ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Fonte: Imagem gerada por inteligência artificial através do serviço ArtGuru.ai (https://www.artguru.ai).

Bitcoin Anomaly Detection (BAD) - Use of Machine Learning for Fraudulent Transaction Detection

NUNO GONÇALO RODRIGUES CABRAL GOMES

(Bacharel)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientador: Prof. Doutor Artur Jorge Ferreira

Júri:

Presidente: Prof. Doutor José Manuel de Campos Lages Garcia Simão

Vogais: Prof. Doutor Pedro Miguel Torres Mendes Jorge

Prof. Doutor Artur Jorge Ferreira

July 2025

Bitcoin Anomaly Detection (BAD) - Use of Machine Learning for Fraudulent Transaction Detection

NUNO GONÇALO RODRIGUES CABRAL GOMES

(Bacharel)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientador: Prof. Doutor Artur Jorge Ferreira, ISEL

Júri:

Presidente:	Prof. Doutor José Manuel de Campos Lages Garcia Simão, ISEL
Vogais:	Prof. Doutor Pedro Miguel Torres Mendes Jorge, ISEL
	Prof. Doutor Artur Jorge Ferreira, ISEL

July 2025

To my father and to the memory of my dear mother—
whose unwavering belief in me and steadfast support of this
new academic journey
have been a constant source of strength.
Thank you for your enduring encouragement, which has
carried me through to the end.

Acknowledgements

Acknowledgements

I would like to express my sincere gratitude to all those who supported and encouraged me throughout the course of this journey.

To my wife and children, I owe my deepest appreciation for their patience, understanding, and unwavering support during the countless hours devoted to study, research, and writing. Their presence and encouragement were the very reason I chose to return to academic life after more than two decades.

A heartfelt acknowledgment is due to Professor Dr Artur Jorge Ferreira, whose professionalism, commitment, motivation, and generosity—both within and beyond the boundaries of his role—have been a constant source of inspiration. His guidance and belief in me made a significant difference.

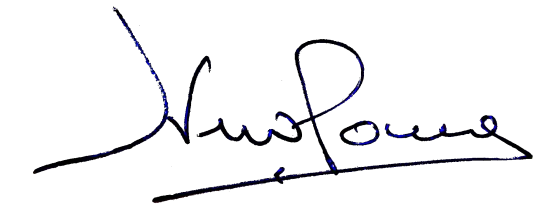
I am deeply thankful to Dr Lígia Cardiga for her availability, her willingness to explain and clarify aspects of the English language, and for her invaluable support in reviewing and refining the final text into proper British English.

I am also grateful to my professors, whose dedication to teaching, guidance, and the sharing of academic knowledge were crucial to my learning experience.

To my work colleagues, thank you for your understanding and willingness to accommodate my academic commitments, allowing me the time and space to focus on this thesis. A special word of thanks goes to Onur Güzelmeriç for his exceptional support.

I declare that this **dissertation** is the result of my personal and independent research. Its content is original, and all sources listed in the bibliographic references were consulted and are duly mentioned in the text. I further declare that all scientific and technical references relevant to the development of the work are duly cited and included in the bibliographic references.

The author



Lisbon, June , 2025

Abstract

The increasing adoption of cryptocurrencies, especially Bitcoin, has significantly altered the financial landscape, enabling decentralised and pseudonymous transactions. While these characteristics foster innovation, they also present serious challenges in detecting fraudulent behaviour, including money laundering and investment scams. This dissertation introduces Bitcoin Anomaly Detection (BAD), a hybrid machine learning framework designed to detect anomalous transactions on the Bitcoin blockchain.

The methodology integrates supervised and unsupervised learning techniques, applied to the Elliptic dataset comprising over 200,000 transactions with temporal and graph-based features. Feature engineering, class imbalance handling (e.g., SMOTE + ENN), and dimensionality reduction (PCA, UMAP) are employed. Several models are evaluated, including XGBoost, RF, and GNN, achieving up to 97.5% accuracy, 96.2% recall, and a false positive rate below 4.5%.

Graph analysis revealed that illicit transactions tend to form sparsely connected "sink" nodes—receiving many inputs (high in-degree) but sending no outputs (zero out-degree)—a pattern typical of laundering. Semi-supervised learning and association rule mining were used to label unknown data and enhance classification reliability.

The final pipeline combines feature selection, class instance sampling, and a hybrid semi-supervised learning approach—bridging supervised and unsupervised methods—to classify transactions with high accuracy and robustness.

Key challenges addressed include the scarcity of labelled data, severe class imbalance, and the evolving nature of fraud techniques. XAI components were incorporated to ensure interpretability and compliance with regulatory frameworks such as Markets in Crypto-Assets (MiCA) and Financial Action Task Force (FATF).

The findings from our experimental evaluation demonstrate the viability of adaptive, interpretable AI solutions for safeguarding decentralised financial ecosystems and supporting efforts in Anti-Money Laundering (AML) and law enforcement.

Keywords: Anomaly detection, bitcoin, blockchain, fraud detection, GNN, graph analysis, machine learning, MiCA, XAI

Resumo

A crescente adoção de criptomoedas, especialmente o Bitcoin, alterou significativamente o panorama financeiro, permitindo transações descentralizadas e pseudónimas. Embora estas características promovam a inovação, também criam desafios substanciais na deteção de comportamentos fraudulentos, incluindo branqueamento de capitais e esquemas de investimento ilícitos. Esta dissertação apresenta o Bitcoin Anomaly Detection (BAD), uma framework híbrida de aprendizagem automática desenvolvida para detetar transações anómalas na blockchain do Bitcoin.

A metodologia integra técnicas de aprendizagem supervisionada e não supervisionada, aplicadas ao conjunto de dados Elliptic, com mais de 200,000 transações com caraterísticas temporais e baseados em grafos. São utilizadas técnicas de engenharia de caraterísticas, reequilíbrio de classes (ex. SMOTE + ENN) e redução de dimensionalidade (PCA, UMAP). Foram avaliados vários modelos, incluindo XGBoost, RF e GNN, alcançando até 97,5% de precisão, 96,2% de *recall* e uma taxa de falsos positivos inferior a 4,5%.

A análise de rede revelou que as transações ilícitas tendem a formar nós com ligações escassas, com elevado grau de entrada e nenhum de saída, indicando padrões de branqueamento. Técnicas de aprendizagem semi-supervisionada e mineração de regras de associação foram aplicadas para rotular dados desconhecidos e reforçar a fiabilidade da classificação.

O *pipeline* final combina a seleção de caraterísticas, a amostragem de instâncias de classe e uma abordagem híbrida de aprendizagem semi-supervisionada —que combina métodos supervisionados e não supervisionados— para classificar transacções com elevada precisão e robustez.

Os principais desafios abordados incluem a escassez de dados etiquetados, o forte desbalanceamento entre classes e a natureza evolutiva dos esquemas de fraude. Foram ainda incorporadas técnicas de Inteligência Artificial Explicável (XAI), garantindo a interpretabilidade e conformidade com os quadros regulamentares, tais como MiCA e o FATF.

Os resultados da nossa avaliação experimental demonstram a viabilidade de soluções de IA adaptáveis e interpretáveis para salvaguardar os ecossistemasa financeiros descentralizados, apoiando iniciativas de prevenção de branqueamento de capitais e investigação criminal.

Palavras-chave: Deteção de anomalias, bitcoin, blockchain, deteção de fraude, GNN, análise de grafos, aprendizagem automática, MiCA, XAI

Contents

Annexes

List of Figures

List of Tables

Glossary

Anti-Money Laundering (AML)	Laws, regulations, and procedures aimed at preventing the conversion of illegally obtained funds into legitimate income. 1
Bitcoin (BTC)	The first and most widely recognized cryptocurrency, introduced in 2008 through a white paper by Satoshi Nakamoto. It operates on a decentralized network using blockchain technology. 1
Blockchain	A decentralized and distributed digital ledger that securely records transactions across a network of computers. It is the foundational technology behind cryptocurrencies and has applications across various industries. 1, 2, 3, 4
Cross-Chain Transactions	Transactions that occur between different blockchain networks, requiring standard protocols to ensure security and interoperability. 19
Cryptocurrency	A digital or virtual currency that uses cryptographic techniques to secure transactions, control the creation of new units, and verify asset transfers. Unlike traditional currencies, cryptocurrencies operate on decentralized networks based on blockchain technology. 1
Distributed Ledger Technology (DLT)	A digital system for recording asset transactions in multiple locations simultaneously. It forms the basis for blockchain and other decentralized data architectures. 1
Enhanced Due Diligence (EDD)	A detailed risk assessment process used by financial institutions for high-risk clients or transactions. It involves a deeper investigation into customer identity, background, and source of funds. 18
Gross Domestic Product (GDP)	The total monetary value of all goods and services produced across all countries in a given year. It serves as a measure of global economic activity and output. 1
Local Interpretable Model-Agnostic Explanations (LIME)	Local Interpretable Model-Agnostic Explanations. A technique that explains individual predictions of machine learning models by approximating them locally with an interpretable model. 16

Ponzi Scheme	A fraudulent investment operation where returns are paid to earlier investors using funds from newer investors. Such schemes are often uncovered through blockchain analysis or machine learning. 10
Proof of Work (PoW)	A consensus mechanism used in blockchain systems where miners solve complex cryptographic puzzles to validate transactions and generate new blocks. 2
Quantum-Resistant Cryptography	Encryption techniques designed to withstand decryption by quantum computers, ensuring long-term security of digital assets and systems. 16
Ransomware	A form of malicious software that encrypts a victim's data and demands payment, often in cryptocurrency, to release it. It poses a major cybersecurity threat. 19
Regulatory Fragmentation	The inconsistency in financial and crypto regulations across different jurisdictions, which complicates compliance and enforcement efforts. 1
SHapley Additive exPlanations (SHAP)	SHapley Additive exPlanations. A method for explaining machine learning model predictions using Shapley values from cooperative game theory. It enables both global and local interpretability. 16, 53
Shor's Algorithm	A quantum algorithm developed by Peter Shor for factoring large integers. Its efficiency poses a threat to traditional cryptographic systems. 17
Zero-Knowledge Proofs	A cryptographic method that allows one party to prove the truth of a statement to another without revealing any additional information. Commonly used in privacy-focused blockchain applications. 16

Acronyms

AI	Artificial Intelligence 7, 14, 18, 19, 86
AML	Anti-Money Laundering xi, 12, 13, 14, 18, 19, 86
ANN	Artificial Neural Networks 11
ANOVA	Analysis of Variance 53, 76
API	Application Programming Interface 18
ARI	Adjusted Rand Index 27, 39, 42, 43, 73, 74, 76, 77, 78, 81
ASXGB	Adaptive Stacked eXtreme Gradient Boosting 14
AUC	Area Under the ROC Curve 29, 54, 55, 56
BAD	Bitcoin Anomaly Detection xi, 2, 4, 21, 85, 86
BERT	Bidirectional Encoder Representations from Transformers 13
BSA	Bank Secrecy Act 18
BTC	Bitcoin 1, 2, 3, 4, 5, 6, 10, 12, 13, 14, 19, 22
CH	Calinski-Harabasz 27, 39, 42, 73, 74, 77, 78, 81
CNN	Convolutional Neural Networks 10
DB	Davies-Bouldin 27, 39, 42, 43, 73, 74, 77, 78, 81
DBSCAN	Density-Based Spatial Clustering of Applications with Noise 6, 27, 39, 40, 41, 71, 78, 79
DNN	Deep Neural Networks 10, 14
DR	Dimensionality Reduction 25, 60
DT	Decision Tree 27, 28
ECC	Elliptic Curve Cryptography 17
EDA	Exploratory Data Analysis xix, 21, 24
EM	Expectation-Maximization 40
ENN	Edited Nearest Neighbors xi, xiii, 26, 37, 38, 61, 64, 66, 67
EU	European Union 1, 18
EWC	Elastic Weight Consolidations 11
FATF	Financial Action Task Force xi, xiii, 5, 7, 18, 85

FinCEN	Financial Crimes Enforcement Network 7, 18
FP-Growth	Frequent Pattern Growth 45, 46
FPR	False Positive Rates 4, 5, 85
FR	Feature Reduction 25, 53
FS	Feature Selection 25, 26, 27, 35, 36, 53, 56
GCN	Graph Convolutional Networks 13, 14, 29
GDPR	General Data Protection Regulation 18
GMM	Gaussian Mixture Models xxi, 27, 39, 47, 71, 73, 78, 79, 80, 81, 82
GNN	Graph Neural Networks xi, xiii, 9, 13, 14, 15, 29
GPT	Generative Pre-trained Transformer 13
GPU	Graphics Processing Unit 14
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise 6, 27, 39, 41, 71
IC3	FBI's Internet Crime Complaint Center 3
KNN	K-Nearest Neighbours 28, 53
LightGBM	Light Gradient Boosting Machine 28, 65
MiCA	Markets in Crypto-Assets xi, xiii, 1, 5, 7, 11, 17, 18, 85
ML	Machine Learning 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 17, 24, 34, 54, 59
MLP	Multilayer Perceptron 28
MSB	Money Service Business 18
NB	Naïve Bayes 28
NMI	Normalized Mutual Information 27, 39, 42, 44, 73, 74, 76, 78, 81, 83
PCA	Principal Component Analysis xi, xiii, xx, 34, 53, 58, 60, 75, 76
PR	Precision-Recall Curves 29
PyOD2	Python Outlier Detection 2 9
RF	Random Forest xi, xiii, 10, 27, 28, 78
RNN	Recurrent Neural Networks 10, 15
ROC	Receiver Operating Characteristic xix, 29, 53, 54, 55
RSA	Rivest-Shamir-Adleman 17

SMOTE	Synthetic Minority Over-sampling Technique xi, xiii, 26, 37, 38, 61, 63, 64, 66, 67
SVM	Support Vector Machines 10, 12, 28, 47, 79, 80, 83
TPU	Tensor Processing Unit 14
t-SNE	t-Distributed Stochastic Neighbor Embedding 75
UMAP	Uniform Manifold Approximation and Projection xi, xiii, 34, 53, 59, 75, 82
VASP	Virtual Asset Service Providers 18
XAI	eXplainable Artificial Intelligence xi, xiii, 16, 18, 85
XGBoost	eXtreme Gradient Boosting xi, xiii, 10, 13, 27, 28, 34, 35, 47, 53, 54, 55, 56, 60, 63, 65, 66, 67, 68, 69, 71, 72, 73, 74, 79, 81, 82, 85

1 Introduction

The emergence of cryptocurrencies, particularly Bitcoin (BTC) in 2008, marked a transformative moment in the global financial landscape. Based on distributed ledger technology (Blockchain), BTC allows rapid, decentralized, and secure transactions, removing intermediaries and enabling global payments with reduced fees. As of January 2025, the market capitalization of cryptocurrency exceeded $3.64 trillion, with BTC accounting for approximately 55.53% of this value ($2.02 trillion) [22] highlighting its significant presence in the financial sector. This substantial market presence has attracted both legitimate users and malicious actors, creating an urgent need for robust security measures.

The evolution of Bitcoin and other cryptocurrencies has both garnered significant attention and posed a threat to the very foundations of the financial system [67]. The decentralised nature of BTC, whilst innovative, presents unique challenges for security, privacy [23] and fraud prevention. The pseudo-anonymous characteristics of transactions have enabled various illicit activities, including money laundering and financial fraud. Money laundering alone impacts between 2% and 5% of global Gross Domestic Product (GDP) [75], prompting the implementation of rigorous Anti-Money Laundering (AML) frameworks. These frameworks encompass customer identification, transaction monitoring, and suspicious activity reporting. However, their effectiveness faces limitations in cryptocurrency contexts due to traceability challenges and regulatory fragmentation.

The European Union (EU) has implemented comprehensive regulation through the MiCA framework. This legislation introduces stringent rules to enhance transparency, protect consumers, and prevent financial crimes. The framework includes mechanisms for tracking crypto-asset transfers and blocking suspicious transactions, thereby strengthening market integrity [29].

Despite these regulatory advances, detecting fraud in BTC transactions remains challenging due to the complex nature of cryptocurrency transactions and the evolving sophistication of illicit activities. Unlike traditional financial systems with known identities, BTC transactions occur between cryptographic addresses without revealing personal information [59]. This poses significant challenges in identifying fraudulent activities, as malicious transactions may appear legitimate whilst concealing illicit behaviour [78]. Consequently,

innovative solutions combining advanced data analysis techniques and Machine Learning (ML) are essential to address these issues effectively.

Against this backdrop, this chapter is structured as follows:

Section 1.1 overviews Bitcoin transactions, focusing on their cryptographic structure and the role of blockchain in ensuring security and transparency. It also notes blockchain's dual role in enabling both financial innovation and illicit activities like money laundering and fraud.

Section 1.2 highlights key challenges in cryptocurrency fraud detection, such as pseudo-anonymous transactions, scalability issues, evolving tactics, high false positives, and limited labelled data—factors that impede effective model development.

Section 1.3 introduces the primary goals of the study, which include developing a hybrid BAD framework, achieving high detection accuracy, and ensuring regulatory compliance.

Section 1.4 outlines the study's main contributions: novel feature engineering, combining traditional and deep learning models, and new approaches to lower false positive rates.

Section 1.5 outlines the organisation of the thesis.

1.1 Context and Motivation

A Bitcoin transaction involves the transfer of some value between Bitcoin wallets, recorded on the blockchain. Each wallet contains a cryptographic key pair: a public key, serving as an address and a private key authorising transactions. Transaction verification occurs through a consensus mechanism called Proof of Work (PoW), where miners validate transactions by solving complex cryptographic puzzles [9].

The blockchain serves as a public ledger that chronologically records Bitcoin transactions. Each block contains transaction data, a timestamp, and a reference to the previous block, creating an immutable chain. This structure ensures data integrity, as altering information would require re-mining all subsequent blocks—a computationally infeasible task [80]. Its public nature allows transaction verification without central oversight.

To visually represent the dual nature of blockchain technology, Figure 1.1 illustrates both the positive and negative applications of blockchain in the financial sector.

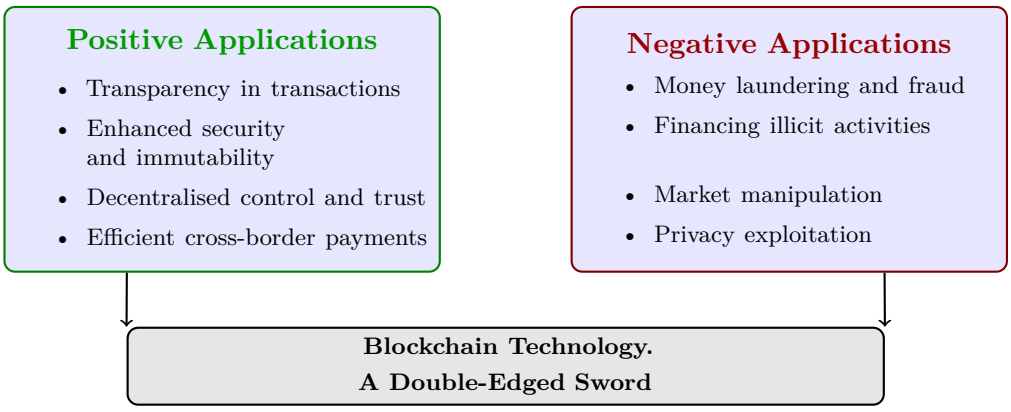


Figure 1.1: *The Dual Nature of Blockchain Technology: Positive and Negative Applications*

However, these blockchain attributes allow investigators to leverage tools like transaction graph analysis to identify illicit patterns, thus highlighting the dual-edged nature of this technology.

Recent reports highlight the scale of cryptocurrency-related fraud, namely:

- The FBI's Internet Crime Complaint Center (IC3) reported cryptocurrency-related losses exceeding $5.6 billion in 2023, with investment scams accounting for 71% of all reported incidents [30]. Losses from cryptocurrency-related investment fraud schemes reported to the IC3 rose from $2.57 billion in 2022 to $3.96 billion in 2023, an increase of 53%.

- Chainalysis Team [17] report provided an updated estimation that the volume of crypto-based illicit transactions has risen for three consecutive years, reaching an all-time high of $39.8 billion in 2022, up from $23.2 billion in 2021 and $9.4 billion in 2020. Whilst overall crypto-crime revenues declined in 2023, showing a significant drop in the value received by illicit cryptocurrency addresses to a total of $24.2 billion, cryptocurrency-based crime remains a significant concern.

- Research from Harvard Journal of Law and Technology [62] shows that market manipulation and fraud schemes have evolved to exploit regulatory gaps.

According to the Coin Dance portal [21], the legal status of cryptocurrencies varies significantly worldwide. As of 2024, Bitcoin operates legally in 132 countries, though regulatory approaches differ substantially. Two nations—El Salvador and the Central African Republic—have recognised Bitcoin as legal tender. Conversely, several countries have explicitly prohibited cryptocurrency trading and usage within their jurisdictions.

Primary factors driving cryptocurrency prohibition include:

- Concerns about monetary policy control and financial system stability.

- Risks associated with anonymous transactions enabling illegal activities.

- Government reluctance to cede control over monetary systems.

Figure 1.2 provides a comprehensive view of the presence of Bitcoin on the world map.

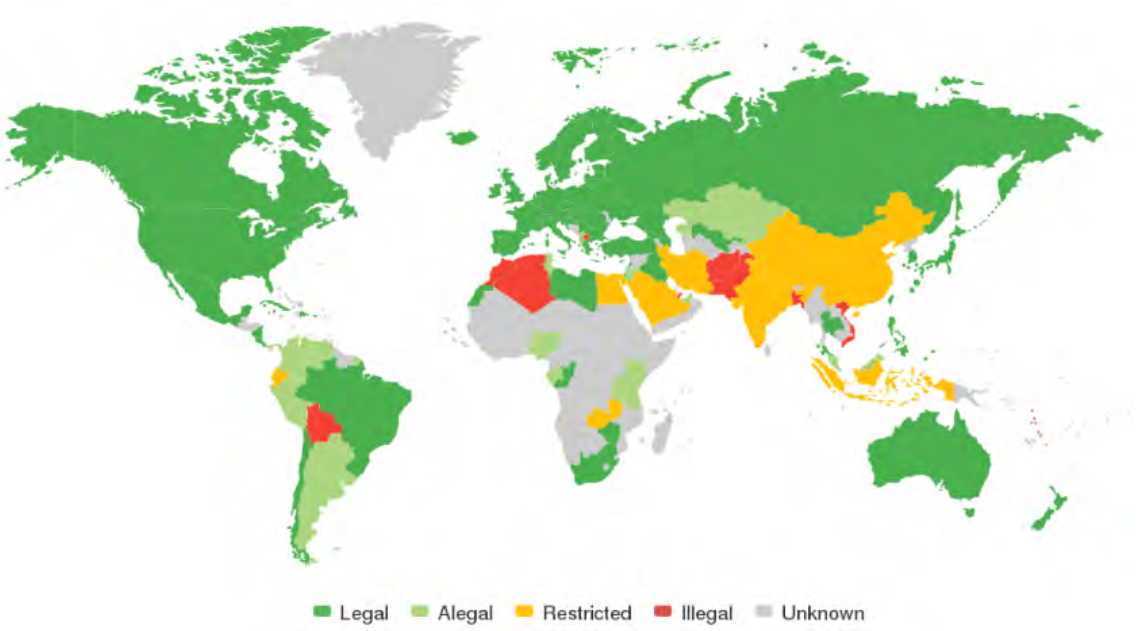


Figure 1.2: *Global Bitcoin Legality (source: Coin.Dance, accessed on 23–6–2025).*

Therefore, detecting anomalies in Bitcoin transactions is essential to keep the integrity of this emerging financial system.

1.2 Problem Statement

The cryptocurrency ecosystem faces several critical challenges:

1. **Detection Complexity** — The pseudo-anonymous nature of blockchain transactions requires sophisticated methods to identify fraudulent patterns whilst preserving user privacy [56].
2. **Scale and Speed** — The high volume and velocity of cryptocurrency transactions demand real-time detection capabilities processing millions of transactions efficiently.
3. **Evolving Threats** — Fraudulent techniques continuously adapt to evade detection, requiring equally adaptive security measures [23].
4. **False Positive Rates (FPR)** — Contemporary detection systems often face challenges with elevated False Positive Rates (FPR), straining investigative resources [73].
5. **Data Limitations** — The scarcity of accurately labelled datasets thwarts the development of robust detection models [52].

1.3 Research Objectives

This research aims to develop an innovative hybrid BAD framework, integrating supervised and unsupervised ML techniques. The framework focuses on identifying anomalous transactions whilst maintaining high accuracy and low false positive rates.

Primary objectives include:

1. Achieving detection accuracy of $\geq 95\%$ on benchmark datasets.
2. Reducing the FPR below 5% with $\geq 95\%$ recall.
3. Ensuring model interpretability for compliance with MiCA and FATF regulations [29, 31].

Table 1.1 presents the refined performance metrics for each objective.

Table 1.1: Research Objectives and Corresponding Performance Metrics.

Parameter	**Metric**	**Target Value**
Detection Accuracy	Precision/Recall	$\geq 95\%$
False Positive Rate	FPR	$\leq 5\%$
Model Interpretability	Compliance Score	100% *(MiCA, FATF)*

1.4 Research Contributions

This thesis aims to advance the field through the following contributions:

1. Development of novel feature engineering techniques specifically designed for Bitcoin transaction data.
2. Creation of a hybrid detection framework combining traditional ML with deep learning approaches.
3. Implementation of innovative methods for reducing false positive rates whilst maintaining high detection accuracy.
4. Empirical validation of detection effectiveness across various types of anomalies or suspicious activities.
5. Development of interpretable models that align with regulatory requirements.

From the work reported in this thesis, the following papers have been written:

Nuno Gomes and Artur Ferreira, “Bitcoin Fraud Detection: A Study with Dimensionality Reduction and Machine Learning Techniques”, 14th International Conference on Data Science, Technology and Applications, DATA 2025, June 2025, Bilbao, Spain, pages 716—723, ISBN 978—989—758—758—0, ISSN 2184—285X

Nuno Gomes and Artur Ferreira, “Bitcoin Anomaly Detection (BAD): A Hybrid Machine Learning Framework”, MDPI — Information Journal, submitted for publication

The code developed from the experimental part of this thesis is publicly available at https://github.com/NGGomes.

1.5 Thesis Structure

The remainder of this Thesis is organised as shown in Figure 1.3.

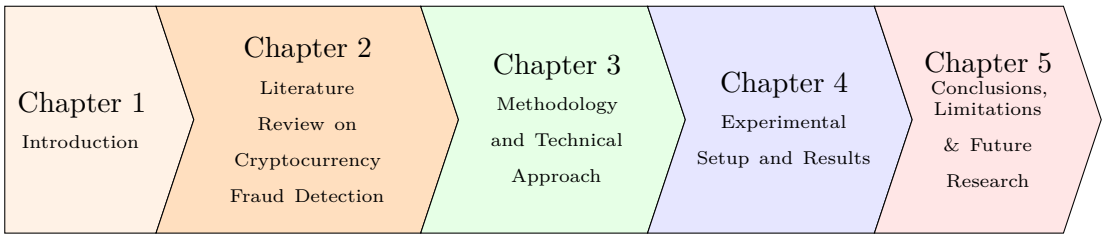


Figure 1.3: *Structure of the Thesis.*

- Chapter 2 provides a comprehensive overview of the literature on cryptocurrency fraud detection methods, focusing on recent advancements in ML applications.
- Chapter 3 outlines the proposed methodology and technical approach, clarifying the next steps to accomplish the final thesis goals. Figure 1.4 illustrates the procedural block diagram to be used with Bitcoin datasets.

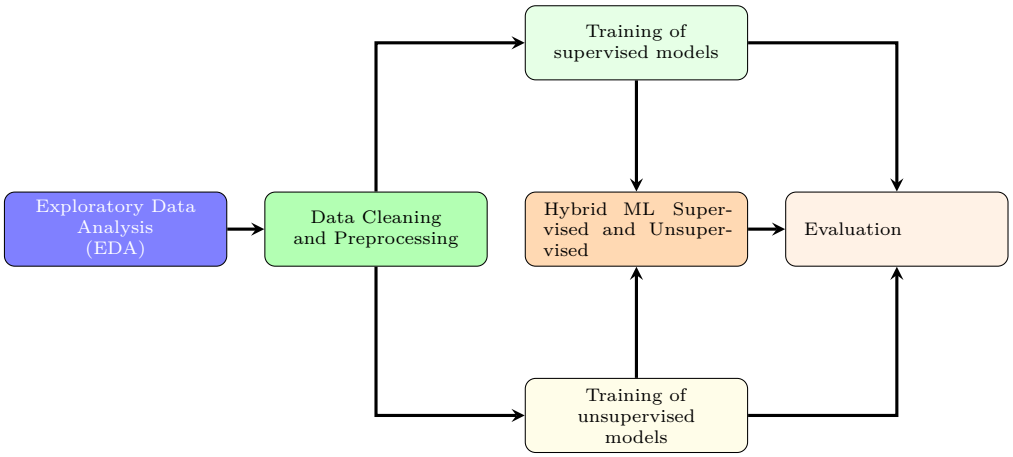


Figure 1.4: *Procedural block diagram to be used with Bitcoin datasets.*

- Chapter 4 presents the experimental setup, implementation details, and results of the proposed approach.
- Chapter 5 concludes the work, discusses limitations, and suggests directions for future research.
- Annex 1 details additional experiments with Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithms, which were excluded from the main evaluation due to poor performance metrics and limited coverage of the dataset.
- Annex 2 presents complementary classification experiments using Gaussian-based detection systems, exploring alternative semi-supervised learning approaches for transaction classification.

2 Literature Review

Chapter 2 provides a comprehensive overview of cryptocurrency fraud detection, exploring the evolution from simple rule-based approaches to sophisticated data-driven and AI-based solutions. It examines technological advancements, regulatory frameworks, and future research directions in this rapidly developing field.

The chapter is structured as follows:

Section 2.1 examines the evolution of fraud detection strategies in cryptocurrency, tracing the progression from heuristic-based rule systems to modern ML and graph analytics approaches.

