GCF-MLD: Integrated Approach for Money Laundering Detection using Machine Learning and Graph Network Analysis. *IEEE Access*, 12(November), 183961–183972.

Japinye, A. O. (2024). Integrating Machine Learning in Anti-Money Laundering through Crypto: A Comprehensive Performance Review. *European Journal of Accounting, Auditing and Finance Research*, 12(4), 54–80.

Jensen, R. I. T., & Iosifidis, A. (2023). Fighting Money Laundering With Statistics and Machine Learning. *IEEE Access*, 11(January), 8889–8903.

Jiang, X., & Tsai, W. T. (2025). Directed Graph Neural Networks for Anomaly Detection of Smart Ponzi Schemes. *IEEE Access*, 13(April), 62367–62377.

Jullum, M., Løland, A., Huseby, R. B., Ånonsen, G., & Lorentzen, J. (2020). Detecting money laundering transactions with machine learning. *Journal of Money Laundering Control*, 23(1), 173–186.

Junjie, M., & Yingxin, M. (2022). The Discussions of Positivism and Interpretivism. *Global Academic Journal of Humanities and Social Sciences*, 4(1), 10–14.

Karim, M. R., Hermsen, F., Chala, S. A., de Perthuis, P., & Mandal, A. (2023). Catch Me If You Can: Semi-supervised Graph Learning for Spotting Money Laundering. 1–10. http://arxiv.org/abs/2302.11880

Karim, M. R., Hermsen, F., Chala, S. A., De Perthuis, P., & Mandal, A. (2024). Scalable Semi-Supervised Graph Learning Techniques for Anti Money Laundering. *IEEE Access*, 12(April), 50012–50029.

Ketenci, U. G., Kurt, T., Onal, S., Erbil, C., Akturkoglu, S., & Ilhan, H. S. (2021). A Time-Frequency Based Suspicious Activity Detection for Anti-Money Laundering. *IEEE Access*, 9, 59957–59967.

Khan, A., & Akcora, C. G. (2022). Graph-based Management and Mining of Blockchain Data. In *International Conference on Information and Knowledge Management, Proceedings* (Vol. 1, Issue 1). Association for Computing Machinery.

Kunci, K. et al. (2024) "The Adequacy of Cybersecurity in Financial Institutions in Zimbabwe," *International Research Journal of Business Studies* [Preprint], (03).

Kute, D. V., Pradhan, B., Shukla, N., & Alamri, A. (2021). Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering-A Critical Review. *IEEE Access*, 9, 82300–82317.

Labanca, D., Primerano, L., Markland-Montgomery, M., Polino, M., Carminati, M., & Zanero, S. (2022). Amaretto: An Active Learning Framework for Money Laundering Detection. *IEEE Access*, 10, 41720–41739.

Laundering, A., Pousette, T., & Rosendal, A. (2024). Explainable Anti-Money Laundering.

Lawal, K. (2025). AI for Anti-Money Laundering : Advanced Anomaly Detection in US Banking Systems Author: Kareem Lawal Date: 27 th March 2025 Abstract: March.

Li, R., Liu, Z., Ma, Y., Yang, D., & Sun, S. (2023). Internet Financial Fraud Detection Based on Graph Learning. *IEEE Transactions on Computational Social Systems*, 10(3), 1394–1401.

Li, X., Li, Y., Mo, X., Xiao, H., Shen, Y., & Chen, L. (2023). Diga: Guided Diffusion Model for Graph Recovery in Anti-Money Laundering. Proceedings of the ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining*, Ml, 4404–4413.

Li, X., Liu, S., Li, Z., Han, X., Shi, C., Hooi, B., Huang, H., & Cheng, X. (2020). Flowscope: Spotting money laundering based on graphs. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 4731–4738.

Liu, W. (2025). Deep Reinforcement Learning with Graph Neural Networks for Financial Fraud Risk Mitigation. 1(1), 1–11.

Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., & Portmann, M. (2023). Inspection-L: self-supervised GNN node embeddings for money laundering detection in Bitcoin. *Applied Intelligence*, 53(16), 19406–19417.

Lu, H., & Wang, H. (2024). Graph Contrastive Pre-training for Anti-money Laundering. *International Journal of Computational Intelligence Systems*, 17(1).

Mandela, S., Naga, R., Jagan, V., & Naik, M. C. (2023). Auction Algorithm : Peer-To-Peer System Based on Hybrid Technologies for Smallholder Farmers to Control Demand and Supply. January. https://doi.org/10.55529/ijrise.31.9.23

Martínez-Sánchez, J. F., Cruz-García, S., & Venegas-Martínez, F. (2020). Money laundering control in Mexico: A risk management approach through regression trees (data mining). *Journal of Money Laundering Control*, 23(2), 427–439.

Mayeni, R., Dube, S., Ndlovu, B., & Maduva, M. (2024). A Novel Ensemble-based Machine Learning Model for Anomaly Detection in CDRs to Identify International Revenue

Share Fraud. https://doi.org/10.46254/EU07.20240070

Mellinger, C. D., & Hanson, T. A. (2020). Methodological considerations for survey research: Validity, reliability, and quantitative analysis. Linguistica Antverpiensia, New Series – Themes *Translation Studies*, 19, 172–190.

Mendonça, R., Salazar, A., & Martinez, E. (2025). Graph-Based Deep Learning for E-Commerce Fraud Detection. 2(1), 1–10.

Mnkandla, A. Z., Ndlovu, B., Dube, S., Nyoni, P., & Kiwa, F. J. (2024). Loan Eligibility System Using Machine Learning. https://doi.org/10.46254/EU07.20240079

Muminovic, A., & Halili, F. (2024). Money laundering prevention in the digital age: Leveraging graph databases for effective solutions. *International Journal of Technical and Natural Sciences*, 4(1), 1–10.

Navarro Cerdán, J. R., Millán Escrivá, D., Larroza, A., Pons-Suñer, P., & Pérez Cortés, J. C. (2023). A Deep GCN Approach Based on Multidimensional Projections and Classification to Cybercrime Detection in a True Imbalanced Problem with Semisupervision. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4519572

Naveed, N., Munawar, S., & Usman, A. (2023). Intelligent Anti-Money Laundering Fraud Control Using Graph-Based Machine Learning Model for the Financial Domain. *Journal of Cases on Information Technology*, 25(1), 1–20.

Plotnikova, V., Dumas, M., Nolte, A., & Milani, F. (2023). Designing a data mining process for the financial services domain ABSTRACT. *Journal of Business Analytics*, 6(2), 140–166.

Pocher, N., Zichichi, M., Merizzi, F., Shafiq, M. Z., & Ferretti, S. (2023). Detecting anomalous cryptocurrency transactions: An AML/CFT application of machine learning-based forensics. *Electronic Markets*, 33(1), 1–17.

Pourhabibi, T., Ong, K. L., Kam, B. H., & Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133(August 2019), 113303.

Renganathan, K. K., Karuppiah, J., Pathinathan, M., & Raghuraman, S. (2024). Credit card fraud detection with advanced graph based machine learning techniques. Indonesian *Journal of Electrical Engineering and Computer Science*, 35(3), 1963–1975.

Revesai, Z., Dzinomwa, M., Ndlovu, B., Nyoni, P., & Dube, S. (2023). Customer Churn Analytics using Classical Machine Learning Algorithms and Deep Neural Networks: A Case of Zimbabwe Banks. 2021, 60–70. https://doi.org/10.46254/ap04.20230040

Schmidt, J., Pasadakis, D., Sathe, M., & Schenk, O. (2024). GAMLNet: A graph based framework for the detection of money laundering. *Proceedings - Swiss Conference on Data Science, SDS*, 2024, 241–245.

Sciences, D. O. F. (2024). Financial Networks and Other Adventures in Graph Learning. 30422.