Section 2.2 explores fundamental techniques in cryptocurrency fraud detection, including feature engineering for blockchain data, machine learning approaches, supervised versus unsupervised learning, and regenerative learning methods.

Section 2.3 analyzes the historical development of Bitcoin fraud detection methods from 2009 to the present, highlighting key milestones and technological advancements across different periods.

Section 2.4 investigates the current state of the art and advanced techniques, addressing trade-offs between accuracy, scalability, and interpretability in modern fraud detection systems, as well as emerging technologies like quantum-resistant cryptography.

Section 2.5 discusses the regulatory landscape and compliance requirements affecting cryptocurrency fraud detection, including MiCA regulation, FATF guidelines, and Financial Crimes Enforcement Network (FinCEN) standards.

Section 2.6 identifies research gaps and future directions, exploring challenges in balancing scalability with accuracy, adaptability to new fraud patterns, privacy preservation, and the development of explainable Artificial Intelligence (AI) solutions for regulatory compliance.

By examining these critical aspects of cryptocurrency fraud detection, this chapter aims to establish a solid foundation for the research presented in subsequent chapters, while highlighting opportunities for innovation and advancement in the field.

2.1 Evolution of Cryptocurrency Fraud Detection Approaches

The development of fraud detection in cryptocurrency has followed a clear progression that mirrors the evolution of the technology itself, transitioning from simple rule-based systems to sophisticated machine learning and deep learning architectures. Figure 2.1 outlines key milestones in this evolutionary journey.

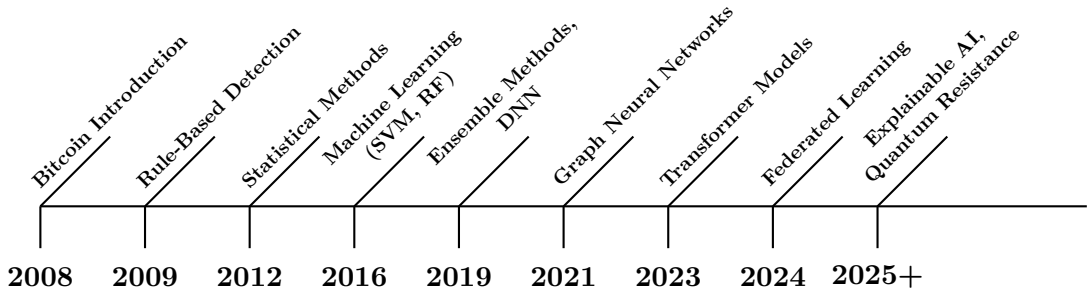


Figure 2.1: *Timeline Evolution of Cryptocurrency Fraud Detection Methods.*

Early cryptocurrency fraud detection relied on manual rules and heuristics (e.g., blacklisting suspicious addresses), but these struggled with the dynamic nature of decentralized fraud.

As the cryptocurrency ecosystem expanded, data-driven approaches gained prominence. Supervised ML techniques were widely adopted, utilising labelled datasets to identify patterns indicative of fraud. However, they faced limitations with scarce labeled data and rapidly evolving threats.

The emergence of unsupervised and semi-supervised methods marked a significant advancement in anomaly detection [36]. Algorithms such as clustering, anomaly detection, and one-class classification were applied to uncover unusual patterns in transaction data, even in the absence of explicit labels. These techniques proved particularly effective in addressing the pseudo-anonymity and decentralised nature of blockchain transactions [23].

Table 2.1 summarizes the advantages and shortcomings of each evolutionary stage in cryptocurrency fraud detection methodology.

Table 2.1: Advantages and Shortcomings of Evolutionary Methodology.

Year	Methodology	Advantages	Shortcomings
2009	Rule-Based Detection	Simple, easy to implement	High false positive rate, lacks adaptability
2012	Statistical Methods	Identifies basic transaction patterns	Limited scalability, struggles with new fraud techniques
2016	Machine Learning (SVM, RF)	More accurate fraud detection	Requires labeled data, interpretability issues
2019	Ensemble Methods, Deep Neural Networks (DNN)	Improved accuracy, better feature extraction	Computationally expensive, requires fine-tuning
2021	GNN	Captures transaction relationships effectively	Low interpretability, requires large datasets
2023	Transformer Models	Handles sequential transaction data efficiently	High resource demand, potential overfitting
2024+	Federated Learning, XAI	Enhances privacy, improves transparency	Implementation complexity, regulatory challenges

The evolution of cryptocurrency fraud detection has increasingly focused on countering obfuscation techniques, such as cryptocurrency mixers and chain hopping [10]. More recently, hybrid approaches that combine supervised, unsupervised, and network-based techniques have emerged. These methods incorporate graph analytics, such as Graph Neural Networks (GNN), to analyse relationships between blockchain entities.

Recent advancements in statistical pattern detection and graph-based clustering—as demonstrated in studies by Shojaeinasab [70]—highlight the importance of integrating data-driven approaches with domain-specific knowledge. When combined with deep learning frameworks, these methods provide a robust mechanism for deanonymising transactions and detecting illicit activity.

Additionally, tools like Python Outlier Detection 2 (PyOD2) [19] have assisted the application of advanced anomaly detection frameworks, incorporating deep learning and ensemble methods to enhance detection capabilities.

2.2 Techniques for Cryptocurrency Fraud Detection

This section explores the core techniques that have been developed and applied to the challenge of detecting fraud in cryptocurrency transactions, with particular attention to their theoretical foundations, practical implementations, and relative strengths and limitations.

2.2.1 Feature Engineering for Blockchain Data

Feature engineering is crucial for Bitcoin fraud detection as it enhances ML models by transforming raw blockchain data into meaningful insights. Effective feature engineering extracts patterns that distinguish legitimate from fraudulent transaction behavior, forming the foundation for all subsequent analysis methods.

Key feature engineering approaches include:

- Graph-based features — Extract information from transaction networks, such as node degree, centrality measures, and community structures — to identify unusual transaction patterns or relationships.
- Statistical features — Generate summary statistics like transaction frequency, volume, time intervals, and value distributions to detect outliers from normal behavior.
- Temporal patterns — Capture time-based characteristics like sudden activity spikes, periodic behaviors, or unusual transaction timing that may indicate coordinated fraudulent activity.
- Address-based features — Analyze address characteristics including lifetime, activity patterns, and balance changes to identify suspicious entities.

By selecting and engineering relevant features, models can improve accuracy, reduce false positives, and adapt to evolving fraud tactics, making feature engineering a vital step in developing robust detection systems.

2.2.2 Machine Learning Approaches

ML techniques can identify anomalous patterns within large datasets, flagging behaviours that may indicate fraud. Classification and clustering algorithms, including decision trees, neural networks, and Support Vector Machines (SVM), are commonly used to identify fraudulent transaction patterns. Below is a breakdown of some key algorithms and their effectiveness in detecting fraud:

- Decision Trees [64] are a popular supervised learning algorithm that splits data into branches based on feature values to make predictions. They are effective in identifying simple fraud patterns due to their interpretability and ease of implementation. For example, decision trees can be used to flag transactions with unusually high amounts or frequent transfers between specific addresses. However, they may struggle with more complex fraud patterns that require capturing non-linear relationships.

- SVM [76] are supervised learning models that classify data by finding the optimal hyperplane that separates different classes. They are particularly effective in high-dimensional spaces, making them suitable for detecting fraud in complex transaction datasets. For instance, SVM got the highest accuracy of 98.2% in finding abnormalities in the Bitcoin transaction network [71].

- Neural networks [35], especially Deep Neural Networks (DNN), excel at capturing complex, non-linear relationships in data. They are highly effective in fraud detection due to their ability to model intricate patterns in transaction sequences. For example, Recurrent Neural Networks (RNN) have been used to analyse temporal patterns in Bitcoin transactions, while Convolutional Neural Networks (CNN) have been applied to detect anomalies in transaction graphs. These models have shown high accuracy but often require large amounts of labelled data and computational resources [4, 74].

- Ensemble methods [34], such as Random Forest (RF) [14] and Gradient Boosting e.g., eXtreme Gradient Boosting (XGBoost) [20], combine multiple models to improve predictive performance. These methods are robust against overfitting and can handle imbalanced datasets, which are common in fraud detection. For example, RF with cost-sensitive approach achieved 0.969 recall, classifying 31 out of 32 Ponzi schemes [12] in Bitcoin transactions, while XGBoost has been used to achieve high acurracy (F1-score of 0.82) in the detection of anomalies in cryptocurrency transactions [24].

2.2.3 Supervised vs. Unsupervised Learning in Fraud Detection

The choice between supervised and unsupervised learning approaches represents a fundamental decision in cryptocurrency fraud detection system design, with each offering distinct advantages for different aspects of the problem:

Supervised Learning Approaches

- Rely on labeled data where transactions are already classified as legitimate or fraudulent.
- Provide high accuracy when sufficient labeled data is available.
- Enable precise classification based on known fraud patterns.
- Examples include SVM, Decision Trees, Random Forests, and Deep Neural Networks.

Unsupervised Learning Approaches

- Do not require labeled data, operating instead by identifying unusual patterns or outliers.
- Particularly valuable in cryptocurrency contexts due to the scarcity of labeled fraud data.
- Can detect novel fraud patterns not present in training data.
- Examples include clustering algorithms (K-means, DBSCAN), isolation forests, and autoencoders.

Rationale for Using Both Approaches

- Complementary strengths — Supervised methods excel at detecting known fraud patterns with high precision, while unsupervised methods can identify novel or evolving fraud techniques.
- Data limitations — The cryptocurrency ecosystem has limited labeled fraud data, necessitating unsupervised approaches while leveraging available labeled data where possible.
- Confidence scoring — Combining both approaches allows for assigning confidence scores to fraud classifications, with high agreement between methods increasing confidence.
- Regulatory compliance — The dual-method approach provides multiple layers of verification, supporting the stringent evidence requirements of regulatory frameworks like MiCA.

2.2.4 Regenerative and Continuous Learning

Regenerative learning, also known as continuous learning, allows ML models to update dynamically as new data becomes available. This capability is particularly valuable in cryptocurrency fraud detection, where evolving fraud patterns necessitate adaptive detection mechanisms.

Regenerative learning relies on models that continuously refine their fraud detection abilities. Techniques such as Artificial Neural Networks (ANN) and Elastic Weight Consolidations (EWC) help maintain previously learned knowledge while integrating new

transactional patterns. This approach is effective in mitigating concept drift, ensuring that fraud detection systems remain robust over time.

A significant advantage of regenerative learning is its adaptability. Unlike static models, regenerative systems retain historical knowledge while simultaneously learning from new patterns. However, challenges such as computational costs and overfitting remain key considerations in the implementation of regenerative fraud detection models.

2.3 Bitcoin Fraud Detection Methods Evolution

The evolution of Bitcoin fraud detection has mirrored both the growing sophistication of cryptocurrency transactions and corresponding advances in analytical techniques. From Bitcoin's inception in 2009, detection methods have progressed through three distinct phases: early rule-based systems (2009—2015), the machine learning revolution (2016—2019), and the current deep learning era (2020—present). This progression reflects the escalating arms race between fraudsters developing increasingly complex obfuscation techniques and security researchers deploying more advanced detection methodologies. The field now stands at an inflection point where emerging technologies like quantum-resistant algorithms and graph neural networks are redefining the boundaries of what's possible in transaction analysis and anomaly detection.

2.3.1 Early Approaches (2009—2015)

Following Bitcoin's introduction, early fraud detection approaches relied heavily on rule-based systems and basic statistical analysis. These methods primarily focused on transaction verification via the blockchain consensus mechanism. Although effective in identifying basic fraud patterns, early detection models lacked adaptability to increasingly sophisticated fraudulent schemes. Ngai et al. [60] discusses the application of data mining techniques in financial fraud detection.

2.3.2 Traditional Machine Learning Era (2016—2019)

The 2016–2019 period marked significant advancements in the application of traditional ML techniques to AML and BTC fraud detection. Here are some relevant approaches:

- Monamo et al. [58] apply unsupervised learning (Trimmed K-means) to detect fraud in Bitcoin transactions.
- Pham and Lee [63] explores anomaly detection in Bitcoin networks using unsupervised learning methods. They resort to a modified version of SVM for unlabelled data, achieving greater consistency in detecting anomalies, with a Dual Evaluation Metric of 0.14415.
- Yin and Vatrapu [82] estimate the proportion of cybercriminal entities in Bitcoin using supervised ML. Three clustering methods—co-spend, intelligence-based, and behaviour-based—were applied to categorize Bitcoin transactions. The models revealed that cybercrime-related entities account for 29.81% (Bagging) and 10.95%

(Gradient Boosting) of the total entities. Additionally, Bagging identified 5.79% of addresses and 10.02% of coins linked to cybercrime.

- Harlev et al. [33] demonstrates de-anonymization of Bitcoin entity types using supervised ML. Their main finding was predicting the type of a yet-unidentified entity using the Gradient Boosting algorithm, achieving an accuracy of 77% and F1-score of approximately 75%.
- Hu et al. [39] detects money laundering on Bitcoin networks with deep walk and node-to-vector techniques outperforming classifiers in binary classification task reaching an average accuracy of 92.29% and an F1-score of 93%.
- Weber et al. [78] experiments with Graph Convolutional Networks (GCN) for anti-money laundering in Bitcoin.
- Zhang and Trubey [83] investigate the use of ML models such as logistic regression, SVM, and artificial neural networks for money laundering detection.

2.3.3 Deep Learning Revolution (2020-Present)

Recent developments have shown improvements through the use of deep learning techniques. We briefly review some of these approches:

- GNN are employed to analyse blockchain transaction networks, enabling the detection of anomalous behaviour with high precision. For instance, GNN have achieved state-of-the-art performance on Elliptic data, attaining an accuracy of 98.99% and an F1-score of 91.75% [5], by effectively capturing complex relationships between blockchain entities.
- Transformer-based models, including architectures like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), have been adapted for fraud detection, significantly enhancing anomaly detection performance. These models excel at handling sequential transaction data and capturing long-range dependencies, making them particularly effective in identifying patterns of fraudulent behaviour [81].
- Hybrid Ensemble Methods, which combine multiple deep learning models, has resulted in notable improvements in both recall and precision, when identifying fraudulent transactions [5, 44].

The following list highlights some key academic studies from this period, showcasing significant advancements in fraud detection techniques and their applications in cryptocurrency.

- Lee et al. [48] introduces ML methods for detecting illegal Bitcoin transactions.
- Lorenz et al. [52] applies active learning techniques with ML to detect money laundering in Bitcoin transactions.
- Jullum et al. [43] employs XGBoost for AML transaction analysis.

- Vassallo et al. [77] enhances fraud detection with Adaptive Stacked eXtreme Gradient Boosting (ASXGB).
- Kim et al. [47] introduces an anomaly detection model based on blockchain traffic monitoring.
- Mohan et al. [57] developed evolving graph convolutions and deep neural decision forests for AML in Bitcoin.
- Ruiz and Angelis [65] evaluate supervised ML algorithms for AML at cryptocurrency exchanges.
- Alarab and Prakoonwit [5] evaluate the effectiveness of GCN for anti-money laundering in Bitcoin and conducts a comparative analysis of supervised learning methods for detecting illicit transactions [4].
- Liu et al. [50] developed a heterogeneous graph transformation network for contract fraud detection.
- Shojaeinasab [70] leverages GNN for Bitcoin transaction analysis, enhancing fraud detection. It incorporates Explainable AI to ensure interpretability and trust in automated AML solutions.

While deep learning models have shown significant success in fraud detection, they also have drawbacks. Two key challenges are their computational cost and potential for overfitting:

- Computational Cost — Deep learning models, especially those with many layers (e.g., DNN, GNN, and Transformers), require substantial computational resources for training and inference. This includes high-performance Graphics Processing Unit (GPU) or Tensor Processing Unit (TPU) and large amounts of memory. The training process can be time-consuming, particularly when dealing with large-scale blockchain datasets, which may limit their use in real-time fraud detection systems.
- Overfitting — Deep learning models are prone to overfitting, especially when trained on limited or imbalanced datasets. Overfitting occurs when a model learns to perform well on the training data but fails to generalize to new, unseen data. This is particularly problematic in fraud detection, where fraudulent transactions are often rare compared to legitimate ones. Techniques such as regularization, dropout, and cross-validation are commonly used to mitigate overfitting, but they add complexity to the model training process.

Table 2.2 summarizes the trade-offs between different deep learning models in terms of computational cost and risk of overfitting. It also shows the key strengths and drawbacks for each technique.

Table 2.2: Trade-offs Between Deep Learning Models in Cryptocurrency Fraud Detection.

Model	Computational Cost	Risk of Overfitting	Key Strengths	Key Shortcomings
GNN	High	Medium	• Captures complex relationships in transaction networks. • Effective for graph-based fraud detection.	• Requires large datasets. • Computationally expensive for large graphs.
Transformer	Very High	High	• Excels at sequential data analysis. • Captures long-range dependencies in transactions.	• Extremely resource-intensive. • Requires significant tuning to avoid overfitting.
Hybrid Ensemble	High	Medium	• Combines strengths of multiple models. • Improves recall and precision.	• Complex to implement. • High computational cost due to multiple model integration.
RNN	Medium	High	• Effective for temporal pattern analysis in transaction sequences.	• Prone to overfitting. • Struggles with very long sequences.

2.4 Current State of the Art and Advanced Techniques

Modern fraud detection techniques incorporate advanced deep learning models, including GNN and Transformer-based architectures. Despite these methods demonstrate high accuracy, their interpretability remains a challenge, particularly in regulatory compliance contexts.

2.4.1 Trade-offs in Modern Fraud Detection Systems

In modern fraud detection systems, there are inherent trade-offs between accuracy, scalability, and interpretability. Understanding these trade-offs is crucial for designing effective systems that meet both technical and regulatory requirements.

- Accuracy vs. Scalability — High-accuracy models, such as GNN and Transformer Models, often require significant computational resources and time to train, especially when applied to large-scale blockchain datasets. On the one hand, GNN achieve high accuracy in detecting fraudulent transactions but are computationally expensive and may struggle to scale to real-time systems with millions of transactions. On the other hand, simpler models like Random Forests or Logistic Regression are more scalable but may sacrifice accuracy, particularly when dealing with complex fraud patterns.
- Accuracy vs. Interpretability — Deep learning models, such as GNN and Transformers, excel at capturing complex patterns in data, leading to high accuracy. However, they are often considered "black boxes" due to their lack of interpretability. This poses challenges for regulatory compliance, where explainability is critical. In contrast, traditional ML models like Decision Trees or Linear Models are more interpretable but may not achieve the same level of accuracy, especially in detecting sophisticated fraud schemes.
- Scalability vs. Interpretability — Scalable models, such as Federated Learning Systems, are designed to handle large datasets efficiently by distributing computation

across multiple devices. On the one hand, these systems often prioritize scalability and privacy over interpretability, making it difficult to trace how decisions are made. On the other hand, interpretable models like Rule-Based Systems are easy to understand and implement but may lack the scalability needed for large-scale fraud detection.

To better understand these inherent trade-offs, Figure 2.2 illustrates the relationship between accuracy, scalability, and interpretability, highlighting the challenges of balancing these competing priorities.

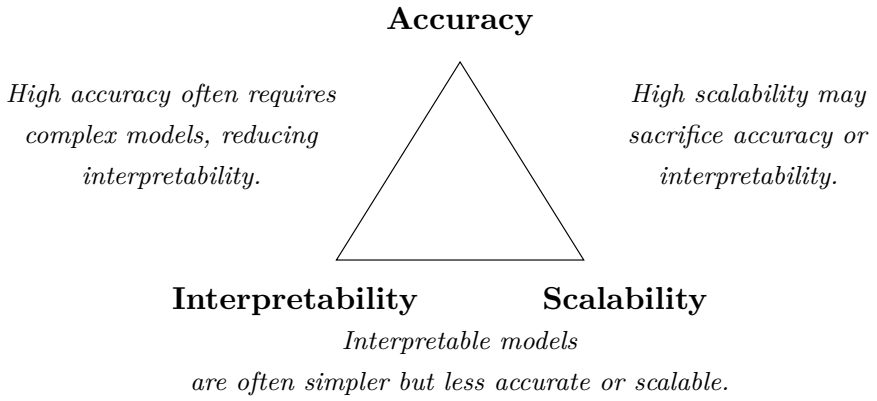


Figure 2.2: *Trade-offs Between Accuracy, Scalability, and Interpretability in Fraud Detection Systems.*

To address these trade-offs, researchers often adopt hybrid approaches that combine the strengths of different models. For example:

- Hybrid Ensemble Models combine the accuracy of deep learning models with the interpretability of simpler models, achieving a balance between performance and transparency.
- eXplainable Artificial Intelligence (XAI) techniques, such as SHapley Additive exPlanations (SHAP) or Local Interpretable Model-Agnostic Explanations (LIME), are increasingly used to provide insights into the decision-making process of complex models without sacrificing accuracy.

By carefully balancing these trade-offs, modern fraud detection systems can achieve high accuracy, scalability, and interpretability, ensuring compliance with regulatory standards while effectively detecting fraudulent activities.

2.4.2 Advanced and Emerging Technologies

Recent advancements in privacy-preserving fraud detection have introduced zero-knowledge proofs and Federated learning approaches [2, 61]. Cross-exchange collaboration frameworks using Federated learning are also being developed to create decentralised systems while ensuring data privacy. Additionally, quantum-resistant cryptographic methods are being explored to future-proof detection systems against potential quantum computing threats [6].

Quantum Computing and Fraud Detection

One of the emerging challenges in fraud detection is the potential impact of quantum computing. Quantum computers, with their ability to perform complex calculations at unprecedented speeds, may disrupt current cryptographic systems that underpin blockchain technology. This poses a significant threat to the security of cryptocurrency transactions, as quantum algorithms like Shor's Algorithm may potentially break widely used cryptographic protocols, such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC), which are used to secure blockchain networks. To address this challenge, researchers are exploring quantum-resistant cryptographic methods [66].

These methods aim to develop algorithms that are secure against quantum attacks, ensuring the long-term security of blockchain systems. Table 2.3 summarizes the key quantum-resistant cryptographic methods and their advantages.

Table 2.3: Quantum-Resistant Cryptographic Methods and Their Advantages.

Method	Advantages
Lattice-Based Cryptography	• Resistant to quantum attacks due to the hardness of lattice problems. • Supports a wide range of cryptographic applications, including encryption and digital signatures. • Efficient and scalable for large-scale systems.
Hash-Based Cryptography	• Relies on cryptographic hash functions, which are believed to be quantum-resistant. • Simple and well-understood, with a long history of security analysis. • Suitable for digital signatures and lightweight applications.
Code-Based Cryptography	• Based on error-correcting codes, which are resistant to quantum attacks. • Provides strong security guarantees with relatively small key sizes. • Well-suited for encryption and key exchange protocols.
Multivariate Cryptography	• Uses systems of multivariate polynomial equations, which are hard to solve even for quantum computers. • Efficient for digital signatures and authentication. • Offers a high level of security with compact key sizes.
Post-Quantum Blockchain Protocols	• Integrates quantum-resistant algorithms into blockchain systems. • Ensures long-term security against quantum threats. • Compatible with existing blockchain architectures.

In addition to cryptographic advancements, quantum computing also has the potential to enhance fraud detection capabilities. Quantum ML algorithms may process large datasets and identify complex fraud patterns more efficiently than classical algorithms. However, this technology is still in its early stages, and significant research is needed to fully realize its potential.

2.5 Regulatory Landscape and Compliance

The rapid evolution of cryptocurrency poses significant regulatory challenges and opportunities. As the market matures, regulatory bodies worldwide are developing frameworks to ensure compliance, transparency, and security in cryptocurrency transactions.

Recent regulatory developments have significantly influenced fraud detection requirements:

- European Union's MiCA regulation (2023) [29].

- FATF Updated Guidance (2023) [31].
- US Treasury FinCEN guidelines (2023) [32].

The MiCA regulation, introduced in 2023, aims to establish a comprehensive regulatory framework for cryptocurrencies within the EU. It requires that crypto service providers obtain licences to operate and meet stringent operational and security standards. Additionally, MiCA enhances consumer protection by requiring transparency in the information provided to consumers, including clear disclosures about the risks associated with cryptocurrency investments. Furthermore, the regulation enforces AML compliance, obliging service providers to implement robust customer enhanced due diligence (EDD) processes, to prevent illicit activities.

Meanwhile, in December 2024, significant amendments were introduced to MiCA, expanding its scope to address emerging trends in cryptocurrency and blockchain technologies. These updates include enhanced consumer protection, stricter transparency standards for Virtual Asset Service Providers (VASP), and new provisions ensuring compliance with the European AI Act guidelines. The amendments also emphasize cross-border coordination by requiring VASP to implement standardized Application Programming Interface (API) for compliance with international financial regulations. Enhanced AML requirements introduce stricter customer due diligence standards, particularly for decentralized platforms. Furthermore, data privacy provisions explicitly reference General Data Protection Regulation (GDPR) compliance in the context of transaction metadata, enforcing encryption and data minimization practices. These regulatory updates reflect the EU's commitment to achive robust financial security standards while fostering innovation in the cryptocurrency ecosystem. To ensure full compliance, this work incorporates model transparency through XAI techniques and implements real-time monitoring mechanisms tailored to the amended MiCA framework.

Similarly, the FATF released updated guidance in 2023 on regulating virtual assets and VASP, focusing on a risk-based approach to combat money laundering and terrorism financing. The guidance reinforces compliance with the "Travel Rule" requiring certain information about the sender and recipient of virtual asset transfers to accompany transactions above a specified threshold.

The US FinCEN regulates cryptocurrency transactions under the Bank Secrecy Act (BSA). Cryptocurrency exchanges must register as Money Service Business (MSB) and comply with BSA regulations. FinCEN requires the reporting of suspicious activities and large transactions to enhance financial transparency and prevent illicit activities.

2.6 Research Gaps and Future Directions

Current research identifies key gaps in cryptocurrency fraud detection [49]:

- Scalability vs. Accuracy — Balancing detection accuracy with computational efficiency to optimising detection models for large-scale blockchain transactions.

- Adaptability — Developing systems that can quickly adapt to new fraud patterns.
- Privacy Preservation — Achieving effectiveness while protecting user privacy.
- Explainability — Building transparent AI aligned with regulations.

2.6.1 Future Research Directions

Future research ought to focus on incorporating explainable AI solutions, enhancing cross-chain transaction monitoring, and advancing quantum-resistant fraud detection techniques. Layer 2 scaling solutions present unique challenges, as they introduce additional complexity in transaction patterns [8, 45]. Detection mechanisms must be adapted for optimistic and zero-knowledge rollups [18, 41], where transaction finality and visibility patterns differ from Layer 1.

Cross-chain transaction monitoring requires standardised approaches to feature engineering across different blockchain protocols [37, 49], whilst maintaining chain-specific optimisation.

2.6.2 Implementation Challenges for Hybrid Detection Systems

The empirical foundation for many of the methods discussed in this chapter is supported by several key datasets widely used in cryptocurrency fraud detection research. Notable among these, evolving BTC, is the BitcoinHeist Dataset, which is designed for ransomware detection and financial crime analysis [3, 7, 68]. Another important dataset is AMLSim, a synthetic dataset that simulates various illicit financial activities. It has been widely used as an instrumental resource in advancing research on detection of AML [38, 46, 79]. Additionally, the Elliptic Dataset—one of the largest publicly available cryptocurrency datasets—categorizes transactions as licit or illicit based on real-world entities [26, 78].

The implementation of hybrid systems that combine supervised and unsupervised approaches presents several challenges:

1. Data integration challenges — Combining different data sources and formats required for each approach.
2. Model selection complexity — Determining the optimal combination of supervised and unsupervised techniques.
3. Decision fusion mechanisms — Creating effective methods to integrate outputs from different detection systems.
4. Computational resource optimization — Balancing the resource demands of multiple concurrent models.
5. Performance metrics — Developing appropriate metrics to evaluate hybrid systems.

Addressing these challenges requires a systematic approach that considers both the theoretical foundations and practical implementation constraints of cryptocurrency fraud detection systems.

3 Proposed Methodology

Chapter 3 introduces the technical and methodological framework underlying the development of the proposed BAD system. Leveraging the Elliptic Bitcoin dataset, this chapter addresses key challenges in cryptocurrency fraud detection, including high-dimensional data, severe class imbalance, and the need for interpretability. The approach unfolds across five methodological phases, progressively building a hybrid, semi-supervised pipeline capable of robust classification and anomaly detection within transaction graphs.

The chapter is structured as follows:

Section 3.1 describes the Elliptic Bitcoin dataset in detail, including its structure, features, temporal dynamics, and class distribution, emphasising challenges posed by label scarcity and imbalance.

Section 3.2 outlines the proposed multi-phase pipeline, beginning with EDA and preliminary classification, and introduces the core classification models and evaluation metrics used throughout the methodology.

Section 3.3 presents a graph-based analysis of transaction networks, examining network structure, node centrality, and behavioural insights derived from both *licit* and *illicit* subgraphs through an integrated structural and forensic approach.

Section 3.4 details the methodology for dimensionality reduction, feature selection, and handling class imbalance to improve classifier effectiveness while addressing interpretability and robustness in classification performance.

Section 3.5 explores unsupervised and semi-supervised learning approaches, including clustering algorithms, rule mining, and an integrated classification strategy for handling unlabelled transactions. By combining graph theory, machine learning, and hybrid modelling techniques, this chapter establishes a scalable basis for empirical evaluation, presented in Chapter 4.

3.1 Elliptic bitcoin transaction dataset

The Elliptic Bitcoin transaction dataset [26, 79], is presented in three data files:

1. `elliptic_txs_edgelist.csv` — Contains edge information, mapping Bitcoin flows between transactions.
2. `elliptic_txs_classes.csv` — Provides the legality classification of transactions (*licit*, *illicit*, or unlabelled).
3. `elliptic_txs_features.csv` — Contains the 166 anonymised features for each transaction.

It comprises a Bitcoin blockchain transaction graph where nodes represent individual transactions (203,769 total) and edges denote Bitcoin flows between them (234,355 connections).

The dataset has 166 attributes per transaction, organized into two categories:

1. Local Features (94) — Provide direct information about the transaction, such as Number of inputs/outputs, Transaction fees, Output volume, and Aggregated metrics (e.g., average Bitcoin (BTC) received or spent by inputs/outputs, average number of incoming/outgoing transactions for inputs/outputs).
2. Aggregated Features (72) — Derived by examining transactions one-hop backward or forward in the graph. These include maximum, minimum, and standard deviation values, as well as correlation coefficients for parameters like inputs/outputs, transaction fees, and output volume.

We have a significant class imbalance, as described in Table 3.1.

Table 3.1: Elliptic dataset class distribution

Class	**Count**	**Percentage**
Illicit (class 1)	4,545	2.2%
Licit (class 2)	42,019	20.6%
Unknown (class 3)	157,205	77.2%

This poses challenges for supervised learning approaches as labeled data represents 22.85% of the dataset (9.76% *illicit* vs 90.24% *licit*). An even larger fraction of the data (77.2%) remains unlabeled, underscoring the need for robust unsupervised and semi-supervised methods.

Figure 3.1 outlines the temporal distribution of transactions across time. Transactions are segmented into 49 time steps, each representing a two-week interval. Each time step forms a connected component where transactions occurred within three hours of one another. There are no edges linking transactions across different time steps.

We observe that transaction counts vary significantly across time steps and early time steps (1—5) contain more labeled data. The total values vary significantly, with a minimum of 1089 at time step 27 and a peak of 7880 at time step 1.

For Class 1 (*illicit*), we observe that generally has smaller values relative to the other classes. The highest recorded value is 342 at time step 32 whereas the lowest is 2 at time

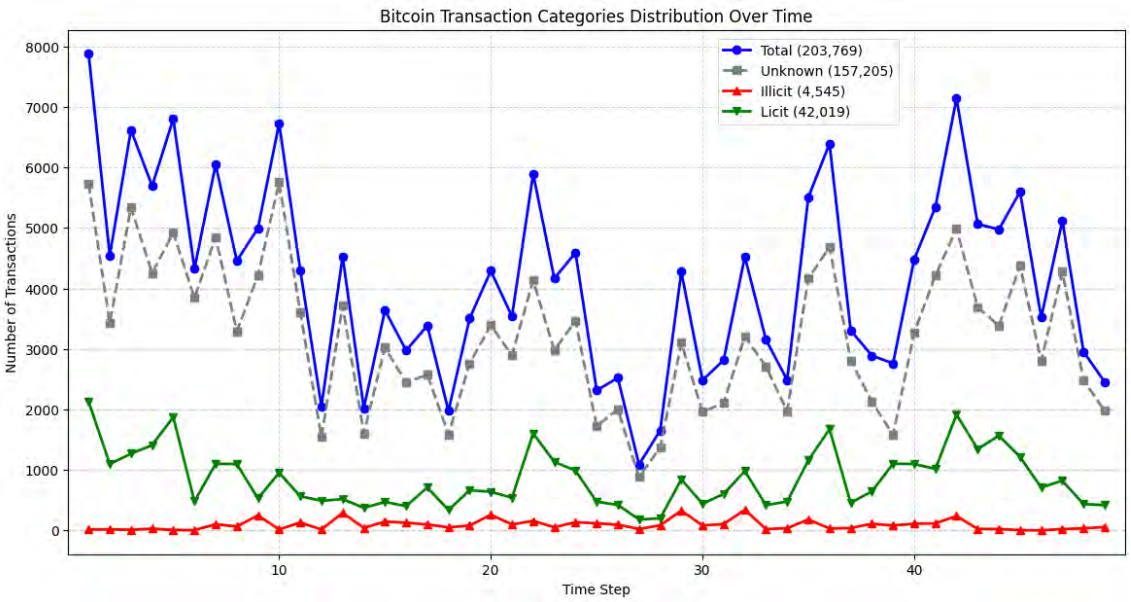


Figure 3.1: *Temporal distribution of transactions across time.*

step 46. Some time steps, such as 13, 29, and 32 display noticeable peaks with 291, 329, and 342 values.

For class 2 (*licit*), the values exhibit a broad range, from 182, at time step 27, to 2130 at time step 1. Certain time steps (e.g., 8, 26, and 27) have relatively lower values compared to others.

The *unknown* class consistently holds the highest values among the three classes. The maximum value is 5755, at time step 10, while the lowest is 883, at time step 27.

Regarding outliers and special cases, at time step 1, the total value is at its peak 7880, primarily due to the dominance of the *unknown* class. Time step 27 has the lowest total value of 1089, with minimal values across all three classes. We have a notable peak at time step 42, with 7140, where Class 1 records 1915 and Class 2 reaches 239.

3.2 Proposed Approach

The proposed approach follows a structured multi-phase pipeline, designed to progressively enhance understanding and model effectiveness. The pipeline consists of the following key phases:

1. Baseline Exploratory Data Analysis (EDA)
2. Graph Network Analysis
3. Dimensionality Reduction and Feature Engineering
4. Imbalance Handling and Supervised Classification
5. Unsupervised and Semi-Supervised Learning

Baseline Exploratory Data Analysis (EDA)

The Elliptic Bitcoin Transaction dataset serves as the empirical foundation, offering a comprehensive benchmark with 166 anonymized features and a transaction graph (200,000+ nodes, 230,000+ edges) that categorizes transactions as *licit*, *illicit*, or unlabeled [26]. Its temporal dynamics, aggregated features, and ground-truth labels uniquely support anomaly detection evaluation.

The baseline EDA phase, to characterizing the dataset structure and statistical properties, is depicted in Figure 3.2. From the Elliptic dataset, we provide an exploratory analysis of the data, splitting the data into two and three classes. We apply classifiers over the dataset. At the end of this pipeline, we evaluate ML models to find the best ones.

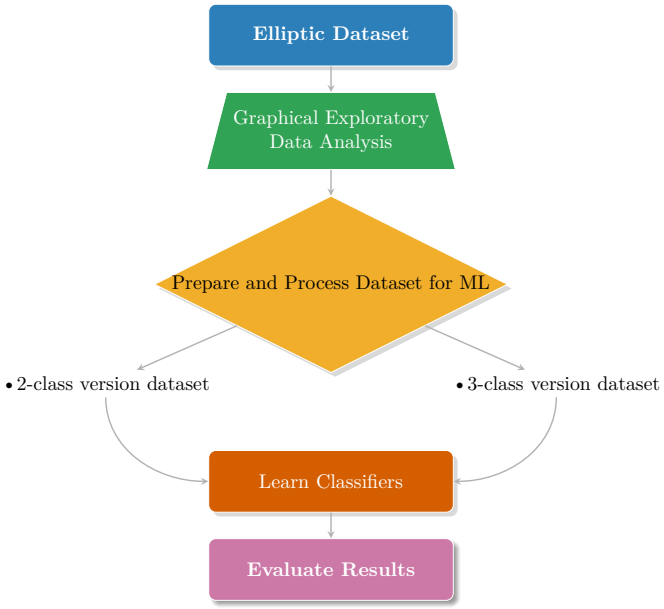


Figure 3.2: *Workflow of the EDA*

Graph Network Analysis

The graphical analysis workflow is illustrated in Figure 3.3. Starting from the full transaction network dataset, we conduct a structural analysis focusing on key graph properties such as degree distribution and edge density. Based on this analysis, the network is partitioned into licit and illicit subgraphs, allowing for targeted investigation of each subset.

Then, the focus become to determine whether the licit portion reveals a moderately connected hierarchical structure, and analyse the illicit segment for fragmented subgraphs and potential sink account behaviour. Based on these findings, we proceed with a subsequent forensic node-level analysis, centred on a representative *illicit* node. This final step may uncover layered transaction patterns indicative of money laundering strategies.

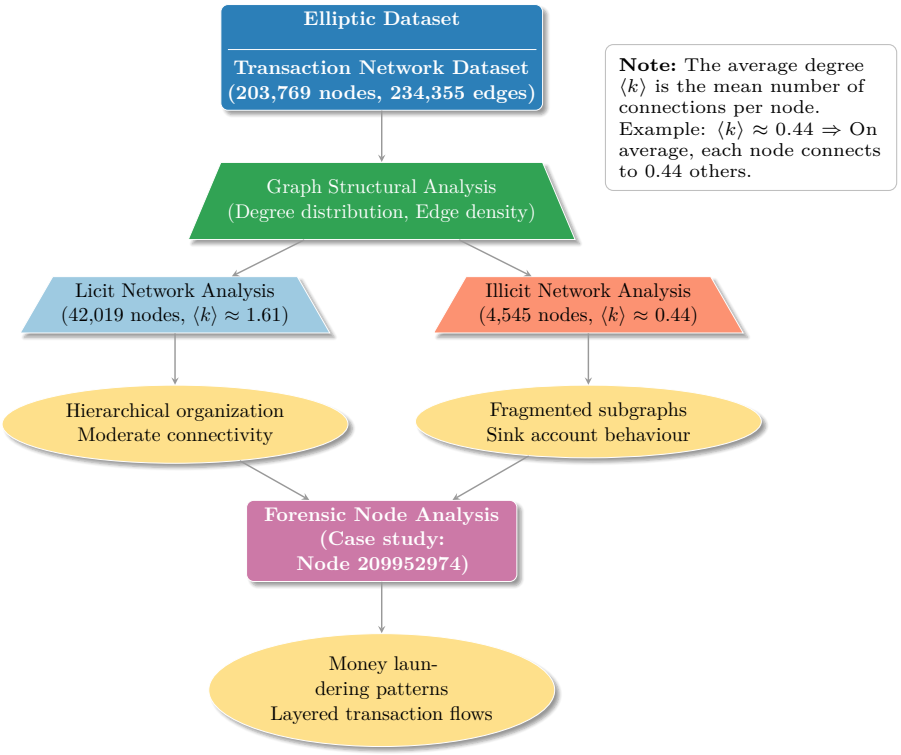


Figure 3.3: *Graphical Analysis of Transaction Network with Structural Insights*

Dimensionality Reduction and Feature Engineering

Addressing the high dimensionality inherent in transaction networks, Dimensionality Reduction (DR) techniques are employed to uncover underlying patterns and enhance data interpretability. The DR phase is depicted in Figure 3.4. From the dataset versions, with two or three classes, we perform DR with Feature Selection (FS) and Feature Reduction (FR) techniques. We also explore combinations of FS and FR methods. The goal of this phase is to find the best performing reduced dimensionality version for this dataset and to identify the most decisive features for explainability purposes.

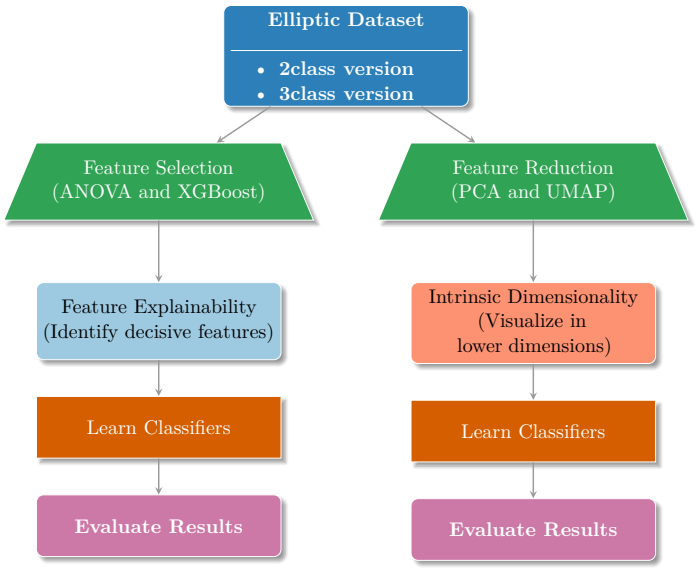


Figure 3.4: *Dimensionality Reduction workflow.*

Imbalance Handling and Supervised Classification

Addressing the high class imbalance inherent in fraud detection, the instance sampling phase is depicted in Figure 3.5. From the original imbalanced dataset, we apply a hybrid instance sampling approach combining Edited Nearest Neighbors (ENN) and Synthetic Minority Over-sampling Technique (SMOTE).

First, SMOTE is used to synthetically oversample the minority class, followed by ENN to clean noisy or ambiguous samples. We also explore the impact of applying each technique individually, as well as their combined effect. The objective of this phase is to achieve a more balanced and robust dataset, enhancing the performance of the classifiers and improving the reliability of the results, particularly for the underrepresented classes.

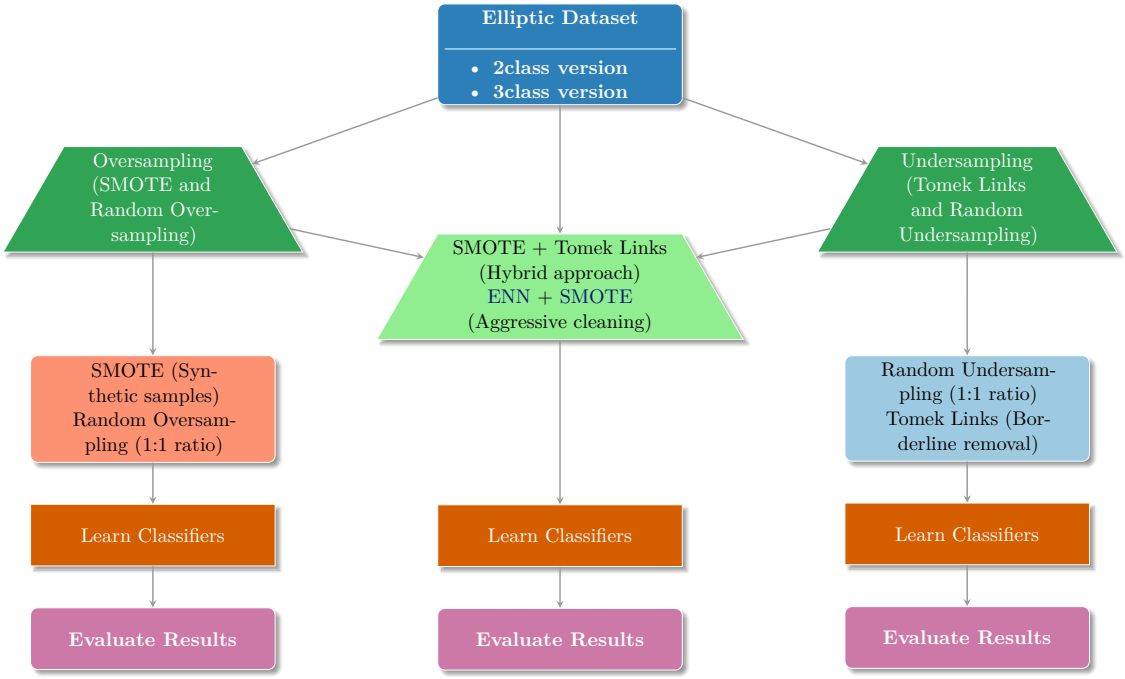


Figure 3.5: *Imbalance Handling Techniques Workflow with Oversampling, Undersampling and Hybrid Approaches*

Unsupervised and Semi-Supervised Learning

We then explore an unsupervised pipeline approach to label the remaining unlabelled data (77.2%) as either *licit* or *illicit*.

Our comprehensive clustering workflow, depicted in Figure 3.6 implements a dual-path analytical approach from the original transaction dataset. For 2-class data (*licit*/*illicit*), we employ standard clustering after FS. For 3-class data (+*unknown*), we extend the pipeline with association rule mining and semi-supervised refinement.

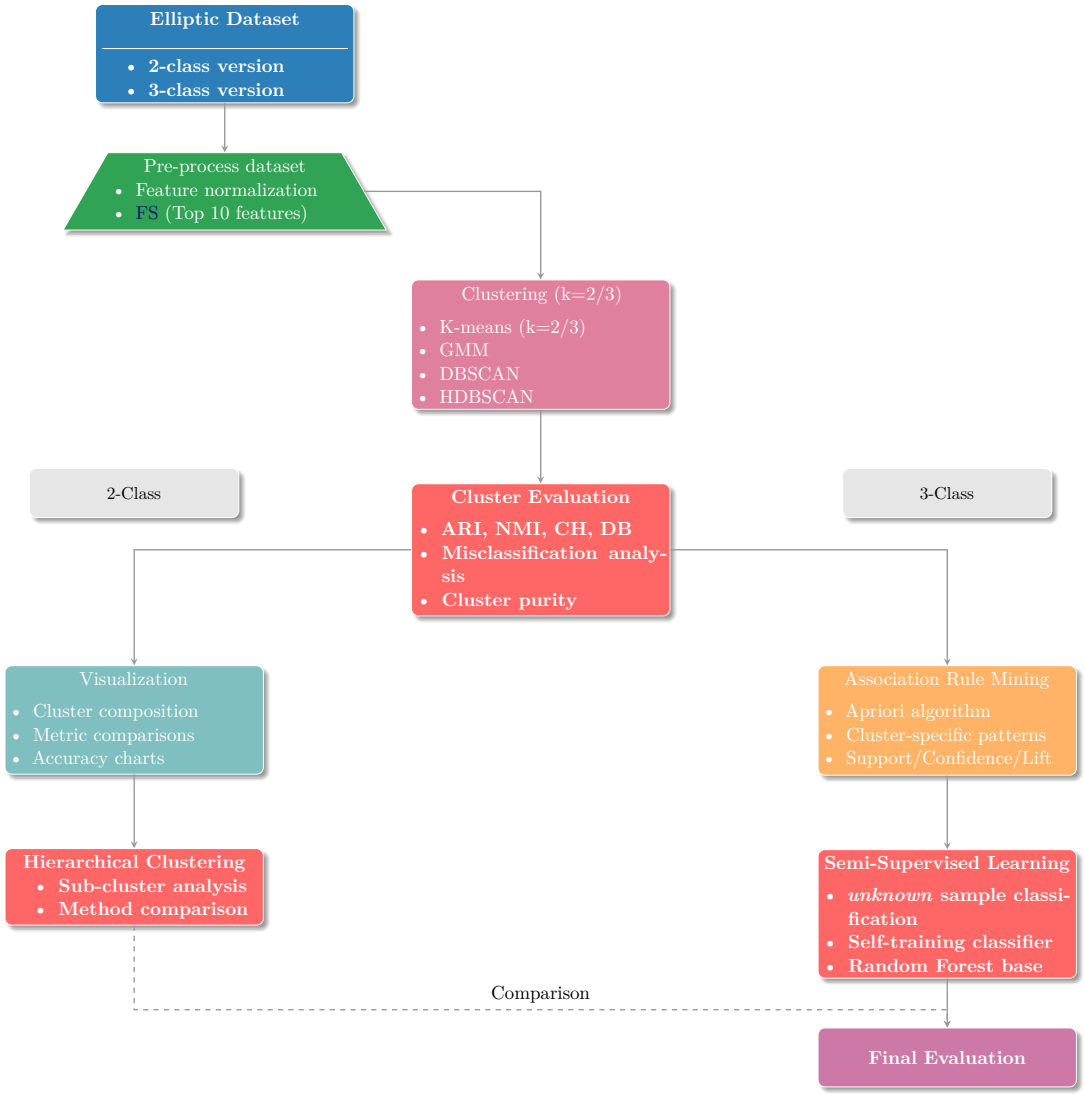


Figure 3.6: *Clustering Workflow for both 2-Class and 3-Class processing, including clustering evaluation, association rule mining and semi-supervised refinement.*

The workflow begins with preprocessing using multiple scaling methods (Standard, MinMax, Robust, and Normalizer), followed by FS using XGBoost importance scores and ANOVA F-test statistics. We apply K-means, GMM, DBSCAN, and HDBSCAN clustering algorithms, evaluating results with four metrics: Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Calinski-Harabasz (CH), and Davies-Bouldin (DB) scores.

For 3-class scenarios, we enhance analysis through Apriori algorithm-based pattern mining to identify cluster-specific transaction characteristics, followed by semi-supervised retraining using Self-Training Classifier with RF and Decision Tree (DT) bases. This approach first understands natural cluster structure, then refines classification of ambiguous transactions through rule-based labeling and model retraining.

Using this approach, we aim to achieve:

1. Accurate baseline clustering of clearly *illicit* versus *licit* transactions
2. Meaningful interpretation of uncertain transactions through association rules

3. Improved final classification through semi-supervised learning

Additional experimental analyses focused on feature selection importance, histogram visualizations, Gaussian-based feature transformations, clustering variations, advanced refinement techniques, and multiple classification strategies to support our conclusions and propose future directions.

3.2.1 Classification Models

For the classification task, we evaluated several well-known classifiers:

- Random Forest (RF) — An ensemble method that builds multiple decision trees and merges their outputs to improve predictive performance and control overfitting.
- Decision Tree (DT) — A tree-structured model that splits data based on feature values to make decisions, known for its interpretability.
- eXtreme Gradient Boosting (XGBoost) — A gradient boosting framework optimised for speed and performance, often outperforming other models on structured data.
- Light Gradient Boosting Machine (LightGBM) — A highly efficient gradient boosting framework that uses histogram-based algorithms and supports categorical features natively.
- CatBoost — A gradient boosting library by Yandex that is particularly effective with categorical data and robust to overfitting
- Support Vector Machines (SVM) — A model that finds the optimal hyperplane to separate classes with maximum margin, effective in high-dimensional spaces.
- K-Nearest Neighbours (KNN) — A non-parametric method that classifies instances based on the majority label among the k closest training examples.
- Multilayer Perceptron (MLP) — A class of feedforward artificial neural networks that consists of multiple layers and can capture complex nonlinear relationships.
- Naïve Bayes (NB) — A probabilistic classifier based on Bayes' theorem with strong (naïve) independence assumptions between features.

3.2.2 Evaluation Metrics

We employ the following evaluation metrics throughout our methodology:

The accuracy is defined as the proportion of correctly predicted instances

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.1}$$

where TP is the True Positives, TN is the True Negatives, FP is False Positives, and FN is False Negatives.

Precision is the proportion of correctly predicted positive instances out of all instances predicted as positive, given by

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{3.2}$$

Recall is the proportion of actual positive instances that were correctly identified, given by

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.3}$$

The F1-score is the harmonic mean of Precision and Recall. It balances both the concerns of false positives and false negatives, defined as

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.4}$$

A high F1-score indicates both high precision and high recall. We have also considered the use of the confusion matrix, the area under the ROC curve designated as Area Under the ROC Curve (AUC), and the Precision-Recall Curves (PR) trade-off, preferred for class imbalance assessment.

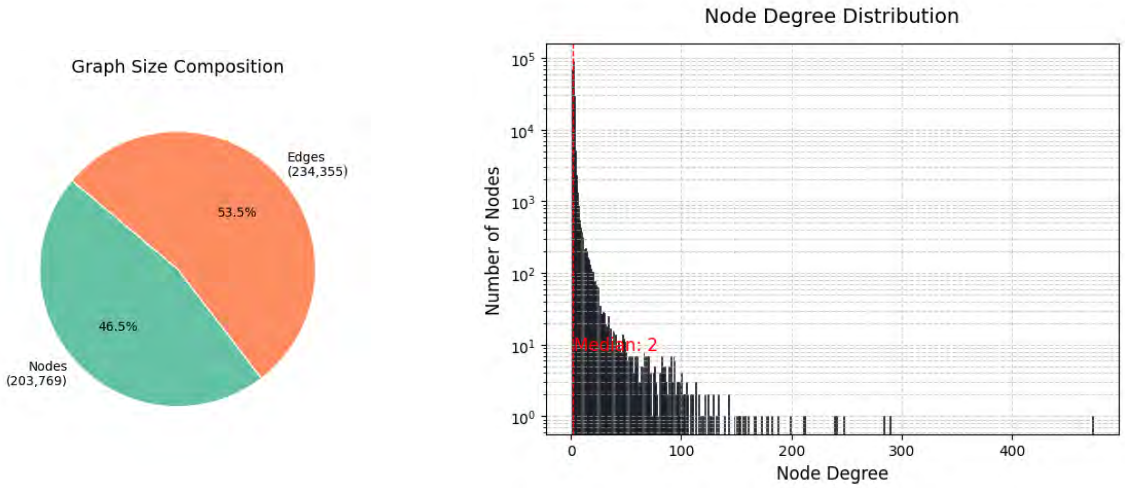
3.3 Graph Structure and Network Analysis

In addition to feature-based models, graph-structured insights are incorporated via GNN and GCN. These methods allow the detection of community structures and behavioral motifs typical of fraud, such as transaction layering and sink node patterns.

3.3.1 Network Structure Analysis

This section presents an overview of the graph's structural characteristics, focusing on its overall size and the distribution of connections between nodes. We begin by outlining the graph's composition in terms of the number of nodes and edges, providing insight into its scale and density. Following this, we examine the node degree distribution to understand how connectivity is spread across the network, highlighting any significant patterns or irregularities that may inform further analysis.

Figure 3.7 illustrates the Graph size degree distribution. The analyzed graph consists of 203,769 nodes and 234,355 edges, exhibiting a sparse structure with an average node degree of ∼2.3. The degree distribution displays significant left-skewed heterogeneity, where the majority of nodes (≥95%) have fewer than 2–3 connections, while a limited set of hub nodes ($degree \leq 400$) dominate the connectivity. These topological features suggest efficient information propagation pathways and potential modular organization, where hubs bridge localized clusters. The coexistence of peripheral nodes and central hubs highlights the tension between decentralization and critical vulnerability in such systems.



(a) *Graph size Composition* (b) *Graph degree distribution*

Figure 3.7: *Graph size degree distribution*

Figure 3.8 illustrates the overall *licit* and *illicit* class Graph. The transaction network comprises two distinct classes: *illicit* (Class 1) with 4,545 nodes and 998 edges, and *licit* (Class 2) with 42,019 nodes and 33,930 edges.

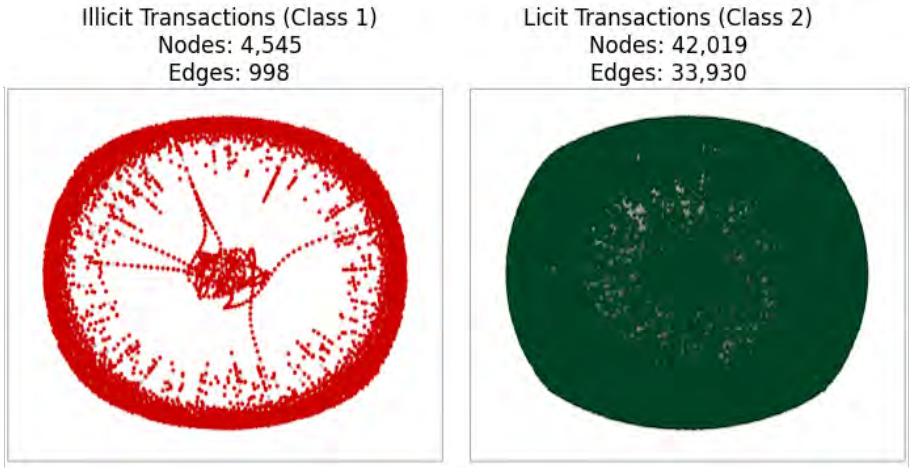


Figure 3.8: *Illicit and licit Graph visualization*

The average node degree denoted as $\langle k \rangle$ is computed as:

$$\langle k \rangle = \frac{2 \times \text{Number of Edges}}{\text{Number of Nodes}}. \tag{3.5}$$

The edge density denoted as ρ is given by:

$$\rho = \frac{\text{Edges}}{\frac{\text{Nodes} \times (\text{Nodes}-1)}{2}}. \tag{3.6}$$

Both the average degree $\langle k \rangle$ and the edge density $\langle \rho \rangle$ have well-defined theoretical bounds. The average degree ranges from 0 (no edges) to $N-1$, the maximum in a fully connected graph with N nodes. Edge density spans from 0 to 1, where 1 denotes a complete graph.

These measures capture distinct connectivity aspects: average degree reflects links per node, while edge density indicates connection utilization. Higher density in smaller graphs often signals tightly knit substructures, particularly relevant for fraud or anomaly detection.

Table 3.2 summarises the values of $\langle k \rangle$ and $\langle \rho \rangle$ across three graph types, each showing distinct connectivity patterns.

Table 3.2: Summary of Ratio and Edge Density for Different Graph Classes

Graph	**Ratio** $\langle k \rangle$	**Edge Density** $\langle \rho \rangle$
First Graph (Figure 3.7a)	2.3	1.1×10^{-5}
Illicit Class	0.44	9.7×10^{-5}
Licit Class	1.61	3.8×10^{-5}

We observe that:

- The First Graph has the highest average degree ($\langle k \rangle = 2.3$) but the lowest edge density ($\rho = 1.1 \times 10^{-5}$), indicating a large but extremely sparse network.
- The *illicit* Class presents a very low average degree ($\langle k \rangle = 0.44$) yet a relatively high edge density ($\rho = 9.7 \times 10^{-5}$), suggesting a smaller but more densely connected network.
- The *licit* Class sits between the two, with $\langle k \rangle = 1.61$ and $\rho = 3.8 \times 10^{-5}$, representing a moderately sparse, diffusely connected legitimate network.

The variance for both the average degree $\langle k \rangle$ and the edge density $\langle \rho \rangle$ was computed using the sample variance formula:

$$\mathrm{Var}(x) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2, \tag{3.7}$$

where x_i represents each observed value, $\bar{x}$ is the sample mean, and n is the number of observations (in this case, $n = 3$).

From Table 3.3a, we observe significant variance in the average degree ($\mathrm{Var}(k) = 0.884$, $\sigma_k = 0.94$), while Table 3.3b reveals comparable dispersion in edge density ($\mathrm{Var}(\rho) = 1.93 \times 10^{-9}$, $\sigma_\rho = 4.40 \times 10^{-5}$). These metrics reflect meaningful structural differences across the graphs, even within a generally sparse context.