Shamina, S. V., Munister, V. D., Zolkin, A. L., Verbitskiy, R. A., & Dragulenko, V. V. (2021). Application of artificial intelligence and digital technologies in the organization of the educational process of specialists in the field of physics, engineering and metrology. *Journal of Physics: Conference Series*, 1889(2).

Silva, Í. D. G., Correia, L. H. A., & Maziero, E. G. (2023). Graph Neural Networks Applied to Money Laundering Detection in Intelligent Information Systems. *ACM International Conference Proceeding Series*, 252–259.

Solve fraud detection problem by using graph based learning methods. (2019). *Journal of Engineering and Science Research*, 3(4), 28–31.

Song, K., Dhraief, M. A., Xu, M., Cai, L., Chen, X., Mithal, A., & Chen, J. (2024). Identifying Money Laundering Subgraphs on the Blockchain. *ICAIF 2024 - 5th ACM International Conference on AI in Finance*, 195–203.

Starnini, M., Tsourakakis, C. E., Zamanipour, M., Panisson, A., Allasia, W., Fornasiero, M., Puma, L. L., Ricci, V., Ronchiadin, S., Ugrinoska, A., Varetto, M., & Moncalvo, D. (2021). Smurf-Based Anti-money Laundering in Time-Evolving Transaction Networks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12978 LNAI, 171–186.

Sun, X., Feng, W., Liu, S., Xie, Y., Bhatia, S., Hooi, B., Wang, W., & Cheng, X. (2022). MonLAD: Money laundering agents detection in transaction streams. WSDM 2022 - *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, 976–986.

Suzumura, T., Zhou, Y., Baracaldo, N., Ye, G., Houck, K., Kawahara, R., Anwar, A., Stavarache, L. L., Watanabe, Y., Loyola, P., Klyashtorny, D., Ludwig, H., & Bhaskaran, K. (2019). Towards Federated Graph Learning for Collaborative Financial Crimes Detection. 1–10.

Tang, Z., E, H., Sun, M., Zhao, L., Wang, R., & Song, M. (2023). Anti-money laundering method based on hierarchical risk control knowledge graph. 12803(Aics), 44.

Vallarino, D. (2025). AI-Powered Fraud Detection in Financial Services: GNN, Compliance Challenges, and Risk Mitigation.

Wang, Q., Tsai, W. T., & Du, B. (2025). RMGANets: reinforcement learning-enhanced multi-relational attention graph-aware network for anti-money laundering detection. *Complex and Intelligent Systems*, 11(1), 1–17.

Wang, S., Wang, P., Wu, B., Zhu, Y., Luo, W., & Pan, Y. (2024). Structural entropy minimization combining graph representation for money laundering identification. *International Journal of Machine Learning and Cybernetics*, 15(9), 3951–3968.

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. 10.

Wójcik, F. (2024). An Analysis of Novel Money Laundering Data Using Heterogeneous Graph Isomorphism Networks. FinCEN Files Case Study. *Econometrics*, 28(2), 32–49.

Xu, H., Yu, K., Wei, M., & Zhu, Y. (2024). Intelligent Anti-Money Laundering Transaction Pattern Recognition System Based on Graph Neural Networks. 2(1), 93–108.

Yu, Q., Ke, Z., Xiong, G., Cheng, Y., & Guo, X. (2024). Identifying Money Laundering Risks in Digital Asset Transactions Based on AI Algorithms. *2024 4th International Conference on Electronic Information Engineering and Computer Communication, EIECC 2024*, 1081–1085.

Zade, N. P. (2024). Exploring Graph-Based Machine Learning Techniques for Transaction Fraud Detection: A Comparative Analysis of Performance.

Zhang, H. (2025). Research on the Application of Knowledge Graphs in Bank Risk Management. 0, 62–69.

Zhang, Q., Zhu, Y., Zhang, R., Chen, R., & Lan, T. (2025). Research on Anti-Money Laundering Technology Based on Graph Attention Mechanism. 13511, 1–11.