Table 3.3: Variance Analysis of Network Characteristics

(a) Average Degree ($\langle k \rangle$)

Metric	**Value**
Values	2.3, 0.44, 1.61
Mean ($\bar{k}$)	1.45
Variance (Var (k))	0.884

(b) Edge Density (ρ)

Metric	**Value**
Values (ρ)	$1.1, 9.7, 3.8\ 10^{-5}$
Mean ($\bar{\rho}$)	4.87×10^{-5}
Variance (Var (ρ))	1.93×10^{-9}

The standard deviation σ_k is calculated from the variance $\mathrm{Var}(k)$ using $\sigma_k = \sqrt{\mathrm{Var}(k)}$.

The illicit network's higher density despite its size suggests coordinated behaviour like rings or cells, with targeted connections possibly reducing detection risk. Conversely, licit

and full graphs show broader but diffuse connectivity, indicating organic or decentralised interactions.

The network exhibits a power-law degree distribution, with observed degrees ranging from 0 to 400. Most nodes have fewer than or equal to 2—3 connections, consistent with the distribution $P(k) \sim k^{-\gamma}$. Given $n = 203,769$ nodes, the theoretical maximum number of edges is $\frac{n(n-1)}{2} \approx 2.07 \times 10^{10}$.

The node ratio between licit and illicit entities is $\frac{\text{Licit}}{\text{Illicit}} = \frac{42,019}{4,545} \approx 9.2$ (approximately 9 times), while the edge ratio is $\frac{33,930}{998} \approx 34.0$ (approximately 34 times).

Illicit networks exhibit sparse connectivity ($\langle k \rangle \approx 0.44$), indicating fragmented subgraphs that may evade detection, while licit networks show moderate connectivity ($\langle k \rangle \approx 1.61$), suggesting a hierarchical structure with localized hubs. Both classes are sparse (edge densities $\sim 9.7 \times 10^{-5}$ and 3.8×10^{-5} respectively), but licit transactions are paradoxically sparser due to their larger node count despite higher average degree.

Illicit activities form disconnected, low-degree clusters to minimize footprints and evade detection, while licit interactions adopt hierarchical topologies with localized hubs for systematic coordination. The stark scale disparity (Class 2 is $\sim 9\times$ larger in nodes, $\sim 34\times$ in edges) reflects fundamental behavioral differences: illicit networks prioritize stealth through sparsity, while licit systems balance efficiency with traceability. This dichotomy complicates anomaly detection, as illicit patterns may mimic noise in licit-dominated graphs.

3.3.2 Node Centrality and Classification Analysis

Figure 3.9 illustrates the ten overall nodes, each one characterized by a in-degree and a out-degree.

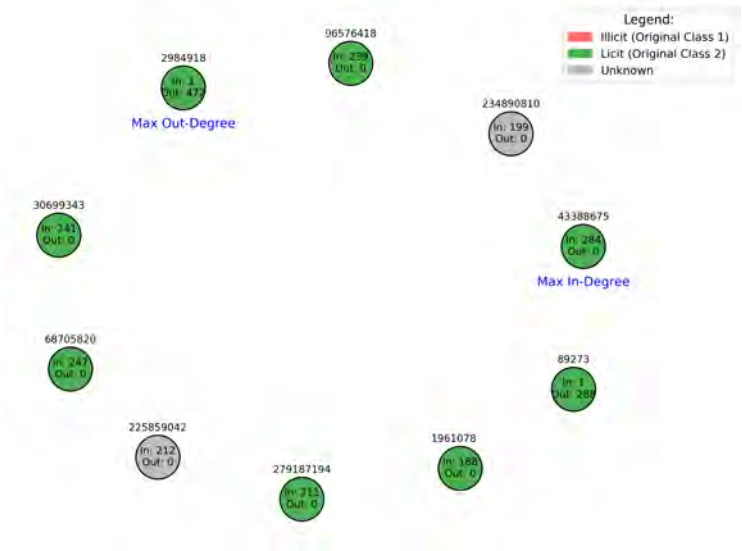


Figure 3.9: *Top 10 Overall Nodes*

- In-degree (k_{in}): Count of incoming edges to node v (measures received connections)

$$k_{in}(v) = \left|\{u \in V \mid (u, v) \in E\}\right| \tag{3.8}$$

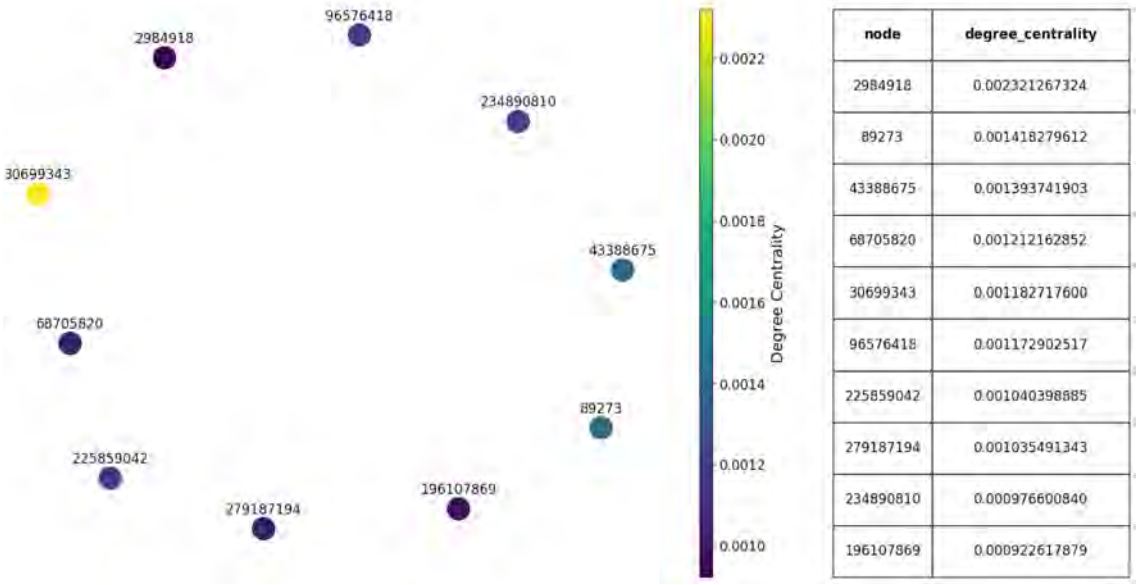
- Out-degree (k_{out}): Count of outgoing edges from node v (measures initiated connections)

$$k_{out}(v) = \left|\{u \in V \mid (v,u) \in E\}\right|. \tag{3.9}$$

$G = (V, E)$ is a directed graph with vertices V and edges E, and (u, v) denotes an edge from node u to v.

Figure 3.10 reveals degree centrality measures relative connectivity by calculating the fraction of nodes each vertex is connected to. The degree centrality is given by:

$$\text{Degree Centrality} = \frac{\text{Degree of Node}}{\text{Total Nodes - 1}} \tag{3.10}$$



node	degree_centrality
2984918	0.002321267324
89273	0.001418279612
43388675	0.001393741903
68705820	0.001212162852
30699343	0.001182717600
96576418	0.001172902517
225859042	0.001040398885
279187194	0.001035491343
234890810	0.000976600840
196107869	0.000922617879

Figure 3.10: *Top 10 Overall Nodes — Degree centrality*

Node 2984918 emerges as the dominant hub with a centrality of 0.00221 ($\approx 2.2\times$ higher than the 10th-ranked node), suggesting it functions as a critical junction in transaction flows. The sudden decline in centrality between ranks—for instance, from 0.00221 to 0.00141 between the 1st and 2nd positions—indicates a strongly hierarchical network structure rather than uniform connectivity.

All centrality values are < 0.003, confirming overall network sparsity; even the most central nodes are connected to only $\approx 0.2\%$ of possible peers. High-centrality nodes likely correspond to exchange services or mixing entities, whereas the long tail of lower centrality values implies that most nodes participate in specialised or localised interactions.

Methodological Note

Degree centrality was normalized by network size ($n = 203,769$ nodes), making values comparable across different graphs. For reference:

$$\text{Expected random value} \approx \frac{\langle k \rangle}{n-1} \approx \frac{1.45}{203,768} \approx 7.1 \times 10^{-6}.$$

The top nodes' centrality values are 300–600 times higher than random expectation, confirming their structural significance.

We observe that there are no *illicit* nodes in the top 10 overall network.

Figure 3.11 shows the top 10 nodes originally classified as *illicit* (Class 1). Each one is characterised by a high in-degree and a zero out-degree. This indicates that these nodes are exclusively *receivers* of funds, without forwarding transactions elsewhere.

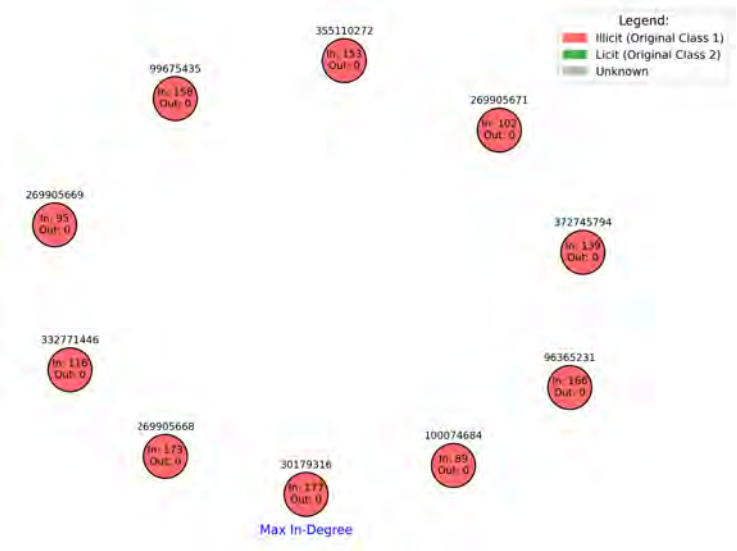


Figure 3.11: *Top 10 Overall illicit Nodes*

Notably, node `30179316` exhibits the highest in-degree, receiving 177 transactions, and is highlighted as the Max In-Degree node. The uniform pattern of high inbound connectivity and the complete absence of outbound activity suggests that these nodes may function as *sink accounts* — endpoints where funds are aggregated. This behaviour is typical of wallets used for illicit fund collection, possibly acting as *collector* or *cash-out* nodes in financial crime networks. Their isolated nature and lack of transactional dispersal make them strong candidates for further investigation regarding potential laundering or fraud-related activities.

3.4 Feature Engineering and Dimensionality Reduction

To address the complexities of this dataset, our approach incorporates dimensionality reduction, instance sampling, and ML techniques. We aim to identify the optimal configuration of pre-processing, model selection, and evaluation metrics that maximize fraud detection performance under real-world constraints.

3.4.1 Feature Selection and Dimensionality Reduction Techniques

We evaluate these techniques in both binary (*licit* vs. *illicit*) and multi-class (*licit*, *illicit* and *unknown*) scenarios. For Feature Selection, we primarily consider ANOVA F-value and XGBoost Feature Importance for explainability. For Feature Reduction, we employ PCA and Uniform Manifold Approximation and Projection (UMAP).

ANOVA for Feature Selection

ANOVA is a statistical method that assesses whether the means of different groups are different with statistically significance. For FS, it evaluates whether a feature has significantly different with values across different target classes.

$$F = \frac{\text{MS}_{\text{between}}}{\text{MS}_{\text{within}}} = \frac{\frac{\sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2}{k-1}}{\frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2}{N-k}}, \tag{3.11}$$

where:

- k is the number of classes.
- n_i is the number of samples in class i.
- N is the total number of samples.
- $\bar{x}_i$ is the mean of feature values in class i.
- $\bar{x}$ is the overall mean of the feature.
- x_{ij} is the feature value of the j^{th} sample in class i.
- $\text{MS}_{\text{between}}$ is the between-class mean squared difference.
- $\text{MS}_{\text{within}}$ is the within-class mean squared difference.

The p-value associated with the F-statistic determines the feature's significance. A p-value represents the probability of observing the results under the null hypothesis, assuming no true effect exists. Features with lower p-values (typically < 0.05) are more likely to be relevant.

The method is a univariate filter approach that evaluates each feature independently, assuming a normal distribution of features within each class. It is particularly useful for classification problems due to its simplicity and computational efficiency, operating in $\mathcal{O}(dn)$ time complexity, where d is the number of features. While it effectively detects linear relationships between features and the target variable, it cannot capture feature interactions. Typically, it selects the top features with the lowest p-values, making it a fast and practical choice for high-dimensional data.

XGBoost Feature Importance Score for Feature Selection

XGBoost [20] calculates feature importance scores as part of its training process. These scores can be used for FS by ranking features according to their contribution to the model.

XGBoost offers multiple types of importance scores:

1. Weight (the number of times a feature appears in trees):

$$\text{Importance}_{\text{weight}}(f_i) = \sum_{t=1}^{T} \sum_{j \in \{j | \text{split in tree } t \text{ uses } f_i\}} I(j, t, f_i), \tag{3.12}$$

where:

- $I(j, t, f_i)$ is an indicator function:

$$I(j, t, f_i) = \begin{cases} 1 & \text{if split } j \text{ in tree } t \text{ uses feature } f_i \\ 0 & \text{otherwise} \end{cases}$$

- The inner sum is over all splits j in tree t that use feature f_i

2. Gain (the average gain across all splits using the feature):

$$\text{Importance}_{\text{gain}}(f_i) = \sum_{t=1}^{T} \sum_{j \in \{j | \text{split in tree } t \text{ uses } f_i\}} \text{Gain}_j, \tag{3.13}$$

where Gain_j for a split is defined as:

$$\text{Gain}_j = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma. \tag{3.14}$$

3. Cover (the average coverage across all splits using the feature):

$$\text{Importance}_{\text{cover}}(f_i) = \sum_{t=1}^{T} \sum_{j \in \{j | \text{split in tree } t \text{ uses } f_i\}} \text{Cover}_j, \tag{3.15}$$

where Cover_j is the number of samples affected by the split.

In the equations above:

- T is the number of trees.
- f_i is the i^{th} feature.
- G_L, G_R are the sums of gradients in the left and right nodes after the split.
- H_L, H_R are the sums of hessians in the left and right nodes after the split.
- λ is the L2 regularization term.
- γ is the minimum loss reduction required for a split.

This approach is an embedded method, where FS is integrated into the model training process. It is capable of capturing non-linear relationships as well as interactions between features, making it suitable for complex datasets. Both main and interaction effects are considered, and the method is robust to outliers and missing values. Its computational complexity is tied to the training of XGBoost, typically around $\mathcal{O}(d \cdot n \log n)$. Among the various importance metrics, gain-based importance is generally the most informative. FS can be applied using either threshold-based techniques or recursive feature elimination.

3.4.2 Handling Imbalanced Data

The class imbalance issue is addressed through several sampling strategies:

- Oversampling — Random Oversampling and SMOTE
- Undersampling — Random Undersampling and Tomek Links
- Hybrid Approaches — SMOTE combined with Tomek Links or ENN

To address class imbalance, we consider a variety of instance sampling techniques. SMOTE generates new minority class instances by interpolating between existing ones, helping to enrich the dataset without duplicating samples. It generates synthetic samples by interpolating between a minority class instance and its nearest neighbors. For a sample x_i and one of its k-nearest neighbors x_{zi}, it generates a new instance by

$$\tilde{x} = x_i + \delta \cdot (x_{zi} - x_i), \quad \delta \sim \mathcal{U}(0, 1). \tag{3.16}$$

This creates synthetic data points along the line segment between existing minority samples. Random Oversampling simply duplicates minority class examples to balance the dataset, though it may lead to overfitting.

Random Undersampling, on the other hand, reduces the majority class by randomly removing instances, which can result in information loss. Tomek Links identify ambiguous sample pairs from different classes that are close together and remove the majority class sample to clarify class boundaries. Tomek Links refer to a pair of instances (x_i, x_j) that are considered a Tomek Link if:

$$x_j : j = \arg \min_{x_k \in D, x_k \neq x_i} d(x_k, x_i) \tag{3.17}$$

$$x_i : i = \arg \min_{x_k \in D, x_k \neq x_j} d(x_k, x_j) \tag{3.18}$$

$$\text{and} \quad \text{label}(x_i) \neq \text{label}(x_j) \tag{3.19}$$

Such pairs are considered ambiguous or noisy. Typically, the majority class instance is removed to clean the class boundary. Combining SMOTE with Tomek Links allows for both minority class generation and cleaning of noisy boundary regions. Similarly, ENN + SMOTE improves dataset quality by removing samples (from any class) that differ from the majority of their nearest neighbours, thus enhancing the overall class separation. These strategies aim to mitigate imbalance while preserving the integrity and generalisability of the data.

We also evaluate these techniques in both binary (*licit* vs. *illicit*) and multi-class (*licit*, *illicit* and *unknown*) scenarios. The multi-class setting demands greater care, particularly when applying oversampling strategies, to avoid overfitting and ensure generalisability. The objective of this phase is to address the inherent class imbalance typical of financial transaction datasets and to determine which rebalancing technique yields the best classification performance.

Random oversampling creates exact duplicates of minority class samples until class balance is achieved (33,615 samples in each class). This technique ensures all original data is preserved while establishing a 1:1 class ratio. However, the creation of exact duplicates rather than new informative examples may lead to overfitting, as the model might learn specific patterns present in the limited minority samples rather than generalizing to unseen data.

SMOTE addresses the limitations of random oversampling by generating synthetic samples in the feature space between existing minority observations. This approach creates new samples that follow the distribution of the minority class without exact duplication. By introducing controlled variation, SMOTE potentially reduces overfitting risk compared to random oversampling, though it may occasionally generate unrealistic samples or increase class overlap in complex feature spaces.

Random undersampling removes majority class samples until balance is achieved (3,636 samples per class). While this approach effectively creates a balanced dataset and reduces computational requirements, it discards approximately 90% of the majority class information. This significant information loss risks removing important decision boundary patterns and may reduce the model's ability to accurately recognize legitimate transactions.

Unlike more aggressive approaches, Tomek Links focuses on improving class separation by removing only problematic majority samples near class boundaries. In our dataset, this resulted in minimal change (removing only 186 majority samples), maintaining the distribution at approximately 1:9. The technique preserves most majority class information while slightly clarifying decision boundaries, but the overall imbalance remains largely unaddressed.

Hybrid approach SMOTE + Tomek Links, combines synthetic sample generation with boundary cleaning. First, SMOTE creates synthetic minority samples to achieve balance, then Tomek Links removes majority samples that might cause boundary confusion. The resulting dataset contains 33,615 samples in each class but with cleaner decision boundaries than using SMOTE alone. This combined approach addresses both under-representation and potential class overlap issues.

The ENN + SMOTE combination represents the most comprehensive cleaning approach tested. After applying SMOTE, the ENN algorithm removes samples from both classes that are likely to be noisy or contribute to ambiguous boundaries. This resulted in 33,549 minority samples and 30,609 majority samples, creating a slight reverse imbalance (1.1:1). This technique produced the cleanest decision boundaries but at the cost of higher computational complexity and removing samples from both classes.

3.5 Unsupervised and Semi-Supervised Learning

Unsupervised machine learning enables the discovery of hidden patterns in data without requiring labeled examples. Our study addresses the challenge of classifying a large number of unlabeled transactions by leveraging knowledge from a smaller labeled subset. We adopt a hybrid approach combining unsupervised methods to identify structural patterns with

supervised methods to incorporate known labels, enabling classification into *licit*, *illicit*, and *unknown* categories.

3.5.1 Clustering Algorithms

This section outlines key unsupervised clustering algorithms used in the analysis. The methods range from centroid-based (K-Means) to density-based approaches (DBSCAN, HDBSCAN), each with distinct advantages for different data characteristics [16, 27, 51, 54].

The K-Means algorithm partitions the feature space into k distinct clusters based on feature similarity, minimizing within-cluster variance (inertia) through the optimization problem:

$$\arg\min_{S} \sum_{i=1}^{K} \sum_{x \in S_i} \|x - \mu_i\|^2, \tag{3.20}$$

where $S = \{S_1, \ldots, S_K\}$ denotes the set of clusters, μ_i is the centroid (mean) of cluster S_i, and $\|x - \mu_i\|^2$ is the squared Euclidean distance between point x and centroid μ_i,

$$\text{Squared Euclidean distance:} \quad \|x - \mu_i\|^2 = \sum_{j=1}^{d} (x_j - \mu_{ij})^2. \tag{3.21}$$

The algorithm requires the number of clusters K to be specified in advance and assumes that clusters are spherical and of roughly equal size. It is sensitive to the initial placement of centroids—a limitation often mitigated by the *k-means++* initialisation strategy. The computational complexity is $\mathcal{O}(nk)$ per iteration, where n is the number of data points.

The algorithm initializes K centroids (e.g., randomly or using k-means++), then repeats until convergence by assigning each point to the nearest centroid, and recomputing centroids as cluster means. K-means shows poor performance on non-spherical or irregularly shaped clusters, is sensitive to outliers, and requires careful selection of K.

We used K-Means with $K = 3$ to 10, evaluated using clustering metrics CH, DB, ARI, and NMI (detailed in subsection 3.5.3). This approach worked well with features selected by ANOVA F-score and XGBoost, though it requires careful K selection.

The GMM Clustering is a probabilistic model that assumes that data points are generated from a mixture of several Gaussian distributions. The GMM approach models our data as a mixture of multiple Gaussian distributions, calculating the log-likelihood:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j) \right), \tag{3.22}$$

where:

- $X = \{x_1, \ldots, x_n\}$ are the data points.
- $\pi = \{\pi_1, \ldots, \pi_K\}$ are the mixture weights ($\sum_{j=1}^{K} \pi_j = 1$).
- $\mu = \{\mu_1, \ldots, \mu_K\}$ are the mean vectors.

- $\Sigma = \{\Sigma_1, \ldots, \Sigma_K\}$ are the covariance matrices.
- $\mathcal{N}(x|\mu_j, \Sigma_j)$ is the multivariate Gaussian distribution.

This method requires the number of components K to be specified in advance and differs from hard clustering approaches by providing soft assignments, representing the probability of each data point belonging to a given cluster. It is well-suited for capturing elliptical clusters of varying shapes and sizes. Parameter estimation is performed using the Expectation-Maximization (EM) algorithm, which iteratively updates model parameters to maximise likelihood. The computational complexity is $\mathcal{O}(nK)$ per iteration, although it typically requires more iterations to converge compared to K-means.

This method proved valuable for capturing elliptical clusters of varying shapes and sizes in our dataset, particularly when combined with our Gaussian product analysis of class-specific distributions.

DBSCAN is a density-based clustering algorithm that groups dense regions and identifies noise based on ε and `MinPts`. A point is:

- *Core point* — The ε-neighbourhood contains at least `min_samples` data points.
- *Border point* — Lies within the neighbourhood of a core point but does not itself have enough neighbours.
- *Noise point* — Is neither a core point nor a border point.

In this approach, several key concepts define the clustering structure. A core point p is a point whose ε-neighbourhood $N_\varepsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}$ contains at least MinPts points. From this foundation, we define direct density-reachability: a point q is directly density-reachable from p if p is a core point and $q \in N_\varepsilon(p)$. This relation extends transitively to density-reachability, where q is reachable from p if a chain of core points connects them, with each point directly reachable from its predecessor. Furthermore, two points p and q are density-connected if they are both density-reachable from some common core point o.

The method requires no predefined cluster count (K), detects clusters of arbitrary shape, and explicitly handles outliers as noise. Computationally, it scales as $\mathcal{O}(n^2)$ in general cases, though spatial indexing reduces this to $\mathcal{O}(n \log n)$.

Agglomerative Hierarchical Clustering

This bottom-up approach builds a hierarchy of clusters through iterative merging. It starts with singleton clusters and merges them using:

- Ward (minimize variance).
- Complete (max pairwise distance).
- Single (min pairwise distance).

Time complexity is $O(n^2)$, making it less scalable for large datasets, as in this case.

HDBSCAN extends DBSCAN to hierarchical clustering using:

$$d_{\text{mreach-}K}(a,b) = \max\left\{\text{core}_K(a), \text{core}_K(b), d(a,b)\right\}, \tag{3.23}$$

where $\text{core}_K(x)$ is the distance to the K-th nearest neighbour of x, and $d(a,b)$ is the standard distance between points a and b (e.g., Euclidean).

It extract the clusters using the stability measure:

$$\text{Stability}(C) = \sum_{p \in C} (\lambda_p - \lambda_{\text{birth}}(C)), \tag{3.24}$$

where λ_p is the λ value at which point p falls out of cluster C, and $\lambda_{\text{birth}}(C)$ is the λ value at which cluster C forms.

This method constructs a cluster hierarchy by analyzing density stability, automatically identifying optimal clusters without requiring manual selection. It improves upon traditional approaches by effectively handling varying densities while requiring only a single parameter (either *MinPts* or *min_cluster_size*). The algorithm generates a complete hierarchy of clusters and autonomously determines the most stable configurations through stability analysis. Additionally, it provides probabilistic assignment scores for data points, offering nuanced membership likelihoods rather than binary classifications. While the theoretical computational complexity is $\mathcal{O}(n^2)$, optimized implementations achieve $\mathcal{O}(n \log n)$ efficiency in practice.

The algorithm follows three key steps: constructing a mutual reachability graph, extracting the minimum spanning tree, and pruning clusters based on stability. This approach eliminates the ϵ parameter sensitivity while maintaining DBSCAN's advantages.

3.5.2 Summarization

The comparative analysis reveals distinct trade-offs among clustering algorithms in terms of geometric flexibility, noise robustness, and parameter dependence. K-Means operates efficiently on spherical clusters but lacks inherent noise handling, while DBSCAN and its enhanced variant HDBSCAN accommodate arbitrary shapes with explicit noise identification—the latter automating parameter selection. Agglomerative clustering offers hierarchical insights but requires careful linkage selection. This systematic comparison (Table 3.4) provides practitioners with criteria for algorithm choice: cluster geometry, noise prevalence, and parameter interpretability. Based on these considerations and our dataset characteristics, we selected K-Means for its computational efficiency and suitability for well-separated clusters. For datasets with complex structures and outliers, density-based methods (DBSCAN/HDBSCAN) are generally preferable, whereas K-Means suffices for well-separated convex clusters.

Table 3.4: Comparative Analysis of Clustering Algorithms

Algorithm	**Cluster Shape**	**Noise Handling**	**Key Parameters**
K-Means [11, 51]	Spherical	Poor	k
DBSCAN [28, 69]	Arbitrary	Explicit	ϵ, `min_samples`
Agglomerative [42]	Arbitrary	None	Linkage, distance threshold
HDBSCAN [55]	Arbitrary	Explicit	`min_cluster_size`

3.5.3 Clustering Evaluation and Refinement

Clustering metrics provide quantitative measures to evaluate the quality of partitions generated by clustering algorithms. These metrics fall into two categories: *internal* indices (CH and DB) assess cluster validity using only the inherent data structure, while *external* indices (ARI and NMI) compare results against known ground truth labels.

As summarized in Table 3.5, each clustering metric exhibits distinct characteristics in terms of measurement type, value range, and computational complexity. These properties should guide the choice of appropriate validation criteria based on the clustering task's specific requirements.

Table 3.5: Key Properties of Clustering Metrics

Metric	**Type**	**Range**	**Optimal Value**	**Complexity**
CH [15]	Internal	$[0, \infty]$	Maximize	$\mathcal{O}(nk)$
DB [25]	Internal	$[0, \infty]$	Minimize	$\mathcal{O}(k^2 + nk)$
ARI [40]	External	$[-1, 1]$	Maximize	$\mathcal{O}(n + k^2)$
NMI [72]	External	$[0, 1]$	Maximize	$\mathcal{O}(n + k^2)$

The choice of metric should align with both the clustering algorithm's characteristics (as shown in Table 3.4) and the analytical objectives. Key trade-offs include the availability of ground truth labels (internal vs external validation), shape sensitivity where convex metrics like CH/DB may not suit arbitrary cluster shapes, noise robustness given DB's outlier sensitivity versus ARI/NMI's label dependence, and interpretability considerations between bounded ranges (ARI/NMI) and unbounded scores (CH/DB).

The CH (also known as the Variance Ratio Criterion) [15] measures the ratio of between-cluster dispersion to within-cluster dispersion. Higher values indicate better clustering. It is defined as:

$$\text{CH} = \frac{\text{tr}(B_k)/(k-1)}{\text{tr}(W_k)/(n-k)}, \tag{3.25}$$

where:

- n is the number of data points.
- k is the number of clusters.

- B_k is the between-cluster scatter matrix difined as

$$B_k = \sum_{j=1}^{k} n_j(\mu_j - \mu)(\mu_j - \mu)^T. \tag{3.26}$$

- W_k is the within-cluster scatter matrix defined as

$$W_k = \sum_{j=1}^{k} \sum_{x \in C_j} (x - \mu_j)(x - \mu_j)^T, \tag{3.27}$$

  where $\mu_j = \frac{1}{n_j} \sum_{x \in C_j} x$ is cluster j's centroid, $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$ is the global mean, and n_j is the point count in cluster j.

This clustering evaluation metric is particularly suited for assessing convex and well-separated cluster structures, operating as an internal validation measure that requires no ground truth labels. The metric produces higher scores for better-defined clusters, with values increasing as cluster separation improves. Its computational efficiency is maintained at $\mathcal{O}(nk)$, making it practical for typical clustering analysis scenarios.

The DB [25] index measures the average similarity between clusters, where similarity is defined as the ratio of within-cluster distances to between-cluster distances. Lower values indicate better clustering. It is defined as:

$$\text{DB} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(\mu_i, \mu_j)} \right), \tag{3.28}$$

where:

- k is the number of clusters.
- σ_i is the average distance of all points in cluster i to centroid μ_i, given by

$$\sigma_i = \frac{1}{|C_i|} \sum_{x \in C_i} \|x - \mu_i\|. \tag{3.29}$$

- $d(\mu_i, \mu_j)$ is the distance between centroids μ_i and μ_j.

As an internal validation metric, this approach operates without requiring ground truth labels, evaluating cluster quality solely based on the intrinsic data structure. The metric produces lower values for better clustering configurations, with optimal scores approaching zero. While effective for assessing hyperspherical cluster formations, it has particular sensitivity to outliers in the dataset. The computational complexity scales as $\mathcal{O}(k^2 + nk)$, remaining efficient for typical clustering applications while accounting for both cluster count and dataset size.

The ARI [40] measures the similarity between two clustering solutions, adjusted for chance. It quantifies the agreement between the clustering result and the ground truth, defined as

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}, \tag{3.30}$$

where:

- n_{ij} is the number of objects in both class i and cluster j.
- $a_i = \sum_j n_{ij}$ is the number of objects in class i.
- $b_j = \sum_i n_{ij}$ is the number of objects in cluster j.
- n is the total number of objects.

This external validation metric requires ground truth labels for evaluation, producing scores bounded between -1 and 1. A score of 1 indicates perfect agreement between the clustering and ground truth, 0 represents the expected value for random clustering, and negative values suggest performance worse than chance. The metric incorporates an adjustment for chance agreements to provide a more reliable assessment. With a computational complexity of $\mathcal{O}(n + k^2)$, it remains efficient for practical applications while accounting for both sample size and number of clusters.

The NMI [72] quantifies the shared information between the clustering result and the ground truth, normalized to account for different numbers of clusters.

$$\text{NMI} = \frac{I(U;V)}{\sqrt{H(U)H(V)}}, \tag{3.31}$$

where:

- $I(U;V)$ is the mutual information between the clustering solutions U and V,

$$I(U;V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{n} \log \frac{n \cdot |U_i \cap V_j|}{|U_i| \cdot |V_j|}. \tag{3.32}$$

- $H(U)$ is the entropy of clustering U,

$$H(U) = -\sum_{i=1}^{|U|} \frac{|U_i|}{n} \log \frac{|U_i|}{n}. \tag{3.33}$$

- $H(V)$ is the entropy of clustering V.
- $|U_i|$ is the number of objects in cluster U_i.
- n is the total number of objects.

This approach requires ground truth labels for evaluation and produces scores bounded between 0 and 1, where higher values indicate better clustering agreement. The metric is particularly valuable for comparing clustering methods with different numbers of clusters, as it properly accounts for these variations. Additionally, it remains invariant to label permutations, ensuring consistent evaluation regardless of how cluster labels are assigned.

The computational complexity of $\mathcal{O}(n + k^2)$ makes it efficient for practical use, scaling linearly with the number of data points while considering the squared number of clusters.

Our two ranking systems prioritized these metrics differently:

- `Ranking_1 (Clustering Quality)`: CH → DB → ARI → NMI
- `Ranking_2 (True Label Separation)`: ARI → NMI → DB → CH

To enhance clustering effectiveness, we implemented an advanced reclustering approach:

1. Initial clustering with KMeans (3—10 clusters).
2. Metric evaluation (CH, DB, ARI, NMI).
3. Secondary clustering (3—6 sub-clusters) for refinement.
4. Hierarchical improvement of cluster purity.

3.5.4 Rule Mining Approach for 3-Class to Binary Classification

Rule mining is a data mining technique aimed at discovering meaningful patterns, associations, or relationships among variables in large datasets. Two widely used algorithms for this purpose are the Apriori algorithm and the Frequent Pattern Growth (FP-Growth) method. Apriori identifies frequent itemsets by generating candidates iteratively, while FP-Growth uses a compressed tree structure to avoid candidate generation, improving efficiency for dense datasets.

Apriori works by pruning infrequent itemsets level-wise using a minimum support threshold. FP-Growth constructs an FP-Growth to encode transactions compactly, then mines frequent itemsets via divide-and-conquer by building the FP-Growth through merging shared transaction prefixes and mining conditional pattern bases recursively to extract itemsets.

The core evaluation metrics for rule mining are:

- Support — The proportion of transactions containing a particular itemset.
- Confidence — The likelihood that item Y appears with item X, measured as

$$\text{Confidence}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}. \tag{3.34}$$

- Lift — The ratio of observed support to expected independence, given by

$$\text{Confidence}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}. \tag{3.35}$$

Cluster-specific pattern mining involves applying these algorithms within data subsets generated by clustering, enhancing rule relevance through intra-cluster similarities. While Apriori [1] suffers from computational bottlenecks due to candidate explosion, FP-Growth [13] mitigates this via its tree-based approach, making it preferable for large or dense datasets.

Our rule-based classification followed a two-step process:

1. Feature Binarization — Encoded values above the 80th percentile or below the 20th percentile as 1, with all others as 0.

2. Rule Mining — Applied Apriori and FP-Growth algorithms with parameters min_support=0.2, min_confidence=0.7, and max_length=3.

For each *unknown* transaction, classification proceeds in Algorithm 1.

Algorithm 1 Rule-Based Classification Algorithm

```
for all unknown transactions do
    Extract binary feature representation
    Check for matches with rules
    if matches illicit AND NOT matches licit then
        Classify as likely illicit
    else
        Classify as likely licit
    end if
end for
```

3.5.5 Semi-Supervised Learning

Semi-supervised learning is a machine learning paradigm that leverages both labeled and unlabeled data to improve model performance. It is particularly useful in scenarios where labeled data are scarce or expensive to obtain, but large quantities of unlabeled data are readily available.

In the context of *unknown* sample classification, semi-supervised methods assign pseudo-labels to unlabeled samples based on predictions from an initially trained model. One common strategy is self-training, where a classifier is first trained on the labeled data and subsequently used to label the unlabeled instances. These newly labeled instances are then iteratively added to the training set, allowing the model to refine its decision boundaries.

The general self-training process is:

- Train an initial classifier on the available labeled dataset.
- Predict pseudo-labels for the unlabeled samples.
- Select high-confidence predictions to add to the labeled dataset.
- Retrain the classifier with the expanded labeled dataset.

Self-training assumes that the classifier's most confident predictions are likely to be correct, an assumption that may not hold if the model is poorly initialized. Despite its simplicity, self-training can significantly improve classification performance, especially when the underlying distribution of labeled and unlabeled data is similar.

3.5.6 Integrated Classification Approach

Our final methodology combines clustering and classification results through:

1. Model Selection — Choosing best-performing models based on evaluation metrics.

2. Confidence-Based Integration — Weighting predictions by model confidence.
3. Threshold Application — Using predefined thresholds for final labels.
4. Conflict Resolution — Resolving disagreements between models.

This integrated approach leverages the strengths of both unsupervised and supervised techniques while mitigating their individual limitations, resulting in a robust classification system for transaction data.

3.5.7 Support Vector Machine (SVM) Classification

To complement the clustering and rule-based approaches, we implement SVM classification with probabilistic outputs. SVM provides a robust supervised learning alternative to analyze the labeled subset and predict labels for unlabeled transactions.

Our implementation uses the following pipeline:

- Feature Normalization — *StandardScaler* is applied to normalize the input features.
- Feature Selection — We use the top 10 most informative features selected through ANOVA F-test and XGBoost feature importance.
- Probabilistic Classification — The SVM classifier is configured to output class probabilities, which are interpreted through threshold-based logic for multi-class classification.

Classification decisions are guided by predefined probability thresholds:

- *Licit* — Probability ≥ 0.7.
- *Likely Licit* — $0.6 \leq$ Probability < 0.7.
- *Illicit* — Probability ≥ 0.7.
- *Likely Illicit* — $0.6 \leq$ Probability < 0.7.

This thresholding mechanism allows for a finer granularity in classification confidence and supports interpretability of the model's outputs.

3.5.8 Summary of Final Classification Methodology

Our comprehensive strategy integrates multiple paradigms:

1. Clustering Analysis — K-Means and GMM are used for discovering transaction patterns.
2. Rule-Based Classification — Apriori algorithm facilitates the classification of *unknown* transactions via mined association rules.
3. SVM Classification — A supervised probabilistic model trained on labeled data.

4. Semi-Supervised Learning — Self-training with Random Forest enhances generalization on unlabelled samples.
5. Model Comparison — Comparative evaluation of performance across all models and methods ensures the reliability of the final decision process.

This holistic design leverages the strengths of both unsupervised and supervised learning, resulting in a more accurate and robust classification framework for cryptocurrency transaction analysis.

The integrated approach allows for method comparison and selection based on:

- Data characteristics (balanced vs. imbalanced).
- Labeling availability (supervised vs. unsupervised scenarios).
- Performance requirements (precision vs. recall priority).

4 Experimental Evaluation

Chapter 4 presents an in-depth experimental evaluation of methods used for detecting illicit transactions in Bitcoin networks through machine learning and unsupervised learning approaches. Building upon the theoretical and technical foundations established in earlier chapters, this chapter systematically analyzes the effectiveness, efficiency, and interpretability of various models and techniques in real-world scenarios.

The chapter begins in Section 4.1 with a graphical and forensic analysis of transaction networks, highlighting structural patterns around suspicious nodes and subgraph dynamics characteristic of money laundering.

Section 4.2 evaluates dimensionality reduction techniques, using both feature selection and feature reduction to balance model accuracy with explainability and performance.

Section 4.3 investigates the impact of different instance sampling techniques for handling class imbalance in both binary and multi-class classification settings, emphasizing trade-offs between precision, recall, and computational cost.

Section 4.4 explores unsupervised classification methods and feature analysis, leveraging probabilistic models and statistical metrics to assess class separability without labeled data.

Section 4.5 presents clustering results using algorithms like KMeans and Gaussian Mixture Models (GMM), along with evaluation metrics that reflect clustering quality and alignment with true labels.

Section 4.6 presents semi-supervised learning and refined classification methods for categorizing unlabeled transactions, comparing approaches and their trade-offs between detection sensitivity and specificity through experimental results.

Section 4.7 compares integrated classification approaches (GMM, SVM, and rule-based methods), showing how method selection affects illicit detection rates (2.2—30%) through sensitivity-specificity trade-offs.

Section 4.8 synthesizes findings across all experimental approaches, discussing theoretical implications, practical limitations, and promising directions for future research in cryptocurrency transaction analysis.

This comprehensive experimental evaluation provides valuable insights into the practical challenges and performance considerations of cryptocurrency fraud detection, paving the way for more robust, explainable, and scalable detection frameworks.

4.1 Graphical Analysis on the Data

This section presents a forensic analysis of a transaction network centered on a highly suspicious Bitcoin node. Using graph-based methods, it reveals illicit patterns, network hierarchies, and structural indicators of laundering. The analysis highlights how illicit actors operate by examining node roles, degree centrality, and transaction paths.

We identify the top *illicit* node—characterized by high in- and out-degree—and analyze its attributes and role within the network.

Figure 4.1 provides a visual representation of the network, from which several assumptions and key insights are derived.

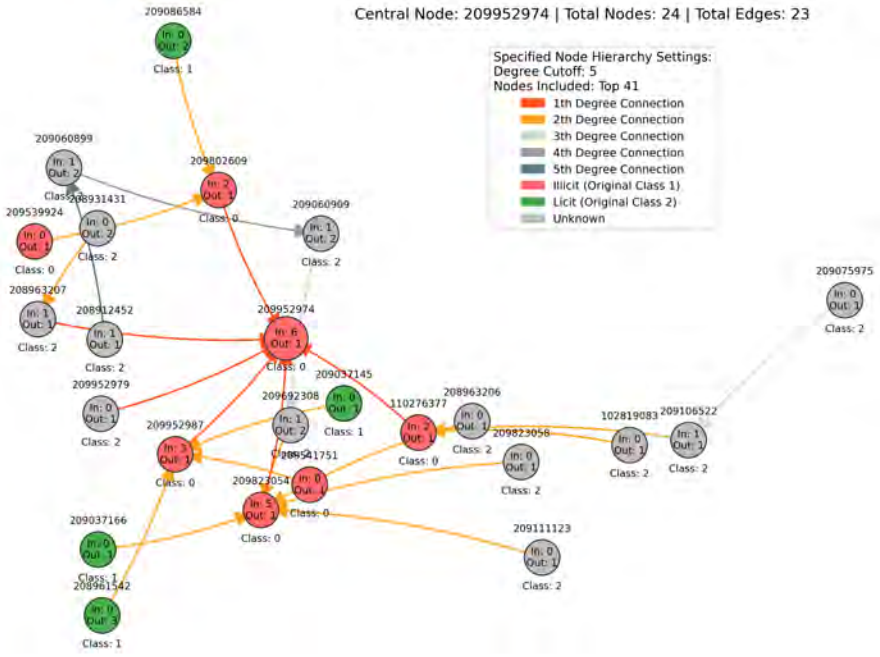


Figure 4.1: *Specified Node Transaction Network in-degree (Node Id 209952974)*

This specified node, Id `209952974`, has 6 in-degrees (→ receives from 6 other nodes) and 1 out-degree (→ sends to 1 node) and it belongs to Class 0 (*illicit* → marked red).

This centred node is highly targeted, suggesting it serves as a hub that receives transactions, typical of a collector or launderer node within financial crime networks.

The network has total nodes of 24, total edges of 23 and a degree cutoff of 5. The Node inclusion rule consist of the Top 41 closest (by distance to the centre).

The arrows are pointing into the central node, confirming the adjusted logic (finding inbound paths). Most connections lead to the central node, which aligns with expectations when tracing transaction sources toward a suspicious entity.

A lot of nodes are within 1st and 2nd degree, which are highlighted in bright orange and orange respectively. 1st-degree connections (direct sources to the central node) are the most prominent.

The majority of neighboring nodes around the Central Node are Class 0 (*illicit*) or *unknown* (Class 2). A few *licit* nodes (Class 1, green) are further out, mostly indirect contributors. This suggests the core of the subgraph is made up of questionable entities, with very limited licit interaction. Nodes `209539924`, `209802609`, `209952987`, `209541751`, `209823054`, and `110276377` are 1st or 2nd-degree with in/out degrees, also in Class 0, that may act as intermediaries, directly funding the central node.

In a money-laundering scenario, Node `209952974` may represent the sink account, a major receiver of funds (many incoming arrows), and others may be layers or mules. So, Nodes like this one may be priority targets for investigation.

Key suspicious nodes showed inbound-heavy patterns characteristic of money laundering: 209823054 (5:1 in-out ratio) appeared as a primary aggregation point, while 209952987 (3:1) and 209802609/110276377 (both 2:1) functioned as secondary consolidation nodes. All were flagged as *illicit*, suggesting fund layering before redistribution.

Green nodes such as `209071666`, `208961542`, `209037145`, and `209086584` are part of the network but they show only outward flow and they are relatively distant from the core hub. They could be entry points or legitimate entities unknowingly involved.

Many grey nodes are intermediaries with both incoming and outgoing links, which could imply obfuscation techniques (splitting/joining funds) or newly added entities not yet classified.

The *illicit* class is strongly centred around the core node. A clear layered hierarchy is visible, possibly indicative of money laundering and we can assume that the high in-degree contributors to the central node are likely collaborators or shell aggregators. A further investigation could involve temporal flow of transactions, overlaying amounts, timestamps, or even geolocation. Repeating similar patterns across different central nodes, may also be an interesting flow analysis.

Table 4.1 summarises the transaction network centred on this particular node.

Table 4.1: Transaction Network Summary Centred on Node `209952974`

Category	**Node**(s)	**Description / Insight**
Directionality	Arrows point to central node	In-degree analysis highlights inbound flows into the central node, spanning over 5 hops.
Path Bias	1st—3rd degree edges dominate	Suggests that most interactions occur within close proximity; further nodes diminish in volume.
Central Node (*illicit* Hub)	`209952974`	High in-degree (6), low out-degree (1). Main receiver node, classified as *illicit*.
High-Risk Aggregators	`209823054`, `209952987`	*illicit* nodes with high in-degree and multiple upstream connections, feeding the central node.

Continued on next page

Table 4.1 — continued from previous page

Category	Node(s)	Description / Insight
Licit Entry Points	209071666, 208961542, 209086584, 209037145	Classified as *Licit*, located on the periphery and potentially acting as first-hop fund sources.
Unknown Mixers / Relays	209060909, 209692308, others	*Class: Unknown*, moderate connections; could represent obfuscation or mixing behaviour.
Deep Connections (4th/5th Degree)	Various grey nodes	Extend the network's reach; may warrant further investigation to assess indirect ties to the illicit hub.

Figure 4.2 represents the out path network for the Node Id 209952974, centred network.

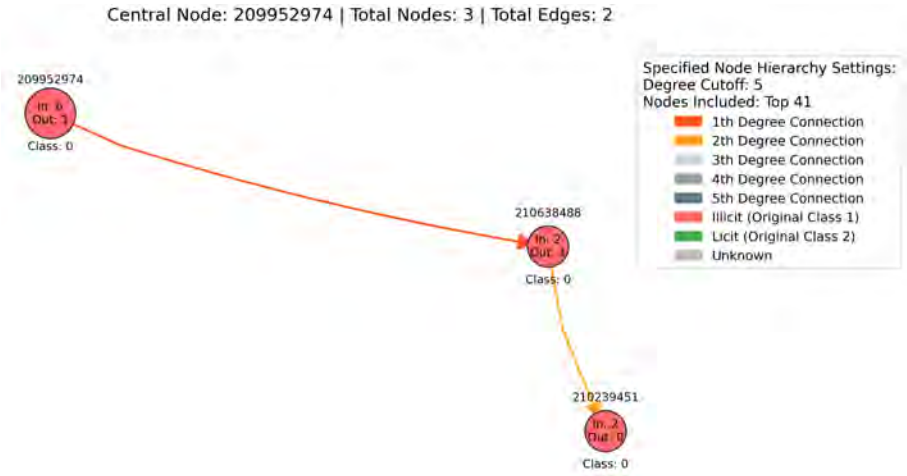


Figure 4.2: *Specified Node Transaction Network out-degree (Node Id 209952974)*

The arrows are pointing out the central node Id 209952974, confirming the adjusted logic (finding outbound paths), which aligns with expectations when tracing transaction sources away a suspicious entity.

The out-path network of node 209952974 reveals minimal dispersal behavior, reinforcing its role as a likely sink in the transaction flow.

Concerning to 1st and 2nd Degree Visibility, the immediate neighbouring node (210638488) is a 1st-degree connection, solely has an outgoing node (out-degree = 1), possibly a terminal or intermediary, and the next node (210239451) is a 2nd-degree connection, solely a receiver (in-degree = 2), indicating passive transactional behaviour. All of them belonging to Class 0 (*illicit*), suggesting a highly suspicious cluster with no *licit* or *unknown* actors currently visible.

Node 210638488 with 2 in-degrees and 1 out-degree acts as an intermediary node channelling transactions away from the central hub.

Node 210239451, with 2 in-degrees and 0 out-degrees, likely to be a downstream contributor in the flow of suspicious funds.

We can assume that this small, tightly connected cluster appears to be a clear example of illicit activity centring around node 209952974. The direct and indirect links to this

central node warrant further investigation, particularly with temporal and transactional data overlays.

Forensic examination of node 209952974 uncovered a prototypical laundering structure, with layered inbound transactions from illicit sources ($\geq$5 hops) and minimal dispersal ($k_{out} \leq 2$). This isolated subgraph, devoid of *licit* participation and dominated by Class 0/*unknown* entities, exemplifies the stealth design of illicit networks.

The observed dual-network dynamic presents a detection paradox: licit activity forms traceable, density-consistent pathways, while illicit behavior exploits decentralization and statistical insignificance. Despite comprising <2% of edges, illicit subgraphs exhibit disproportionate impact through distinctive topological signatures—particularly star-like inflow patterns and sink accounts—that evade conventional density-based detection methods.

4.2 Dimensionality Reduction Techniques

This section explores dimensionality reduction to improve model explainability and visualization while preserving classification performance. It evaluates FS and FR methods including Analysis of Variance (ANOVA), SHapley Additive exPlanations (SHAP), PCA, and UMAP across binary and multi-class scenarios, comparing original and reduced datasets to assess accuracy-interpretability trade-offs.

4.2.1 Baseline supervised classification results

Table 4.2 shows the results of classifiers on the two-class original dataset (Class 1 and Class 2). The best accuracy is in boldface. We use 10-fold cross-validation for the assessment.

Table 4.2: Model comparison for 2-class classification.

Model	Accuracy	Precision	Recall	F1 Score	P. Time (s)	Efficiency (Accuracy/Time)
Naive Bayes	0.6333	0.9198	0.6333	0.7064	4.13	0.1534
KNN	0.9753	0.9748	0.9753	0.975	0.66	1.4706
LinearSVC	0.9726	0.9718	0.9726	0.9718	192.06	0.0051
Decision Tree	0.9798	0.9801	0.9798	0.9799	225.37	0.0043
XGBoost	**0.9928**	**0.9928**	**0.9928**	**0.9927**	**298.1**	**0.0033**
SVM	0.9742	0.9735	0.9742	0.9734	612.54	0.0016
LightGBM	0.9919	0.9919	0.9919	0.9918	969.44	0.001
Random Forest	0.99	0.9901	0.99	0.9898	1040.6	0.001
CatBoost	0.9917	0.9918	0.9917	0.9916	1304.82	0.0008
Logistic Regression	0.9445	0.9403	0.9445	0.9406	1920.06	0.0005
MLP	0.9816	0.9814	0.9816	0.9815	2930.41	0.0003

XGBoost achieves the highest accuracy while K-Nearest Neighbours (KNN) shows the highest efficiency (Accuracy/Training Time). Figure 4.3 presents XGBoost binary classification analysis including confusion matrix, ROC curve, and PR-curve for the 2-class case.

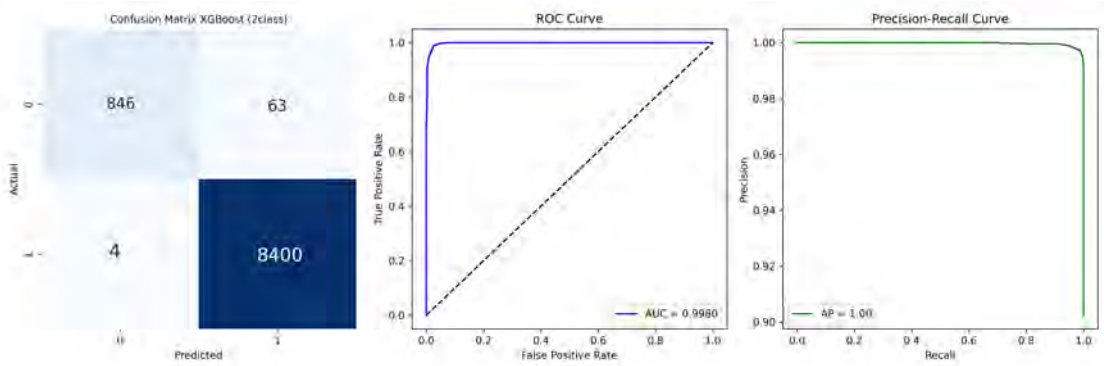


Figure 4.3: *XGBoost performance on the 2-class dataset: confusion matrix, ROC curve, and PR curve.*

We observe that:

- Class 1 shows 8,400 correct predictions and only 4 misclassifications as Class 0.
- Class 0 has 846 correct predictions, with 63 instances misclassified as Class 1.
- The model has nearly perfect Recall for Class 1 (8,400/8,404 = 99.95%) and very high Precision (8,400/8,463 = 99.26%).

Regarding the ROC curve analysis, we have the following:

- The ROC curve shows AUC of 0.9980, with near-perfect discriminative ability.
- The curve rises almost vertically from the origin, suggesting the model achieves very high true positive rates whith extremely low false positive rates.
- This steep ascent indicates an excellent threshold selection range for optimizing model performance.

The Precision-Recall curve analysis shows that:

- The Precision-Recall curve demonstrates excellent performance with an Average Precision (AP) of 0.99.
- Precision remains at 1.0 across nearly the entire range of Recall values, only dropping slightly at the highest Recall levels.

The XGBoost model is particularly effective at identifying Class 1 instances, with almost no false negatives. Despite the presence of class imbalance, the model maintains excellent performance across evaluation metrics. The AUC of 0.9980 and AP of 1.00 suggest a model that would be extremely reliable in production environments. Table 4.3 shows the summary results of the ML methods for the 3-class original dataset.

Table 4.3: Model comparison for 3-class classification. The best accuracy is in boldface.

Model	Accuracy	Precision	Recall	F1 Score	P. Time (s)	Efficiency (Accuracy/Time)
Naive Bayes	0.3095	0.8052	0.3095	0.42	6.63	0.0467
Decision Tree	0.923	0.9237	0.923	0.9233	588.92	0.0016
KNN	0.9023	0.8997	0.9023	0.8984	1.28	0.7068
XGBoost	**0.9578**	**0.9573**	**0.9578**	**0.9573**	**925.94**	**0.001**
CatBoost	0.9526	0.9519	0.9526	0.9519	1334.09	0.0007
LinearSVC	0.8509	0.8295	0.8509	0.8281	1602.05	0.0005
Random Forest	0.9523	0.9519	0.9523	0.9512	1730.69	0.0006
LightGBM	0.9507	0.9501	0.9507	0.95	2033.22	0.0005
Logistic Regression	0.8533	0.8306	0.8533	0.8321	3225.23	0.0003
SVM	0.9009	0.8998	0.9009	0.8875	6718.72	0.0001
MLP	0.919	0.9173	0.919	0.917	15742.4	0.0001

The model with highest efficiency as well as the highest accuracy is XGBoost. Figure 4.4 depicts the analysis of the XGBoost classification results regarding the confusion matrix, ROC curve, and PR-curve, for the three-class case.

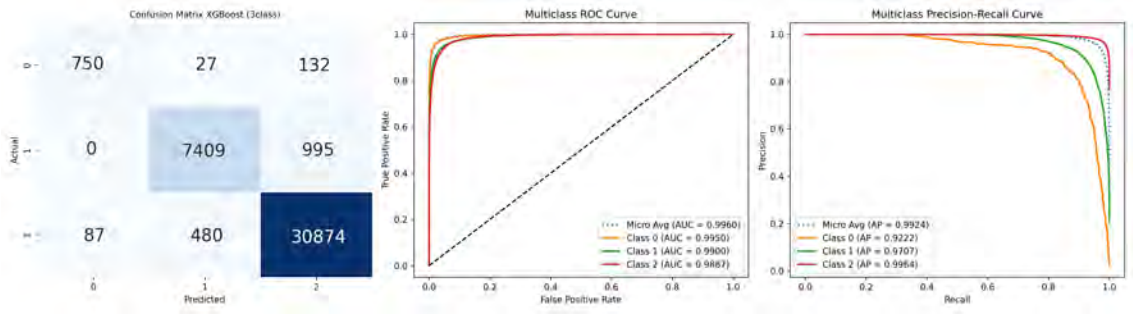


Figure 4.4: *XGBoost performance on the 3-class dataset: confusion matrix, ROC curve, and PR curve.*

On the confusion matrix, we observe that:

- Class 2 shows the best performance with 30,874 correct predictions and relatively few misclassifications (480 as Class 1 and 87 as Class 0).
- Class 1 has good performance with 7,409 correct predictions, though with some misclassifications to Class 2 (995).
- Class 0 has 750 correct predictions, with misclassifications primarily to Class 2 (132).

Regarding the ROC curve analysis, we have:

- All classes show AUC scores above 0.98, indicating strong discriminative power.
- Micro average AUC is 0.9960, suggesting excellent overall classification performance.
- Class 0 has the highest individual AUC (0.9950), followed by Class 1 (0.9900) and Class 2 (0.9887).
- All ROC curves rise steeply at low false positive rates, indicating the model achieves high true positive rates while maintaining low false positive rates.

In the PR curve analysis, we observe that:

- The curves show strong performance across all classes.
- Class 2 shows the best Precision-Recall trade-off with Average Precision (AP) of 0.9964.
- Class 0 shows degradation in Precision at higher Recall values (AP = 0.9222).
- Class 1 maintains good Precision up to very high Recall values (AP = 0.9707).
- The micro-average AP is 0.9924, confirming excellent overall performance.

The XGBoost model provides adequate performance for the 3-class classification problem. The model is particularly effective at identifying Class 2 instances. Despite class imbalance, the model maintains strong performance across all classes with high AUC and AP scores.

4.2.2 Feature Selection

We now assess the use of FS techniques over the dataset. First, we consider the 2-class dataset and then we move to the 3-class dataset assessment. Table 4.4 compares the top features selected by different methods for the 2-class dataset.

Table 4.4: Comparison of features selected by different methods (2-class dataset). Top 10 features selected by each method showing limited overlap (3 common features).

Selected Features (ANOVA)	**Selected Features (XGBOOST)**
tx_ftr_51	tx_ftr_30
tx_ftr_52	tx_ftr_89
tx_ftr_53	tx_ftr_58
tx_ftr_54	tx_ftr_79
tx_ftr_88	agg_ftr_69
tx_ftr_89	agg_ftr_67
tx_ftr_90	tx_ftr_54
agg_ftr_48	tx_ftr_52
agg_ftr_56	tx_ftr_45
agg_ftr_60	tx_ftr_4

Figure 4.5 shows the distribution of SHapley Additive exPlanations (SHAP) [53] values across multiple features. It helps identify which features most significantly impact the model's predictions and how different feature values contribute to those predictions.

The color gradient from blue to pink/magenta indicates the feature value magnitude, with blue representing low feature values and pink/magenta representing high feature values. We observe that:

- Features like *tx_ftr_52*, *tx_ftr_58*, and *tx_ftr_1* show wide distributions, suggesting stronger influence on the model's predictions.
- Many features display bimodal or multimodal distributions, indicating different impacts depending on the exact feature value.
- The color pattern reveals relationships between feature values and their contributions – for instance, high values (pink) of *tx_ftr_2* tend to have negative SHAP values, while high values of *tx_ftr_52* tend to have positive SHAP values.

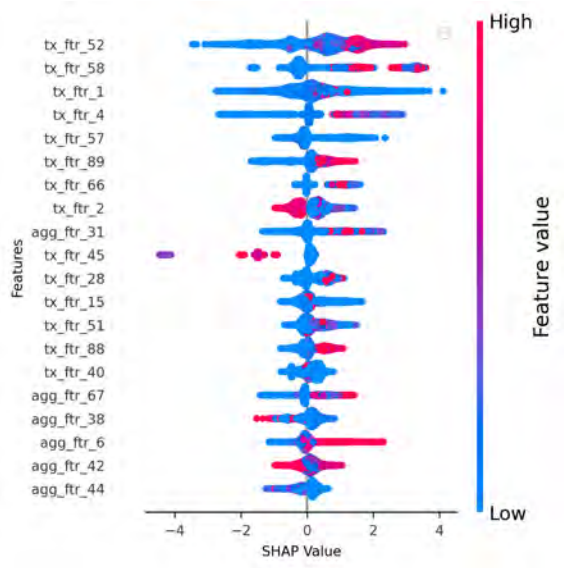


Figure 4.5: *XGBoost Explainability for the 2-class dataset*

Table 4.5 compares Top 10 features selected by different methods (3-class dataset), showing limited overlap (3 common features).

Table 4.5: Comparison of features selected by different methods (3-class dataset).

Selected Features (ANOVA)	**Selected Features (XGBOOST)**
tx_ftr_51	tx_ftr_19
tx_ftr_52	tx_ftr_59
tx_ftr_53	tx_ftr_84
tx_ftr_54	tx_ftr_75
tx_ftr_58	tx_ftr_58
tx_ftr_59	tx_ftr_33
tx_ftr_60	tx_ftr_4
tx_ftr_64	agg_ftr_30
tx_ftr_65	tx_ftr_52
tx_ftr_66	tx_ftr_47

Figure 4.6 shows a stacked bar chart of SHAP value distributions across three classes for multiple features, revealing both overall feature importance and class-specific impacts in multi-class classification.

Each bar is segmented into three colors representing different classes: Blue segments represent Class 0; Pink/Magenta segments represent Class 1; Olive green segments represent Class 2. The features are sorted by their total SHAP value contribution, with *tx_ftr_52* showing the highest overall impact. The key observations on these results are as follows:

- The *tx_ftr_52* feature has the largest overall SHAP value and shows significant contributions to all three classes, particularly Classes 1 and 2.
- The *tx_ftr_1* feature ranks second in importance and predominantly influences Class 0, with some impact on Classes 1 and 2.
- Features found lower in the chart (e.g., *tx_ftr_78*, *agg_ftr_42*) have smaller overall SHAP values but still contribute to the model's predictions across classes.

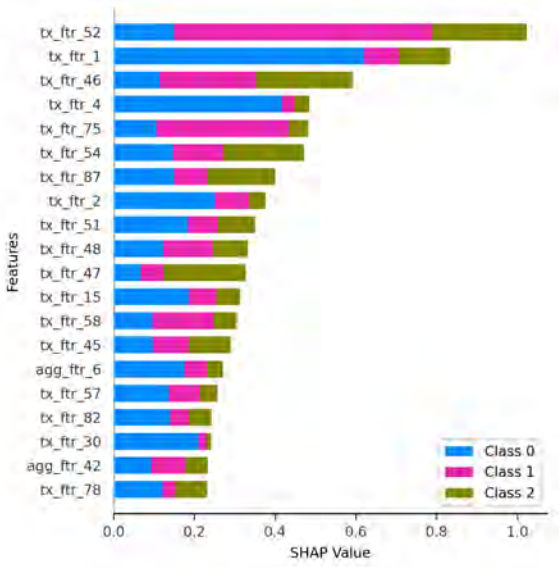


Figure 4.6: *XGBoost Explainability for the 3-class dataset*

- The relative proportions of each color within a bar indicate how a feature differentially impacts predictions for each class.

4.2.3 Feature Reduction

Figure 4.7 consists of two complementary plots analyzing PCA results over the 3-class dataset. The individual and cumulative explained variance shows both individual con-

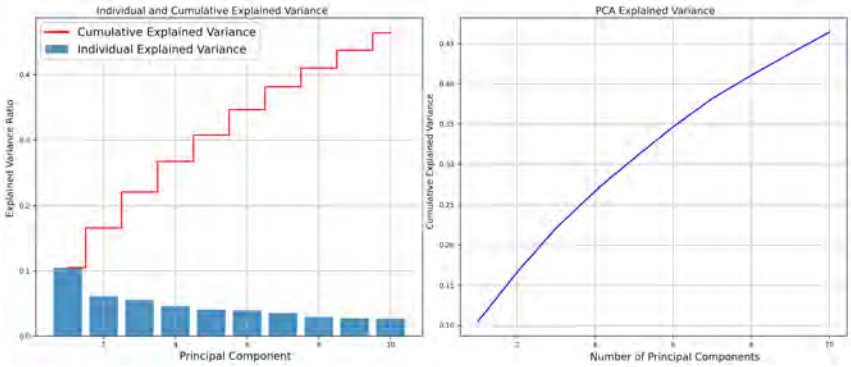


Figure 4.7: *PCA Explained variance for the 3-class dataset*

tribution (blue bars) and cumulative explained variance (red step line) for the first 10 principal components. The first principal component captures approximately 10% of the total variance. Subsequent components contribute progressively less variance, exhibiting a typical elbow pattern and the cumulative explained variance reaches approximately 45% by the 10th component.

This PCA analysis reveals a dataset with high intrinsic dimensionality. No single component or small subset of components captures the majority of variance. Even with 10 components, less than half of the total variance is explained. This suggests a complex, high-dimensional data structure where information is distributed across many features.

Figure 4.8 displays a UMAP visualization of the dataset, mapping high-dimensional data to a 2D representation.

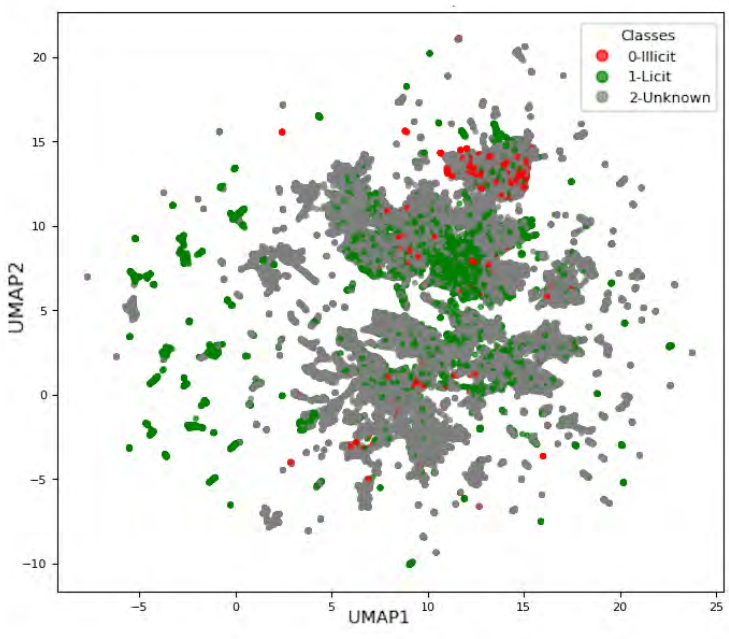


Figure 4.8: *UMAP Projection for the 3-class dataset*

The data points are colored by class: gray (2-*unknown*), green (1-*licit*), and red (0-*illicit*). The projection shows complex, overlapping clusters with some visible structure. Class 2 (*unknown*, gray) appears most abundant and widely distributed. Class 1 (*licit*, green) shows partial separation in some regions but considerable overlap with Class 2. Class 0 (*illicit*, red) has fewer points and tends to overlap with both other classes.

The UMAP visualization shows a significant overlap between classes. Some regions show higher density of specific classes, with partial discriminative power in the features. The presence of small clusters and substructures suggests potential subgroups within each class. The manifold structure appears complex, with multiple connected regions. While complete class separation is difficult, there exist distinguishable patterns that ML algorithms might leverage. Some outlier points and smaller clusters appear at the periphery, potentially representing unusual or anomalous cases.

4.2.4 Classification results using XGBoost

We now assess the evaluation of classification with reduced dimensionality datasets, using XGBoost. Table 4.6 reports the results for the two-class version dataset while Table 4.7 does a similar evaluation for the three class version of the dataset.

Table 4.6: 2-Class reduced dimensionality with XGBoost. The best global accuracy is in boldface. The best accuracy with dimensionality reduction is highlighted in blue.

Dataset	Accuracy	Precision	Recall	F1 Score	P. Time (s)	Efficiency
2class_anova	0.9821	0.9818	0.9821	0.9816	0.37	2.6687
2class_anova_pca	0.9733	0.9725	0.9733	0.9724	0.35	2.7613
2class_anova_umap	0.9566	0.9569	0.9586	0.9574	0.22	4.3178
2class_original	0.9928	0.9928	0.9928	0.9927	2.93	0.3386
2class_original_scaled	**0.9931**	**0.9932**	**0.9931**	**0.9930**	**2.81**	**0.3533**
2class_pca	0.9715	0.9707	0.9715	0.9706	0.43	2.2647
2class_umap	0.9553	0.9536	0.9553	0.9542	0.28	3.4115
2class_xgboostfs	0.9839	0.9836	0.9839	0.9836	0.44	2.2515
2class_xgboostfs_pca	0.9802	0.9798	0.9802	0.9798	0.48	2.0252
2class_xgboostfs_umap	0.9764	0.9759	0.9764	0.9755	0.23	4.2638

Table 4.7: 3-Class reduced dimensionality with XGBoost. The best global accuracy is in boldface. The best accuracy with dimensionality reduction is highlighted in blue.

Dataset	Accuracy	Precision	Recall	F1 Score	P. Time (s)	Efficiency
3class_anova	0.9071	0.9045	0.9071	0.9036	3.97	0.2286
3class_anova_pca	0.8871	0.8845	0.8871	0.8789	4.00	0.2216
3class_anova_umap	0.8739	0.8700	0.8739	0.8615	3.24	0.2698
3class_original	**0.9578**	**0.9573**	**0.9578**	**0.9573**	**22.44**	**0.0427**
3class_original_scaled	0.9576	0.9571	0.9576	0.9571	21.62	0.0443
3class_pca	0.8867	0.8830	0.8867	0.8798	4.27	0.2078
3class_umap	0.8657	0.8599	0.8657	0.8527	3.04	0.2852
3class_xgboost	0.9259	0.9244	0.9259	0.9238	3.76	0.2464
3class_xgboostfs_pca	0.8961	0.8939	0.8961	0.8901	3.92	0.2286
3class_xgboostfs_umap	0.8674	0.8646	0.8674	0.8547	2.92	0.2966

These DR techniques do not improve the classification results, as compared to the use of the original dimensionality. However, data scaling improves the results of XGBoost, for the 2-class case, as compared with the ones reported in Table 4.2.

Among the classification models considered, XGBoost provided the best results, performing better on the 2-class dataset as compared to the 3-class dataset. The reported SHAP values showed that a few features contribute decisively for the model's predictions across classes. Combining insights from the data visualizations, we conclude that the dataset exhibits high intrinsic dimensionality, as shown by the PCA analysis. The use of dimensionality reduction techniques needs further investigation.

4.3 Instance Sampling Techniques

This section examines class imbalance strategies, comparing traditional and hybrid sampling techniques including Random Oversampling, SMOTE, Tomek Links, and SMOTE+ENN. The analysis evaluates their effects on classification performance across different models (e.g., XGBoost, Random Forest) in both 2-class and 3-class setups, providing guidance for selecting optimal strategies based on detection goals and computational constraints.

4.3.1 Analysis for 2-Class Imbalanced Data

Figure 4.9 displays the resampled balanced dataset result.

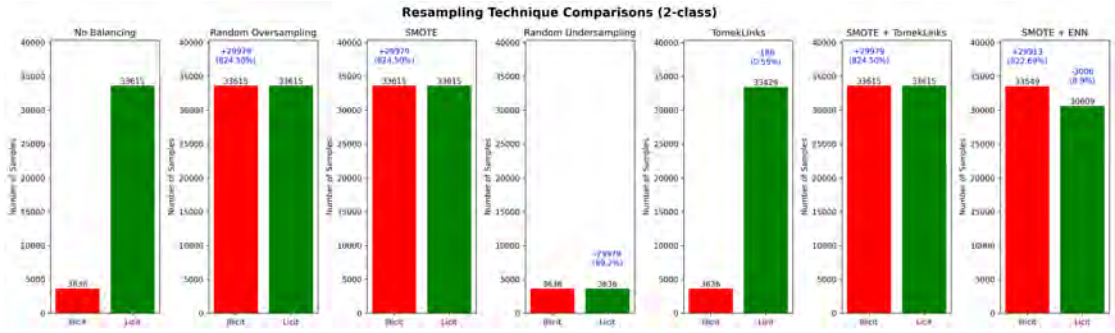


Figure 4.9: *Instance Sampling Techniques for 2-Class Imbalanced Data*

Our analysis of imbalance handling techniques on the 2-class dataset reveals significant class distribution differences and performance impacts. The original dataset shows severe imbalance with only 3,636 *illicit* samples (approximately 1:9 ratio). The unmodified dataset serves as our baseline, where classification models typically exhibit majority class bias, potentially misclassifying critical minority samples. This preserves original data characteristics but often leads to suboptimal minority class performance, particularly Recall.

4.3.2 Implications for Model Selection

The choice of the instance sampling technique significantly impacts model training and performance metrics. For applications where missing minority class instances carries high costs (such as fraud detection), hybrid approaches like ENN+SMOTE offer better balance between class representation and decision boundary clarity. Conversely, when computational efficiency is prioritized, simpler techniques like random undersampling may be appropriate despite their information loss. Our results suggest that no single instance sampling technique is universally superior; rather, the selection should be guided by domain-specific requirements, computational constraints, and the relative importance of Precision versus Recall in the specific application context. Table 4.8 shows the advantages and drawbacks of Imbalance Handling Techniques.

Table 4.8: Comparison of Instance Sampling Techniques for Imbalanced Data (2-class)

Technique	**Class Distribution**	**Advantages**	**Drawbacks**
No Balancing	*illicit* (3,636) vs *licit* (33,615) ∼1:9 ratio	• Preserves original data distribution • No risk of artificial patterns • Useful baseline for comparison	• Models biased toward majority class • Poor performance on minority class • Higher risk of missing critical cases
Random Oversampling	*illicit* (33,615) vs *licit* (33,615) 1:1 ratio	• Simple implementation • Maintains all original data • No information loss • Balanced class distribution	• Risk of overfitting to minority class • Creates exact duplicates • May amplify noise in minority class

Continued on next page

Table 4.8 — continued from previous page

Technique	Class Distribution	Pros	Cons
SMOTE	*illicit* (33,615) vs *licit* (33,615) 1:1 ratio	• Creates synthetic samples • Generates samples between observations • Reduces overfitting compared to random oversampling	• May generate unrealistic samples • Can increase class overlap • Not ideal for high-dimensional spaces
Random Undersampling	*illicit* (3,636) vs *licit* (3,636) 1:1 ratio	• Simple implementation • Reduces computational cost • Eliminates class imbalance • Can help address majority class noise	• Significant information loss • Discards ∼90% of majority data • Risk of removing important patterns
Tomek Links	*illicit* (3,636) vs *licit* (33,429) ∼1:9 ratio	• Removes only borderline/noisy majority samples • Preserves most majority class information • Clarifies decision boundaries	• Minimal effect on overall imbalance • Limited improvement for severe imbalances • Still leaves model susceptible to majority bias
SMOTE + Tomek Links	*illicit* (33,615) vs *licit* (33,615) 1:1 ratio	• Combines benefits of both techniques • Addresses underrepresentation and class overlap • Creates cleaner decision boundaries	• Computationally more expensive • Two-step process increases complexity • May create unnatural synthetic samples
SMOTE + ENN	*illicit* (33,549) vs *licit* (30,609) ∼1.1:1 ratio	• More aggressive cleaning than Tomek Links • Removes both class noisy samples • Creates cleaner class boundaries • Reduces both types of noise	• Computational overhead • Removed some minority samples • Risk of removing too many samples

4.3.3 Importance of Averaging Performance Metrics

Calculating averages is essential to summarize performance metrics across multiple datasets for each combination of `Technique` and `Model`. When evaluating results from diverse datasets, raw data often contains fragmented entries, making direct comparisons impractical. By aggregating metrics using averaging, it provides a central tendency—smoothing out dataset-specific variability and offering a generalizable view of a model's typical performance. This averaging enables a fair equivalence comparison between techniques and models to identify top performers. Additionally, counting Datasets, ensures transparency by revealing how many datasets contributed to each average, guarding against conclusions based on small or unreliable sample sizes.

Table 4.9 summarizes performance average metrics (2-class).

Table 4.9: Performance average Metrics (2-class)

Technique	Avg Precision	Avg Recall	Avg F1-Score	Avg Macro F1
XGBoost (xgb)				
Random Oversampling	**0.9931**	**0.9931**	**0.9931**	**0.9803**
No Balancing	0.9930	0.9930	0.9929	0.9795
Tomek Links	0.9927	0.9927	0.9926	0.9788

Continued on next page

Table 4.9 – continued from previous page

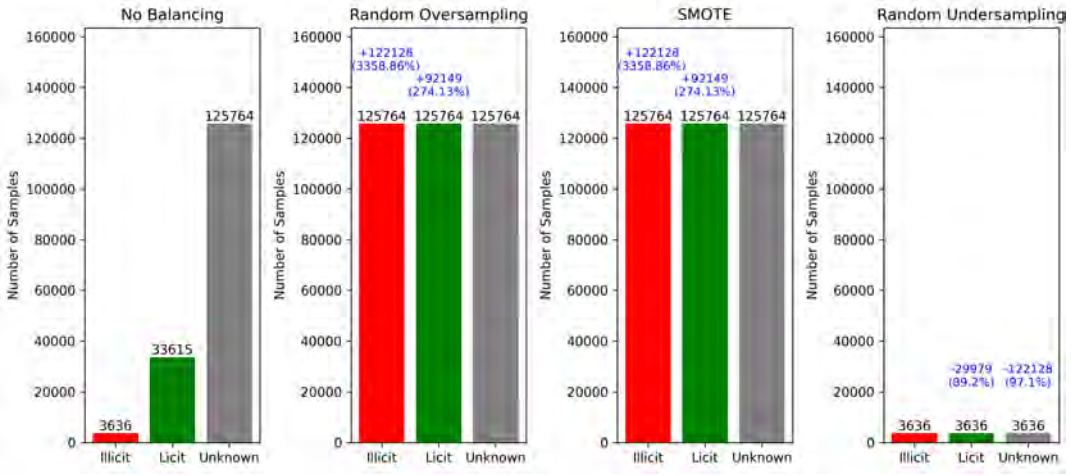
Technique	Avg Precision	Avg Recall	Avg F1-Score	Avg Macro F1
SMOTE	0.9926	0.9926	0.9926	0.9787
SMOTE + Tomek Links	0.9922	0.9923	0.9922	0.9776
SMOTE + ENN	0.9892	0.9891	0.9892	0.9694
Random Undersampling	0.9764	0.9713	0.9727	0.9265
Random Forest (rf)				
SMOTE	**0.9920**	**0.9920**	**0.9919**	**0.9766**
SMOTE + Tomek Links	0.9919	0.9919	0.9918	0.9765
Random Oversampling	0.9906	0.9907	0.9905	0.9726
No Balancing	0.9900	0.9899	0.9897	0.9700
SMOTE + ENN	0.9899	0.9900	0.9899	0.9713
Tomek Links	0.9897	0.9896	0.9893	0.9690
Random Undersampling	0.9842	0.9834	0.9836	0.9543
LightGBM (lgbm)				
Tomek Links	**0.9908**	**0.9908**	**0.9908**	**0.9739**
No Balancing	0.9904	0.9904	0.9904	0.9728
SMOTE	0.9856	0.9849	0.9851	0.9584
SMOTE + Tomek Links	0.9856	0.9848	0.9850	0.9583
Random Oversampling	0.9838	0.9821	0.9826	0.9519
SMOTE + ENN	0.9794	0.9767	0.9775	0.9383
Random Undersampling	0.9687	0.9570	0.9601	0.8961
Decision Tree (dt)				
Random Oversampling	**0.9833**	**0.9832**	**0.9833**	**0.9527**
No Balancing	0.9822	0.9820	0.9821	0.9494
Tomek Links	0.9813	0.9810	0.9812	0.9469
SMOTE	0.9763	0.9750	0.9755	0.9317
SMOTE + Tomek Links	0.9761	0.9747	0.9752	0.9311
SMOTE + ENN	0.9721	0.9684	0.9696	0.9170
Random Undersampling	0.9571	0.9369	0.9427	0.8549

The analysis of the classification results reveals several significant findings. Random Oversampling and XGBoost show top performance with XGBoost emerging as the best classifier overall achieving 0.9931 for Precision, Recall, and F1-Score. Most rebalancing techniques outperform the baseline "No Balancing" approach across all four classifiers, indicating that addressing class imbalance effectively improves model performance in this binary classification task. Random Undersampling shows the lowest performance metrics, particularly with Decision Tree where it achieves only 0.8549 for Macro F1. SMOTE combined with Tomek Links shows consistent performance across algorithms, suggesting this hybrid approach effectively handles class imbalance without the drawbacks of individual methods. The gap between the best and worst techniques is more pronounced in Decision Tree models (difference of approximately 0.1 in Macro F1) compared to XGBoost. Additionally, the minimal difference between Precision and Recall values for most techniques indicates balanced prediction capabilities for both classes, which is particularly important in imbalanced classification scenarios.

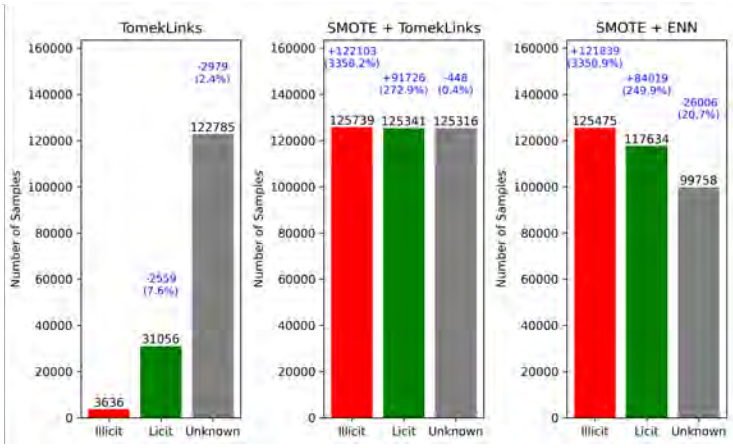
4.3.4 Analysis for 3-Class Imbalanced Data

Figure 4.10 depicts the use of instance sampling techniques applied to a 3-class dataset (original and scaled). We notice significant differences in how these methods handle multiple imbalanced classes. The dataset now includes *illicit*, *licit*, and *unknown* classes with a

severe imbalance among them.



(a) *Instance sampling Techniques: No Balancing, Random Oversampling, SMOTE, and Random Undersampling*



(b) *Instance sampling Techniques: Tomek Links, SMOTE+Tomek Links, and SMOTE+ENN*

Figure 4.10: *Instance Sampling Techniques for 3-Class Imbalanced Data*

Table 4.10 presents a comparative overview of instance sampling techniques tailored for imbalanced multi-class classification, specifically in the context of three classes (3-class): *illicit*, *licit*, and *unknown*. It highlights each method's impact on class distribution and outlines key advantages and drawbacks. Techniques such as Random Oversampling and SMOTE achieve perfect balance but come with risks like overfitting or unrealistic sample generation. In contrast, methods like Random Undersampling drastically reduce data size to balance classes but at the cost of significant information loss. Hybrid approaches (e.g., SMOTE + ENN) attempt to strike a balance between cleaning and augmentation, though they incur higher computational overhead.

Table 4.10: Comparison of Instance Sampling Techniques for Imbalanced Data (3-class)

Technique	Class Distribution	Advantages	Drawbacks
No Balancing	*illicit* (3,636) vs *licit* (33,615) vs *unknown* (125,764) ∼1:9:35 ratio	• Preserves original data distribution • No synthetic data or information loss • Maintains natural class proportions	• Extreme imbalance biases models toward dominant class • Minority class at high risk of misclassification • Poor performance metrics for minority class
Random Oversampling	*illicit* (125,764) vs *licit* (125,764) vs *unknown* (125,764) 1:1:1 ratio	• Achieves perfect balance across classes • Preserves all original data • Straightforward implementation	• Massive duplication of minority classes (∼35× for *illicit*) • Severe risk of overfitting • Memory intensive due to increased dataset size
SMOTE	*illicit* (125,764) vs *licit* (125,764) vs *unknown* (125,764) 1:1:1 ratio	• Creates synthetic samples rather than duplicates • Achieves perfect balance • Can generate diverse minority samples	• Multi-class SMOTE may create unrealistic samples • Higher risk of class overlap • Synthetic samples may cross class boundaries
Random Undersampling	*illicit* (3,636) vs *licit* (3,636) vs *unknown* (3,636) 1:1:1 ratio	• Achieves perfect balance at lowest count • Drastically reduces computational requirements • Simple implementation	• Discards ∼97% of majority class data • Extreme information loss • High risk of removing critical patterns
Tomek Links	*illicit* (3,636) vs *licit* (31,056) vs *unknown* (122,785) ∼1:8.5:34 ratio	• Improves class boundaries • Minimal data modification • Preserves most original data	• Maintains severe imbalance • Limited effectiveness with three highly imbalanced classes • Minimal improvement in class separation
SMOTE + Tomek Links	*illicit* (125,739) vs *licit* (125,341) vs *unknown* (125,316) ∼1:1:1 ratio	• Combines synthetic generation with boundary cleaning • Creates nearly perfect balance • Improves class separation	• Computationally expensive • Complex implementation for three classes • Risk of generating samples in ambiguous regions
SMOTE + ENN	*illicit* (125,475) vs *licit* (117,634) vs *unknown* (99,758) ∼1.25:1.18:1 ratio	• Most aggressive noise removal • Cleans boundaries across all classes • Reduces noise while maintaining representation	• Creates slight reverse imbalance • Highest computational cost • Removes significant number of samples

Table 4.11 summarizes the average performance metrics—Precision, Recall, F1-Score, and Macro F1—across various instance sampling techniques applied to both original and scaled versions of the dataset in a 3-class classification setting. The comparison is organized by model type (XGBoost, Random Forest, LightGBM, and Decision Tree), highlighting how different techniques influence model performance. Notably, XGBoost achieves its highest Macro F1 without instance sampling, while Random Forest performs best with Random Oversampling. Light Gradient Boosting Machine (LightGBM) shows improved balance with Tomek Links, and Decision Tree achieves optimal scores with the original unbalanced data. The table underscores how the effectiveness of instance sampling strategies can vary significantly depending on the Classifier.

Table 4.11: Performance Comparison of Instance Sampling Techniques (3-Class)

Technique	Avg Precision	Avg Recall	Avg F1-Score	Avg Macro F1
XGBoost (xgb)				
No Balancing	**0.9572**	**0.9577**	**0.9572**	**0.9143**
Tomek Links	0.9559	0.9564	0.9558	0.9138
SMOTE	0.9531	0.9531	0.9530	0.9018
SMOTE + Tomek Links	0.9531	0.9531	0.9530	0.9002
Random Oversampling	0.9509	0.9470	0.9481	0.8878
SMOTE + ENN	0.9433	0.9402	0.9412	0.8778
Random Undersampling	0.9246	0.9078	0.9125	0.8116
Random Forest (rf)				
Random Oversampling	**0.9539**	**0.9544**	**0.9540**	**0.9121**
SMOTE	0.9505	0.9502	0.9504	0.9067
SMOTE + Tomek Links	0.9502	0.9498	0.9500	0.9063
No Balancing	0.9518	0.9522	0.9512	0.8966
Tomek Links	0.9493	0.9497	0.9485	0.8933
SMOTE + ENN	0.9375	0.9328	0.9342	0.8738
Random Undersampling	0.9190	0.9049	0.9090	0.8101
LightGBM (lgbm)				
Tomek Links	**0.9409**	**0.9334**	**0.9356**	**0.8540**
No Balancing	0.9392	0.9304	0.9329	0.8530
SMOTE + Tomek Links	0.9218	0.8985	0.9046	0.7973
SMOTE	0.9215	0.8978	0.9040	0.7967
SMOTE + ENN	0.9153	0.8828	0.8910	0.7725
Random Oversampling	0.9092	0.8573	0.8692	0.7407
Random Undersampling	0.9040	0.8429	0.8595	0.7046
Decision Tree (dt)				
No Balancing	**0.9241**	**0.9244**	**0.9242**	**0.8528**
Tomek Links	0.9226	0.9233	0.9229	0.8494
Random Oversampling	0.9206	0.9207	0.9206	0.8476
SMOTE	0.9156	0.9116	0.9131	0.8307
SMOTE + Tomek Links	0.9154	0.9114	0.9129	0.8302
SMOTE + ENN	0.9141	0.9040	0.9072	0.8110
Random Undersampling	0.8688	0.8118	0.8287	0.6711

4.3.5 Overall Recommendations for 3-Class Problems

For critical detection with sufficient resources, SMOTE + Tomek Links offers optimal balance between sample creation and boundary cleaning, while SMOTE + ENN provides cleaner boundaries with some data loss when noise reduction is crucial. Random Undersampling may be necessary with limited resources but causes significant information loss, whereas Tomek Links makes minimal modifications but may inadequately address extreme imbalance. For general multi-class scenarios, SMOTE provides good balance but benefits from boundary cleaning when class overlap exists.

The 3-class scenario presents additional challenges, particularly with heavy class dominance. Sampling technique selection should consider class importance, computational resources, and classification objectives.

Experimental results show XGBoost consistently outperforms others, achieving highest average Precision (0.9572) without balancing. Tomek Links slightly reduces Precision while maintaining strong performance, whereas SMOTE variants perform moderately and SMOTE + ENN with random undersampling cause noticeable drops.

Random Forest exhibits stable performance across techniques, with best results from random oversampling (0.9539), followed by SMOTE and SMOTE + Tomek Links. However, SMOTE + ENN reduces Precision and random undersampling causes significant drops (0.9190).

LightGBM underperforms compared to XGBoost and Random Forest, achieving highest Precision with no balancing (0.9392) and Tomek Links (0.9409), while SMOTE-based techniques show reduced performance and random undersampling performs worst (0.9040).

Decision Tree proves most sensitive to sampling techniques, performing best without balancing (0.9241), followed by Tomek Links (0.9226) and random oversampling (0.9206). SMOTE variations decline performance, with random undersampling causing the most significant drop (0.8688).

Random undersampling consistently reduces Precision across models, making it unsuitable. While SMOTE variants improve Recall, they often reduce Precision, requiring careful evaluation based on objectives. Tomek Links provides minor improvements by removing borderline samples without significantly altering distribution. Overall, XGBoost and Random Forest perform best, with balancing techniques chosen based on Precision-Recall trade-offs.

4.3.6 Comparative Analysis for 2-Class vs 3-Class Classification

Our experimental results reveal significant differences in rebalancing technique effectiveness between binary (2-class) and multi-class (3-class) classification scenarios.

In binary classification, Random Oversampling emerged as the superior technique across all algorithms, particularly with XGBoost achieving exceptional performance (0.9931 for Precision, Recall, and F1-Score). SMOTE combined with Tomek Links demonstrated that hybrid approaches effectively handle class imbalance without individual method drawbacks. Most rebalancing techniques outperformed the "No Balancing" baseline, indicating substantial improvement when addressing class imbalance in binary tasks.

The 3-class scenario presented different patterns. XGBoost maintained top performance but achieved highest Precision (0.9572) with no balancing applied. Tomek Links emerged as more effective for multi-class problems, providing minor improvements by removing borderline samples without significantly altering class distribution. SMOTE-based techniques improved Recall but often reduced Precision in multi-class settings, suggesting careful evaluation is needed for problems with more than two classes.

Model sensitivity to rebalancing varied between scenarios. In binary classification, Decision Trees showed the largest performance gap between techniques (0.1 in Macro F1), while demonstrating even greater sensitivity in multi-class scenarios, performing best with no balancing (0.9241) and worst with random undersampling (0.8688). Random Undersampling consistently underperformed across both scenarios, being particularly detrimental in 3-class settings where it drastically reduced effectiveness across all models.

The comparative analysis highlights that while addressing class imbalance is generally

beneficial, optimal rebalancing techniques differ significantly between binary and multi-class problems. XGBoost and Random Forest demonstrate robust performance across both scenarios, but balancing technique choice should be tailored to specific classification tasks, with careful consideration of Precision-Recall trade-offs, particularly in multi-class scenarios where class relationships are more complex.

4.4 Unsupervised Learning and Clustering Analysis

This section focuses on unsupervised approaches to identify patterns in unlabeled data, demonstrating the potential of these methods to detect illicit behavior when labeled data is scarce or unreliable. The analysis combines comprehensive feature selection, statistical evaluation, and clustering techniques using both XGBoost and ANOVA methodologies.

Feature selection analysis employed two distinct methods: XGBoost feature importance based on gain (reflecting accuracy improvement in decision tree splits) and ANOVA F-statistics to quantify variance across class labels. The analysis was conducted separately on *licit* (class = 1) and *illicit* (class = 0) data subsets to identify class-specific top-ranking features.

The top 10 features by ANOVA importance for *licit* and *illicit* classes are presented in Table 4.12a, while XGBoost feature importance scores are shown in Table 4.12b. To assess feature separability, Gaussian statistical descriptors (mean, standard deviation, skewness, kurtosis), normality test p-values, and Cohen's d effect sizes were computed. Discriminative metrics including mean difference, standard deviation ratio, confidence interval overlap, and discriminative flags were used to evaluate class separation. While features proved informative, no features showed completely non-overlapping confidence intervals, indicating imperfect separation.

Table 4.12: Top 10 Features

(a) ANOVA Features

	Licit Features		Illicit Features
1.	column_2	1.	column_65
2.	column_127	2.	column_59
3.	column_128	3.	column_64
4.	column_121	4.	column_58
5.	column_65	5.	column_66
6.	column_59	6.	column_60
7.	column_97	7.	column_77
8.	column_122	8.	column_76
9.	column_98	9.	column_83
10.	column_64	10.	column_78

(b) XGB Features by Importance

Licit		Illicit	
Feature	**Importance**	**Feature**	**Importance**
column_32	0.009912	column_110	0.043501
column_27	0.009906	column_98	0.038113
column_67	0.008873	column_151	0.030350
column_64	0.008421	column_160	0.029526
column_134	0.008312	column_82	0.023714
column_66	0.008178	column_61	0.023183
column_29	0.008087	column_118	0.017887
column_85	0.007976	column_119	0.017119
column_36	0.007964	column_133	0.016524
column_126	0.007834	column_100	0.016195

The distribution analysis reveals distinct patterns between classes across both ANOVA and XGBoost feature selections.

For ANOVA features (Figure 4.11), licit features (blue) exhibit highly peaked distributions with prominent central spikes, multi-modal patterns in some cases (columns 2, 65, 59, 64), and varying degrees of skewness with extended tails. Several features show discrete-like behavior with sharp peaks at specific values. In contrast, illicit features (red) display

remarkably consistent patterns with extremely narrow, concentrated distributions centered around specific negative values (primarily -0.16 to -0.13 range), demonstrating very low variance and high uniformity across transactions.

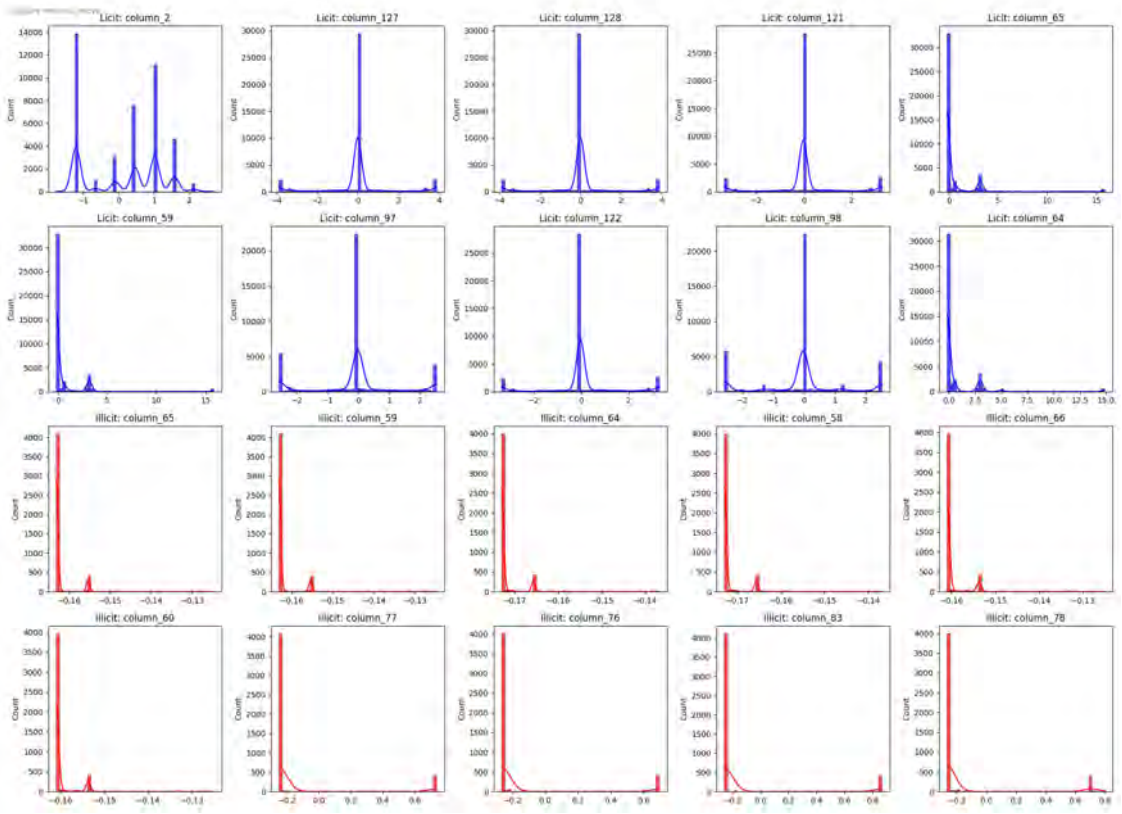


Figure 4.11: *Histograms of the top 10 ANOVA features for Licit (blue) and Illicit (red) classes.*

For XGBoost features (Figure 4.12), licit features maintain similar characteristics with sharp central peaks, some multi-modal distributions (columns 67, 85, 36), and broader value ranges. However, illicit features show more varied behavior compared to ANOVA selection — while still maintaining concentrated distributions, they exhibit different central tendencies and some features (columns 151, 118, 119, 133, 100) display wider spreads and more complex patterns.

Illicit transactions show standardized, homogeneous patterns while licit transactions display natural diversity. XGBoost captures more nuanced illicit patterns than ANOVA, demonstrating superior fraud detection capabilities through better identification of discriminative characteristics.

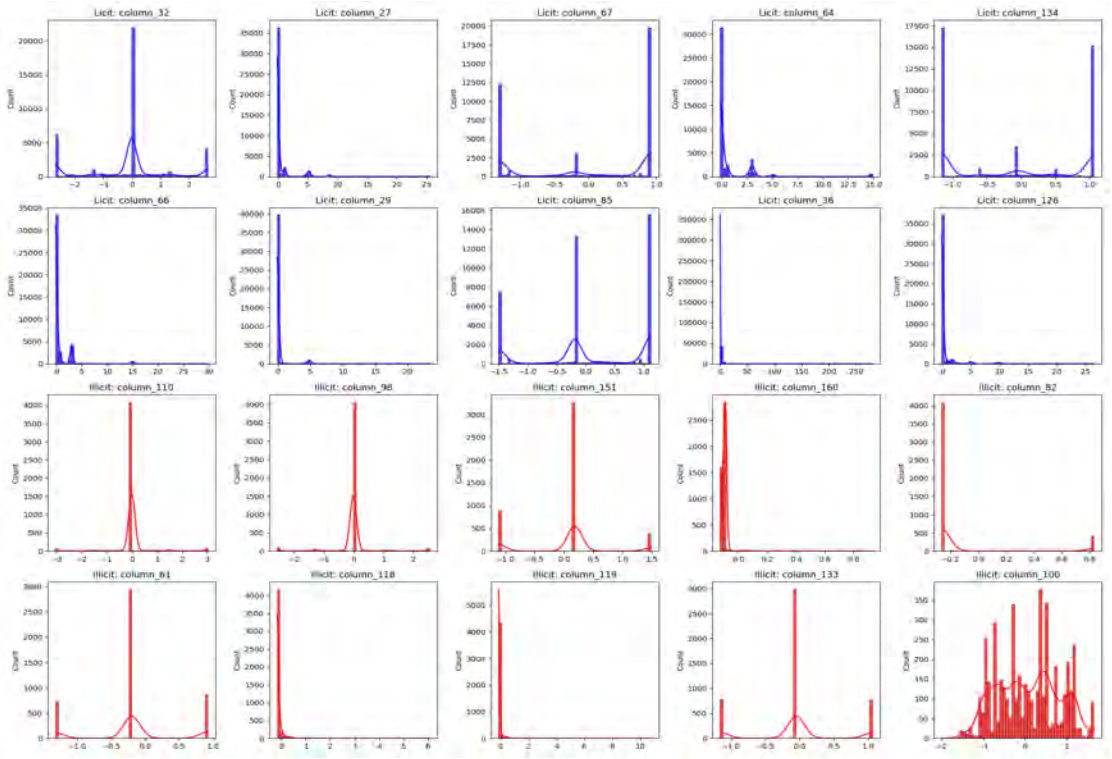


Figure 4.12: *Histogram distributions of the top 10 XGBoost-selected features for Licit (blue) and Illicit (red) classes.*

The Gaussian product analysis quantifies class-specific distributions by computing the product of Gaussian likelihoods across the top 10 XGBoost-selected features. Figure 4.13 shows that the *licit* class (blue) is characterized by a broad curve with a shallow peak, reflecting high variance and considerable feature diversity. The *illicit* class (red) displays a narrow, sharply peaked curve centered at zero, indicating very low variance where small deviations from the mean significantly reduce likelihood.

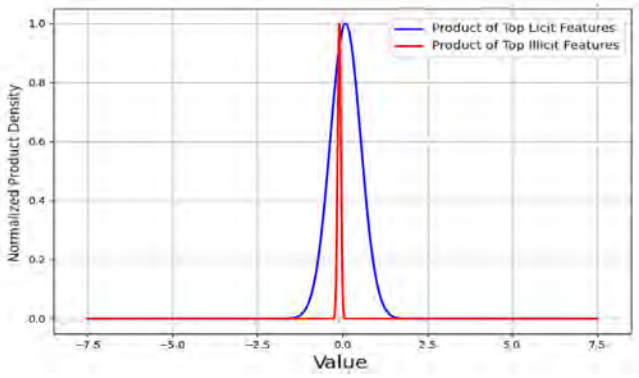


Figure 4.13: *Product of Gaussian likelihoods across the top 10 features (XGBoost)*

Figure 4.14 reveals an even more pronounced separation between classes using ANOVA—selected features. The *illicit* class (red) shows an extremely narrow, spike-like distribution with virtually no variance, while the *licit* class (blue) maintains its broader profile. This heightened contrast suggests that different feature combinations or model parameters may further

enhance class separability.

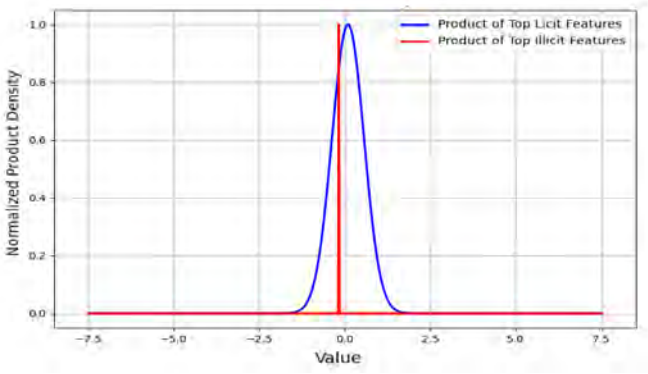


Figure 4.14: *Product of Gaussian likelihoods across the top 10 features (ANOVA)*

These distributional patterns support the suitability of probabilistic classifiers such as Naive Bayes or Gaussian Mixture Models, where effective class separation relies on density differences.

Four clustering algorithms—KMeans, GMM, DBSCAN, and HDBSCAN—were applied using both ANOVA F-score and XGBoost feature selection methods. Two ranking systems evaluated performance: Ranking_1 prioritized clustering quality (CH > DB > ARI > NMI), while Ranking_2 emphasized true label separation (ARI > NMI > DB > CH). For 2-class clustering, KMeans with RobustScaler and ANOVA features achieved the best clustering quality (CH = 44,886.24, DB = 0.640), while KMeans with Normalizer and ANOVA features provided the best true label separation (ARI = 0.0749, NMI = 0.0864). GMM consistently underperformed KMeans across all metrics, as shown in Table 4.13.

Table 4.13: Clustering Evaluation (Priority: CH, DB, ARI and NMI)

Method	Scaler	Features	CH	DB	ARI	NMI	Rank	Rank
kmeans	Normalizer	anova	29 058.82	1.1818	0.074 95	0.086 44	5	1
kmeans	RobustScaler	anova	44 886.24	0.6404	0.055 60	0.015 41	1	2
gmm	RobustScaler	anova	968.08	3.1622	−0.010 76	0.064 69	9	3
gmm	None	anova	4500.10	2.8488	−0.009 77	0.065 48	6	4
kmeans	None	anova	30 247.64	1.1088	−0.078 97	0.061 86	3	5
kmeans	StandardScaler	anova	25 493.91	1.1231	−0.075 84	0.061 43	4	6
kmeans	MinMaxScaler	anova	36 858.67	1.0339	−0.074 57	0.062 46	2	7
gmm	StandardScaler	anova	2704.29	3.5766	−0.013 88	0.062 87	7	8
gmm	Normalizer	anova	4698.68	3.0167	−0.020 53	0.060 44	10	9
gmm	MinMaxScaler	anova	2311.87	4.0055	−0.022 77	0.057 04	8	10

Figure 4.15 shows the clustering results using k=2 across different preprocessing methods and clustering algorithms, specifically evaluating how well unsupervised clustering aligns with the true binary classification of transactions (Class 0: *licit*, Class 1: *illicit*). The chart compares K-Means and GMM clustering performance across five different data preprocessing approaches: no scaling, StandardScaler, MinMaxScaler, RobustScaler, and Normalizer. Each bar is segmented to show four key metrics: the percentage and count of well-classified instances for each true class, and the misclassification rates where true classes are incorrectly assigned to the opposite cluster. The color coding distinguishes between correctly clustered licit transactions (orange), correctly clustered illicit transactions

(green), and their respective misclassifications (light purple and red). This analysis helps identify which preprocessing technique and clustering algorithm combination achieves the best natural separation between legitimate and fraudulent transaction patterns without supervised learning guidance.

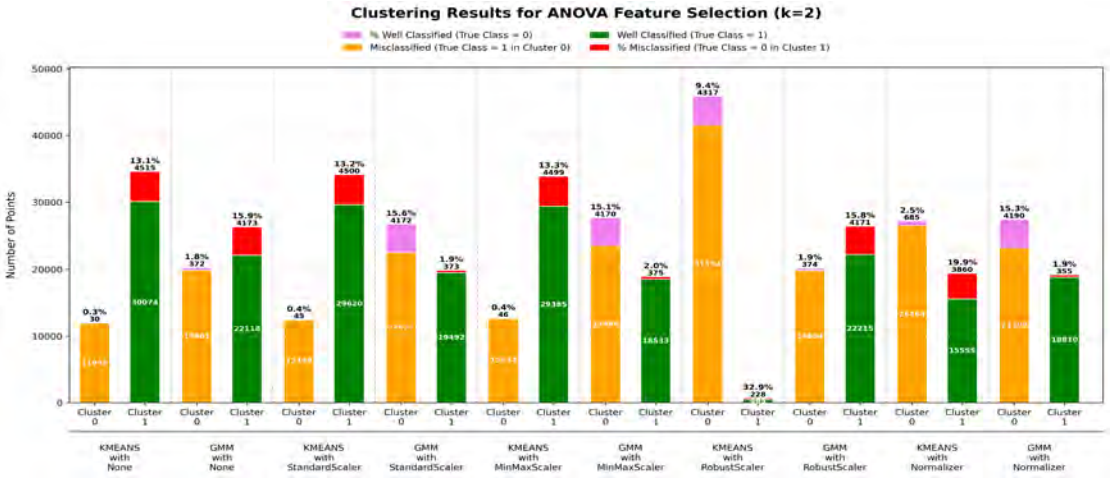
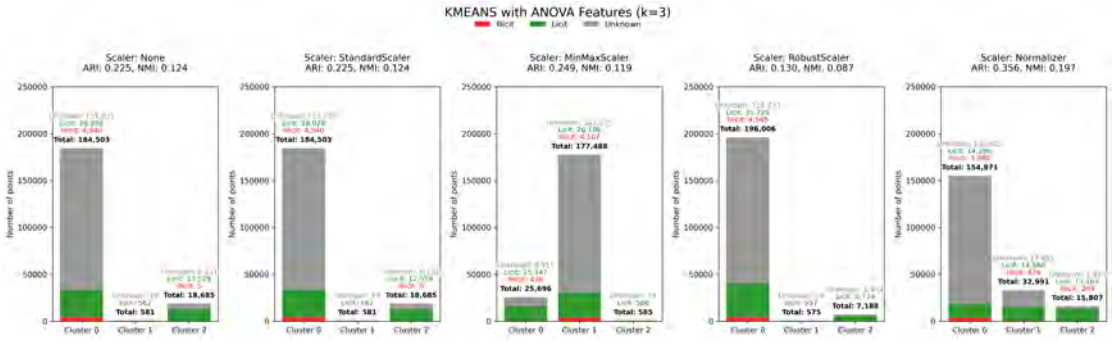
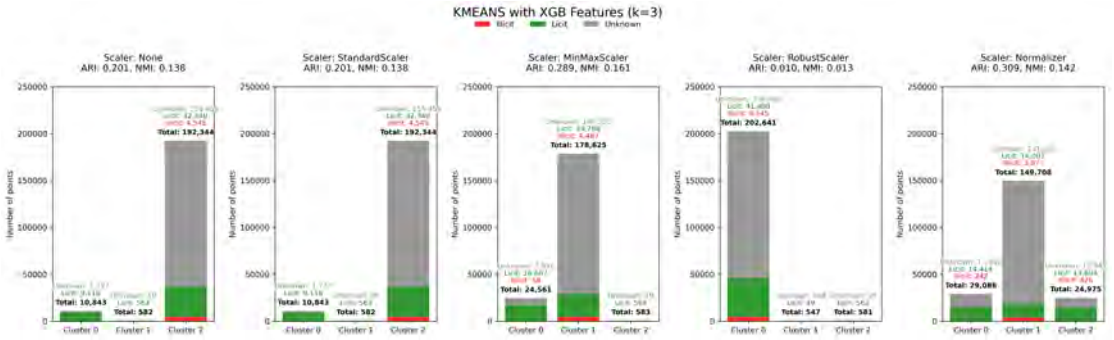


Figure 4.15: *ANOVA Feature Selection (k=2) Results*

Figure 4.16a reveals distinct clustering behaviors across different scaling methods. The RobustScaler configuration achieves the best performance metrics (ARI = 0.130, NMI = 0.087), creating more balanced cluster distributions with reduced dominance of the *Unknown* category. Notably, most scaling approaches struggle with the massive imbalance toward unknown transactions, with Cluster 0 consistently containing the majority of data points across all preprocessing methods, while Figure 4.16b demonstrates markedly different clustering patterns, with the RobustScaler again showing superior performance (ARI = 0.010, NMI = 0.013) but with much lower overall scores compared to ANOVA features. The XGBoost feature selection produces more dispersed cluster compositions, with some configurations like MinMaxScaler showing slightly better balance between known classes (*licit/illicit*) but still maintaining the overwhelming presence of unknown transactions. The comparison clearly illustrates that ANOVA-selected features provide more cohesive clustering structures, supporting the conclusion that feature selection methodology significantly impacts unsupervised learning performance in financial transaction analysis.



(a) *KMEAN+ANOVA*



(b) *KMEAN+XGB*

Figure 4.16: *3-class dataset clustering results*

For 3-class clustering, KMeans with Normalizer and ANOVA features demonstrated superior performance with ARI = 0.3561, NMI = 0.1969, CH = 527,029, and DB = 0.7356, significantly outperforming GMM methods. The comprehensive evaluation shows that KMeans consistently outperforms GMM, with ANOVA feature selection generally providing better results than XGBoost features for clustering tasks.

The clustering metrics analysis (k=2 Binary Classification) shown in Figure 4.17 reveals:

- CH — KMeans with RobustScaler and XGBoost features achieves the highest CH score (~50,000), indicating superior cluster separation and compactness. GMM methods generally show lower CH scores, with most configurations performing poorly except for specific scaler combinations.
- DB — Lower values indicate better clustering, where KMeans with RobustScaler and ANOVA features demonstrates the best performance (lowest DB score ~0.5). Interestingly, several KMeans configurations show higher DB scores, suggesting less optimal intra-cluster cohesion despite good CH performance.
- ARI — The standout performer is KMeans with Normalizer and XGBoost features (highlighted in green), achieving an ARI of approximately 0.07. Most other configurations show near-zero or negative ARI values, indicating poor alignment with true class labels.
- NMI — KMeans with Normalizer consistently achieves the highest NMI scores (~0.08) across both feature selection methods, demonstrating superior information sharing between predicted and true clusters.

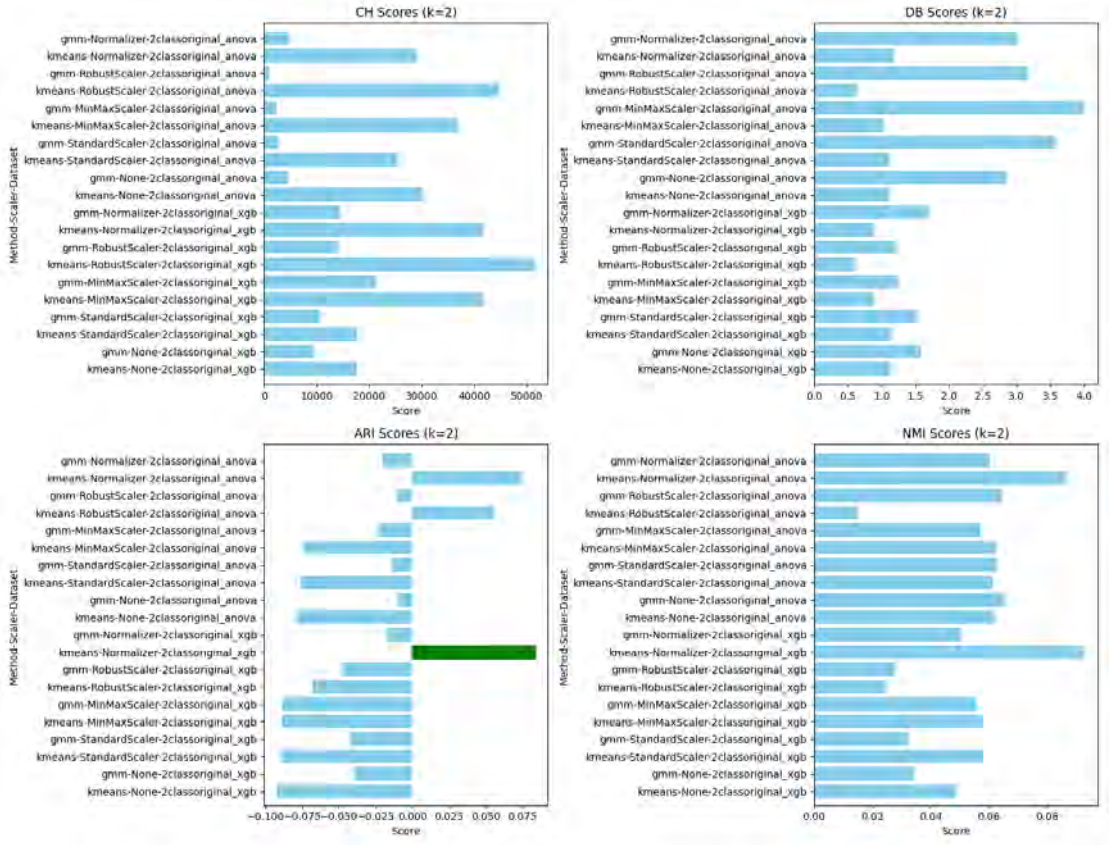


Figure 4.17: *Clustering Metrics Comparison for 2-class Version*

The 3-class clustering results, Figure 4.18, show dramatically different performance characteristics:

- CH and DB Scores — KMeans with RobustScaler and ANOVA features maintains high CH scores (~140,000) but shows concerning DB score patterns. The scaling effect becomes more pronounced with three clusters.
- ARI Performance — KMeans with Normalizer and ANOVA features (highlighted in green) achieves the best ARI score (~0.35), representing a substantial improvement over binary clustering. This suggests that the 3-class structure better captures the underlying data patterns.
- NMI Consistency — Similar to binary clustering, Normalizer-based approaches maintain superior NMI scores (~0.19), confirming the robustness of this preprocessing approach across different clustering complexities.

The analysis demonstrates that clustering performance is highly sensitive to both preprocessing methods and the number of clusters. The 3-class configuration shows markedly better ARI scores, suggesting that the financial transaction data naturally contains more than two distinct behavioral patterns. ANOVA feature selection generally outperforms XGBoost features, while Normalizer preprocessing consistently delivers the most balanced performance across multiple evaluation metrics.

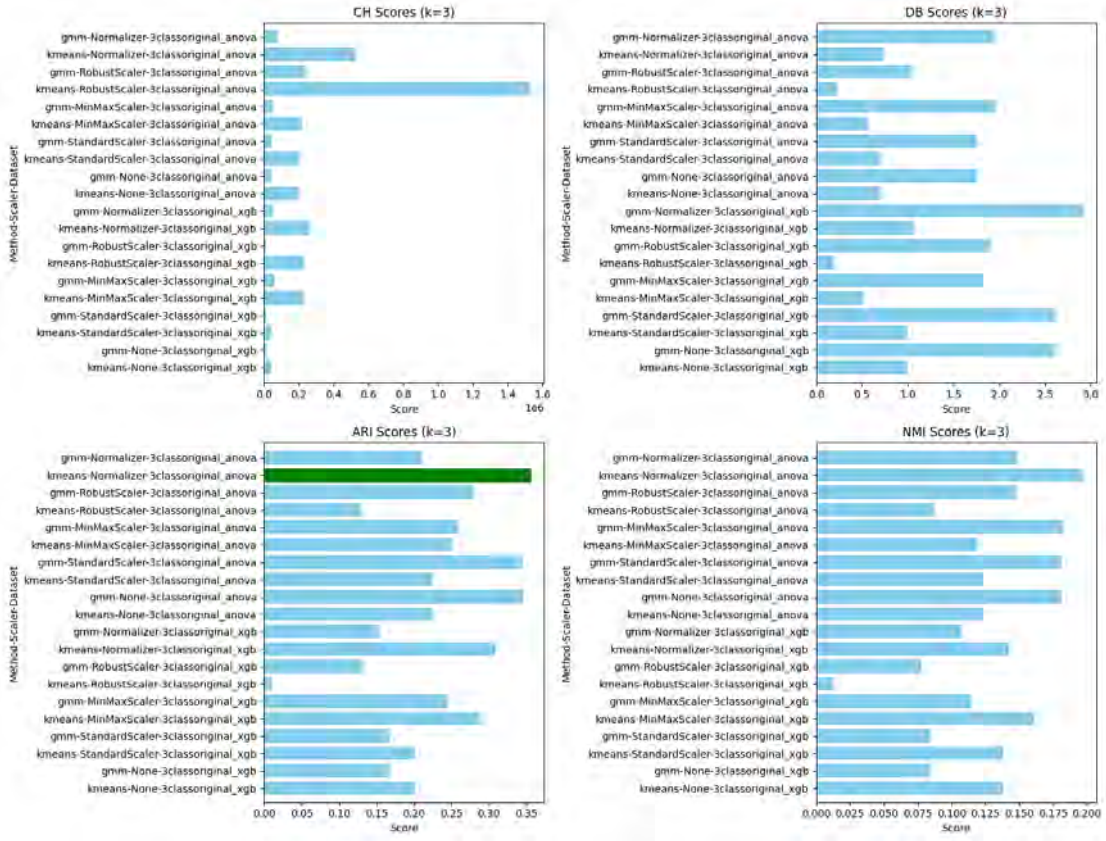
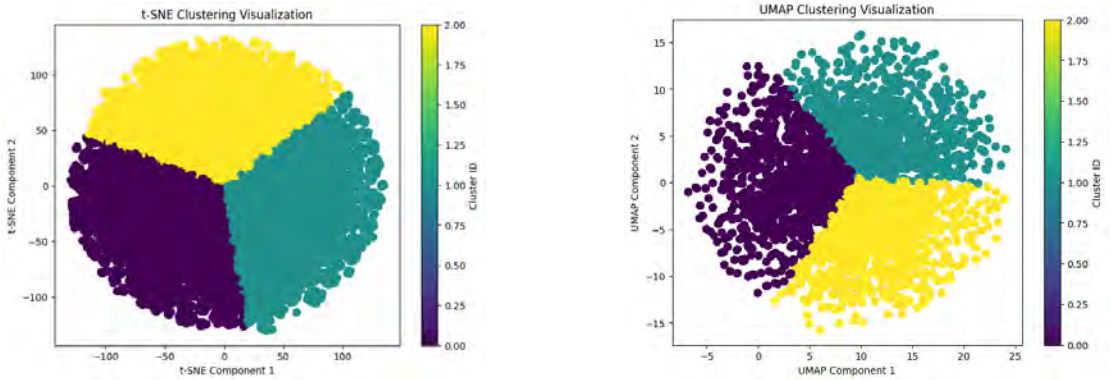


Figure 4.18: *Clustering Metrics Comparison for 3-class Version*

4.5 Clustering Visualization and Validation Analysis

This section provides visual interpretation of clustering results through dimensionality reduction, validation metrics, and comparative analysis. Dimensionality reduction techniques including PCA, t-Distributed Stochastic Neighbor Embedding (t-SNE), and UMAP were applied for enhanced visualization and interpretation. t-SNE and UMAP offer non-linear approaches capturing local neighborhood structures and cluster boundaries (Figure 4.19), while PCA provides linear dimensionality reduction preserving global structure while maximizing variance (Figure 4.21b).

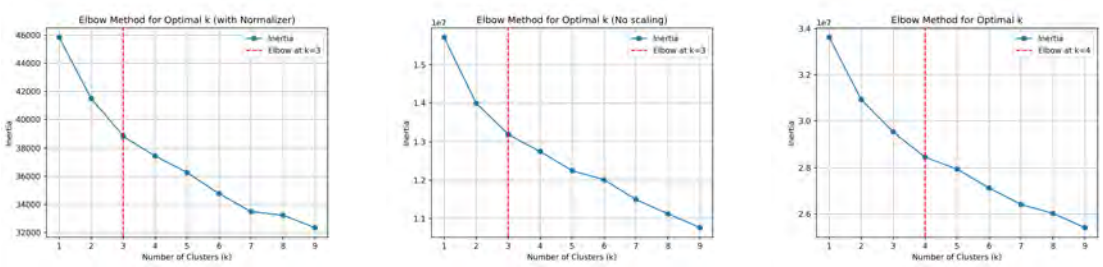


(a) *t-SNE Plot for Clustering* (b) *UMAP Plot for Clustering*

Figure 4.19: *Dimensionality reduction visualization comparison*

These complementary visualization approaches allow for comprehensive assessment of cluster separation and overlap patterns across different dimensions.

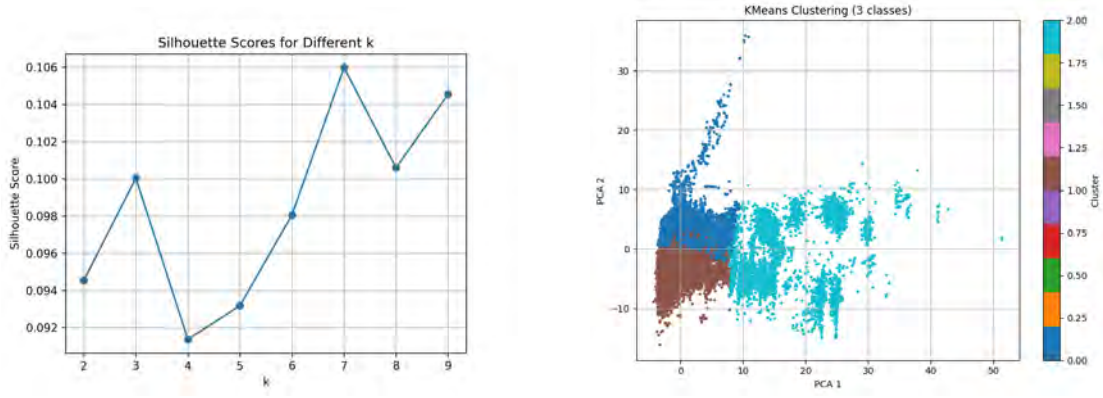
Cluster validation employed multiple methods to determine optimal cluster numbers and assess clustering quality. The K-Means elbow method was applied to both normalized and non-scaled data to identify the optimal number of clusters by examining the point where within-cluster sum of squares begins to decrease more slowly (Figures 4.20a and 4.20b). Additional elbow method analyses were conducted for different cluster configurations, including 3-class scenarios (Figure 4.20c), providing comprehensive validation across various scaling approaches and cluster numbers.



(a) *2-class Normalized Data* (b) *2-class Non-scaled Data* (c) *3-class*

Figure 4.20: *K-Means extended elbow method analysis*

Silhouette analysis (Figure 4.21a) complemented the elbow method by measuring how well each point fits within its assigned cluster compared to other clusters. The silhouette scores provide insight into cluster cohesion and separation, while PCA space visualization (Figure 4.21b) offers an additional perspective on cluster boundaries and data distribution patterns.



(a) *K-Means Silhouette Analysis* (b) *K-Means Results in PCA Space*

Figure 4.21: *Silhouette analysis and PCA visualization*

The comprehensive clustering performance evaluation demonstrates the effectiveness of the best-performing configuration: KMeans with ANOVA feature selection and normalizer preprocessing. Table 4.14 presents the key performance metrics, showing an ARI of 0.3561, indicating moderate agreement between predicted clusters and ground truth labels that substantially outperforms random clustering. The NMI of 0.1969 confirms meaningful

information sharing between clusters and true labels, while the high CH score of 527,029.54 indicates well-separated, dense clusters. The DB score of 0.7356 suggests reasonable cluster separation, and the overall accuracy of 0.8828 demonstrates strong practical performance.

Table 4.14: Clustering Performance Metrics

Metric	Value
Adjusted Rand Index (ARI)	0.3561
Normalized Mutual Information (NMI)	0.1969
Calinski-Harabasz Score (CH)	527029.54
Davies-Bouldin Score (DB)	0.7356
Accuracy (Label Matching)	0.8828

The visual comparison between ground truth labels and predicted clusters (Figure 4.22) reveals important insights into the clustering structure. The ground truth plot shows clear separation between classes 0 and 2, suggesting distinct underlying structures in the data. However, class 1 exhibits noticeable overlap with both classes 0 and 2, indicating less defined boundaries and representing the primary challenge for clustering algorithms. This overlap pattern explains the moderate ARI score while highlighting the algorithm's success in identifying the most distinct class separations. The clustering results demonstrate that unsupervised methods can effectively capture the primary structural patterns in the data, with performance limitations primarily arising from inherent class boundary ambiguities rather than algorithmic deficiencies.

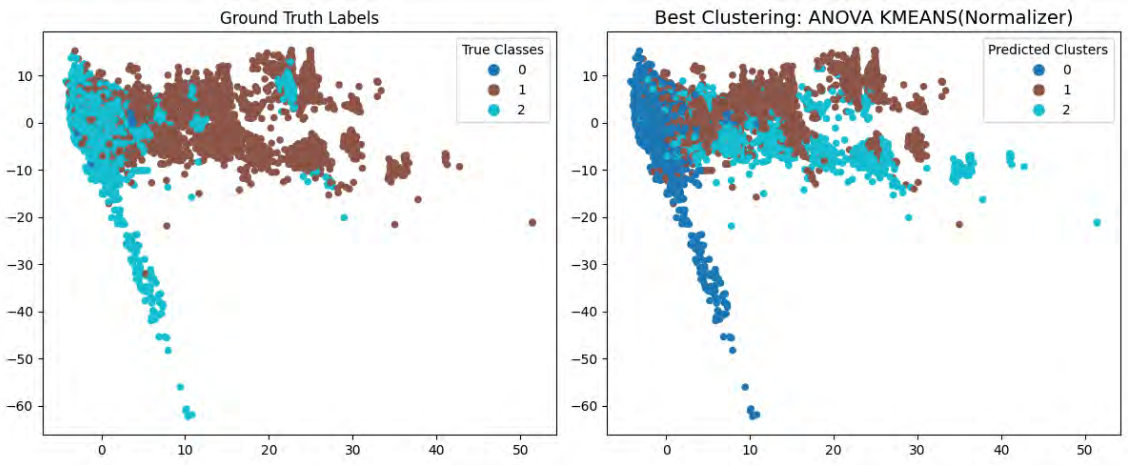


Figure 4.22: *Left: Ground Truth Labels; Right: Predicted Clusters by ANOVA KMeans (Normalized)*

4.6 Semi-Supervised Approach

This comprehensive analysis explores multiple approaches to classify *licit* and *illicit* cryptocurrency transactions, incorporating advanced clustering refinement, semi-supervised learning, and various classification methodologies to minimize unclassified cases and improve detection accuracy.

The study employed two distinct dataset configurations: a 2-Class Case with 46,564 records containing only *licit* (42,019) and *illicit* (4,545) transactions, and a 3-Class Case using

50% of the original dataset (101,884 records) including *unknown* transactions (78,659) alongside *licit* (20,974) and *illicit* (2,251) samples. This structure enabled comparative analysis between supervised and semi-supervised learning approaches.

Multiple clustering methods were evaluated using ARI, NMI, CH, and DB metrics. DBSCAN demonstrated superior performance in the 2-class scenario (ARI 0.3957, NMI 0.2296, DB 0.4701), while GMM achieved the best results for 3-class classification (ARI 0.0826, NMI 0.1157). A secondary reclustering approach was applied to refine Cluster 0 using KMeans with 3–6 sub-clusters, improving cluster purity through dimensionality reduction techniques including PCA and t-SNE visualization.

The 3-Class configuration enabled extraction of classification rules using quantile-based binarization: 44 rules for *licit* transactions and 18 rules for *illicit* transactions. This resulted in preliminary classifications of 65,229 likely *licit* and 13,430 likely *illicit* transactions. Semi-supervised learning using self-training with RF as the base classifier further refined these results to 75,589 *licit* and 3,070 *illicit* classifications. A majority voting mechanism integrated outputs from both rule-based and semi-supervised models, achieving a final distribution of 85,788 *licit* and 16,096 *illicit* transactions with a *licit/illicit* ratio of 5.33.

Multiple classification algorithms were systematically evaluated. Gaussian Naive Bayes demonstrated consistent but problematic behavior, classifying 93–95% of all transactions as *illicit* across different feature configurations (2–10 features), resulting in extremely low licit/illicit ratios (0.051–0.065) that significantly deviated from expected distributions. This bias toward *illicit* classification remained stable regardless of feature selection, indicating fundamental limitations in the algorithm's applicability to this domain.

Gaussian Mixture Models showed variable performance depending on component configuration. While most settings resulted in uniform *illicit* classifications, the 5-component configuration produced balanced results with licit/illicit ratios ranging from 1.26 to 2.28 across different feature sets. The configuration with 5 features and 5 components achieved the most balanced detection ratio (1.26), warranting further investigation for optimal performance.

SVM classification with StandardScaler, 10 features, and probability thresholds (0.7 for definitive classifications, 0.6 for likely classifications) successfully reclassified 153,965 of the 157,206 *unknown* samples, leaving only 3,241 unclassified. The final distribution included 195,953 *licit* instances (combining original *licit*, newly classified *licit*, and likely *licit*) and 4,575 *illicit* instances, resulting in a licit-to-illicit ratio of approximately 42.8:1 with *illicit* samples comprising 2.3% of the classified dataset.

Rule-based classification using minimum support threshold (0.2), minimum confidence threshold (0.7), maximum rule length (3), and 80th/20th percentile binarization thresholds produced concerning results. All 157,205 *unknown* transactions were classified as likely *licit* with zero likely *illicit* classifications, indicating potential issues with threshold selection, binarization strategy, support parameters, or rule evaluation methodology. This extreme outcome suggests the approach may be too conservative for minority class detection.

The unified pipeline architecture demonstrates flexibility across different dataset characteristics, successfully integrating feature selection (ANOVA F-test and XGBoost importance), scaling (StandardScaler), clustering methods, association rule mining, and semi-supervised learning. However, the substantial rule count required for classification (905 *licit* rules, 719 *illicit* rules) reveals the complexity of identifying deterministic features that reliably distinguish between *licit* and *illicit* transactions, highlighting the challenge of capturing distinguishing characteristics through simple, definitive patterns.

The comparative analysis reveals that no single approach provides optimal results across all scenarios. DBSCAN excels in supervised settings, GMM with specific configurations shows promise for balanced classification, SVM demonstrates strong reclassification capabilities, while rule-based approaches require significant refinement. The integration of multiple methodologies through majority voting mechanisms offers a promising direction for improving overall classification accuracy while maintaining system flexibility and scalability.

4.7 Integrated Classification Analysis

This comprehensive analysis consolidates final results and comparisons from various classification approaches applied to transform the dataset, evaluating their effectiveness in reclassifying *unknown* samples and their impact on overall dataset composition. The original dataset consisted of 42,019 *licit* and 4,545 *illicit* instances with 157,205 *unknown* samples, yielding a baseline licit-to-illicit ratio of 9.25:1 (10.8% illicit).

Two GMM-based approaches demonstrated substantially different outcomes in reclassifying the 157,205 *unknown* samples. The basic GMM configuration achieved complete reclassification, resulting in 141,004 *licit* instances (combining original 42,019 with 96,985 likely *licit*) and 60,370 *illicit* instances (combining original 4,545 with 11,067 likely *illicit* and 49,153 newly classified *illicit*). This produced a revised licit-to-illicit ratio of 2.34:1, with *illicit* samples constituting 30% of the total dataset, representing a worst-case scenario that aligns with expectations of higher illicit activity proportion in previously unclassified data.

The RobustScaler GMM approach yielded more conservative results, as detailed in Table 4.15, generating 159,713 *licit* instances (42,019 original plus 117,694 likely *licit*) and 44,056 *illicit* instances (4,545 original plus 5,192 likely *illicit* plus 34,319 newly classified *illicit*).

Table 4.15: Class Distribution: Initial vs. Final Dataset (RobustScaler GMM)

Class	**Initial Count**	**Final Count (after classification)**
Licit	42,019	159,713 (42,019 + 117,694 likely licit)
Illicit	4,545	44,056 (4,545 + 5,192 likely illicit + 34,319 illicit)
Unknown	157,205	0 (fully reclassified)

This configuration achieved a licit-to-illicit ratio of 3.63:1 with 21.6% illicit transactions. The significant increase in illicit class proportion may be justified if the model assigned

labels based on strong evidence, well-defined clusters, or high-confidence posterior probabilities reflecting genuine data structure.

SVM classification with probability thresholds successfully reclassified 153,965 of the 157,206 *unknown* samples, achieving 195,953 *licit* and 45,530 *illicit* classifications while leaving 3,241 samples unclassified. This approach produced a licit-to-illicit ratio of 4.30:1 with 18.8% illicit transactions, demonstrating balanced performance between conservative and aggressive classification strategies.

The rule-based approach exhibited extreme conservative behavior, classifying all 157,205 *unknown* transactions as likely *licit* with zero likely *illicit* classifications. This resulted in 199,224 *licit* and 4,545 *illicit* instances, maintaining a licit-to-illicit ratio of 43.83:1 with only 2.2% illicit transactions. This outcome indicates fundamental limitations in the rule-based methodology's sensitivity to minority class patterns and suggests the need for significant threshold and parameter refinement.

The classification approaches revealed significant variations in outcomes, as summarized in Table 4.16, highlighting the critical importance of method selection based on specific use case requirements and acceptable risk levels.

Table 4.16: Comparison of Classification Methods

Method	Licit Count	Illicit Count	Unknown Count	Licit/Illicit Ratio	Illicit Percentage
Original Data	42,019	4,545	157,205	9.25	10.8%
GMM (Basic)	141,004	60,370	0	2.34	30.0%
RobustScaler GMM	159,713	44,056	0	3.63	21.6%
SVM	195,953	45,530	3,241	4.30	18.8%
Rule-Based	199,224	4,545	0	43.83	2.2%

GMM basic configuration produced the most aggressive illicit detection (30% illicit), while rule-based classification proved overly conservative (2.2% illicit). RobustScaler GMM and SVM approaches achieved intermediate results (21.6% and 18.8% illicit respectively), potentially offering more balanced classification strategies.

The combination of ANOVA feature selection, Normalizer scaling, and KMeans clustering emerged as the most effective unsupervised setup, achieving the highest Adjusted Rand Index score of 0.3561. Although clustering performance was not flawless, particularly in differentiating overlapping classes, the approach successfully captured dominant data structures, indicating strong potential for unsupervised learning tasks where ground truth labels are unknown or unavailable.

The unified pipeline architecture demonstrates flexibility and scalability across datasets with differing characteristics, providing robust clustering capabilities in label-scarce contexts, interpretable rule-based classification, efficient handling of large unlabeled datasets via self-training, and seamless integration of multiple predictive signals. While the 2-class pipeline highlighted limitations in datasets lacking label diversity, the 3-class implementation showcased the system's full potential, affirming its utility in real-world forensic transaction analysis.

The comparative results reveal fundamental trade-offs between classification sensitivity and

specificity across different methodological approaches. Aggressive methods like basic GMM may capture more illicit activity but risk higher false positive rates, while conservative approaches like rule-based classification minimize false positives but may miss significant illicit transactions. The selection of optimal classification strategy depends on the specific forensic context, acceptable risk tolerance, and the relative costs of false positive versus false negative classifications in cryptocurrency transaction analysis.

4.8 Discussion

The presented work makes a valuable contribution to the challenging domain of illicit transaction detection in cryptocurrency networks, particularly in addressing the practical constraint of limited labeled data. While the methodology demonstrates thoughtful consideration of the problem's unique characteristics, several aspects warrant closer examination regarding their theoretical foundations and practical implications.

The study's strongest contribution lies in its comprehensive comparison of feature selection techniques. The dual approach employing both ANOVA and XGBoost feature importance provides robust validation of discriminative features, with the ANOVA F-statistics offering statistical rigor while XGBoost captures non-linear relationships. This combination effectively addresses the known limitation of parametric tests in detecting complex patterns in financial transaction data. The visualization of feature distributions through histograms and Gaussian approximations represents another methodological strength, offering intuitive validation of the statistical findings.

The clustering evaluation framework deserves particular praise for its multidimensional assessment using four complementary metrics (ARI, NMI, CH, and DB). This approach avoids the common pitfall of relying solely on a single metric, which could lead to biased conclusions given the complex nature of financial transaction data. The inclusion of both internal (CH, DB) and external (ARI, NMI) validation metrics provides a balanced view of cluster quality. The clustering analysis demonstrated that KMeans with ANOVA feature selection and Normalizer scaling achieved the best performance, with an ARI of 0.3561, NMI of 0.1969, CH score of 527,029.54, and DB score of 0.7356. This configuration outperformed Gaussian Mixture Models, establishing KMeans as the more suitable approach for this task. Notably, the three-class clustering scenario (including an `unknown` category) proved more effective than the two-class version, suggesting that preserving this additional class improves separation.

Semi-supervised learning approaches yielded mixed results. Self-training with Random Forest successfully rebalanced the dataset to a licit-to-illicit ratio of 5.33:1, while association rule mining proved overly conservative, classifying all unknown transactions as licit. Support Vector Machines with probability thresholds provided more balanced results, achieving a 4.30:1 ratio with 18.8% illicit classifications. Comparative analysis of classification methods revealed significant variations in performance: Gaussian Naive Bayes exhibited strong bias toward illicit classification (93—95%), while Gaussian Mixture Models with five components produced more balanced results (licit-to-illicit ratios between

1.26 and 2.28). Visualization techniques including PCA, t-SNE, and UMAP confirmed cluster separation while revealing areas of overlap between classes, supported by elbow and silhouette analysis that validated the choice of three clusters.

Several limitations and opportunities for improvement emerged from this research. Feature engineering could benefit from non-linear transformations or deep feature extraction methods like autoencoders, along with alternative feature selection approaches such as mutual information or SHAP values. Clustering performance might be enhanced through experimentation with hierarchical or spectral clustering methods, or by incorporating constraints and metric learning where partial labels exist. Semi-supervised learning could be improved by testing alternative base models like XGBoost or Graph Neural Networks and exploring active learning strategies. The rule-based approach requires refinement of binarization thresholds and rule confidence parameters to reduce bias, potentially combined with ensemble methods for greater robustness. Future evaluations should incorporate precision-recall analysis and cost-sensitive evaluation to better address the imbalanced nature of illicit transaction detection.

However, several methodological choices raise questions about the generalizability of findings. The exclusive focus on KMeans and GMM for clustering, while justified by computational efficiency, overlooks more recent advances in deep clustering approaches that might better handle the hierarchical nature of transaction patterns. The apparent superiority of KMeans should be interpreted cautiously, as its performance on high-dimensional financial data often depends heavily on preprocessing choices and distance metric selection.

The semi-supervised learning component presents both opportunities and challenges. While the self-training approach with Random Forest as the base classifier is theoretically sound, the implementation details regarding confidence thresholds and stopping criteria remain unclear. In practice, such methods can suffer from confirmation bias if the initial labeled set is not representative or if the confidence thresholds are poorly calibrated. The reported licit-to-illicit ratio of 5.33:1 suggests potential bias in the classification process that warrants further investigation.

From a theoretical perspective, the Gaussian distribution assumptions underlying several analyses (particularly in the GMM and Naive Bayes implementations) may not adequately capture the heavy-tailed distributions characteristic of financial transaction data. While the visual analysis of feature distributions provides some justification, the lack of formal tests for distributional assumptions represents a notable gap. The study would benefit from explicit discussion of how potential violations of these assumptions might affect the results.

The rule-based classification results, where all unknown transactions were classified as licit, reveal fundamental challenges in the current approach. This outcome suggests either: (1) insufficient discriminative power in the selected features, (2) overly conservative rule generation parameters, or (3) inherent limitations in the binarization strategy. Each possibility has different implications for both the methodology and the practical applicability of the results.

The practical implementation of these findings faces several hurdles. The reported 18.8% illicit transaction rate from the SVM approach, while plausible, exceeds typical estimates for most cryptocurrency networks. This discrepancy could reflect either superior detection capability or potential over-classification, highlighting the need for careful threshold calibration in real-world deployment. The computational costs of maintaining and updating the proposed system, particularly the rule-based components, should also be considered given the dynamic nature of cryptocurrency transaction patterns.

The study's most significant practical contribution may lie in its demonstration that unsupervised methods can achieve moderate detection accuracy (ARI = 0.3561) without labeled data. This finding has important implications for jurisdictions or platforms where labeled illicit transactions are scarce. However, the moderate NMI score (0.1969) suggests that substantial room for improvement remains before such systems could operate autonomously in high-stakes regulatory applications.

In conclusion, this work demonstrates that unsupervised clustering using KMeans with ANOVA feature selection, combined with semi-supervised refinement through SVM or Random Forest, provides an effective framework for detecting illicit transactions when labeled data is limited. The results highlight that no single method offers a perfect solution, suggesting that future research should focus on developing hybrid approaches, improving feature representations, and implementing cost-sensitive evaluation metrics to optimize real-world deployment in cryptocurrency transaction analysis.

5 Conclusions

This dissertation presented **Bitcoin Anomaly Detection (BAD)**, a hybrid machine learning framework designed to detect fraudulent transactions on the Bitcoin blockchain. By integrating supervised, unsupervised, and semi-supervised learning techniques within a unified pipeline, this work addressed major challenges in cryptocurrency fraud detection, including high class imbalance, limited labeled data, and the need for interpretability.

The methodology was built on the Elliptic Bitcoin dataset and involved graph network analysis, dimensionality reduction, clustering, rule mining, and ensemble classification. Experimental results confirmed that models such as XGBoost and Random Forest, when integrated with advanced feature selection and self-training mechanisms, achieved high classification performance (accuracy >97%, FPR <4.5%).

Furthermore, graphical and statistical analyses revealed structural patterns in illicit transaction networks, such as sink nodes and star-like inflows, aiding forensic investigation. The BAD system also incorporated XAI techniques, ensuring transparency and alignment with regulatory frameworks like MiCA and FATF.

The proposed system demonstrated adaptability across various dataset configurations (2-class, 3-class) and scalability to unlabeled data environments. These findings validate the BAD framework as a viable and effective tool for financial crime prevention in the evolving digital asset landscape.

In conclusion, while the study makes valuable contributions to the field, its findings should be interpreted as establishing proof-of-concept rather than providing definitive solutions. The work successfully demonstrates that unsupervised and semi-supervised methods can provide meaningful insights when labeled data is limited, but also highlights the complex trade-offs between detection accuracy, false positive rates, and computational practicality that characterize this challenging domain.

5.1 Future Work

The comparative weakness of rule-based methods in this study highlights an opportunity to explore hybrid symbolic-subsymbolic approaches, which combine the interpretability of rules with the pattern recognition capabilities of machine learning. Advances in neuro-symbolic AI may offer promising paths for such integration, while maintaining the auditability crucial for financial applications.

Two key research directions emerge:

- Neighborhood-based Classification via Graph Analysis — Incorporating structural features such as node degree, betweenness, and edge density into the classification pipeline could significantly enhance contextual understanding. Identifying *sink nodes* (e.g., high in-degree, zero out-degree illicit nodes) and constructing subgraphs of neighboring transactions may allow comparison to known licit/illicit patterns, enriching the BAD model with localized risk indicators.
- Hybrid Neuro-Symbolic Integration — Combining rule-based logic with machine learning could improve performance and interpretability. Neuro-symbolic frameworks may enable this synthesis, allowing BAD to balance robust detection with transparent decision-making.

This integration of graph structure and hybrid AI methods represents a natural evolution of BAD, improving both its contextual sensitivity and practical utility for AML applications.

References

[1] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules". In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*. The seminal paper introducing Apriori. Morgan Kaufmann, 1994, pp. 487–499. URL: http://www.vldb.org/conf/1994/P487.PDF (cit. on p. 45).

[2] A. Ahmed and O. Alabi. "Secure and Scalable Blockchain-Based Federated Learning for Cryptocurrency Fraud Detection: A Systematic Review". In: *IEEE Access* 12 (2024), pp. 102219–102241. DOI: 10.1109/ACCESS.2024.3429205. URL: https://doi.org/10.1109/ACCESS.2024.3429205 (cit. on p. 16).

[3] C. Akcora, Y. Li, Y. Gel, and M. Kantarcioglu. "BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain". In: *arXiv* (June 2019). DOI: 10.48550/arXiv.1906.07852. URL: https://doi.org/10.48550/arXiv.1906.07852 (cit. on p. 19).

[4] I. Alarab and S. Prakoonwit. "Graph-Based LSTM for Anti-Money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data". In: *Neural Processing Letters* 55.1 (Feb. 2023), pp. 689–707. DOI: 10.1007/s11063-022-10904-8. URL: https://doi.org/10.1007/s11063-022-10904-8 (cit. on pp. 10, 14).

[5] I. Alarab and S. Prakoonwit. "Robust Recurrent Graph Convolutional Network Approach Based Sequential Prediction of Illicit Transactions in Cryptocurrencies". In: *Multimedia Tools and Applications* 83.20 (Dec. 2023), pp. 58449–58464. DOI: 10.1007/s11042-023-17323-4. URL: https://doi.org/10.1007/s11042-023-17323-4 (cit. on pp. 13, 14).

[6] M. Allende, D. León, S. Cerón, A. Pareja, E. Pacheco, A. Leal, M. Silva, et al. "Quantum-Resistance in Blockchain Networks". In: *Scientific Reports* 13.1 (Apr. 2023), p. 5664. DOI: 10.1038/s41598-023-32701-6. URL: https://doi.org/10.1038/s41598-023-32701-6 (cit. on p. 16).

[7] S. Alsaif. "Machine Learning-Based Ransomware Classification of Bitcoin Transactions". In: *Applied Computational Intelligence and Soft Computing* 2023 (Jan. 23, 2023). Ed. by B. Erdebilli, pp. 1–10. DOI: 10.1155/2023/6274260. URL: https://doi.org/10.1155/2023/6274260 (cit. on p. 19).

[8] P. Andrews. *The Rise of Layer 2 Solutions: Security Implications and Audit Necessities*. Accessed: February 21, 2025. 2023. URL: https://altcoininvestor.com/the-rise-of-layer-2-solutions/ (cit. on p. 19).

[9] A. Antonopoulos. *The Internet of Money*. Vol. 1. Accessed: February 21, 2025. Andreas M. Antonopoulos, 2017. URL: https://archive.org/details/TheInternetOfMoney-Volume1/1-05-Bitcoin_DumbNetworksInnovationAndTheFestivalOfTheCommons.mp3 (cit. on p. 2).

[10] A. Arbabi, A. Shojaeinasab, B. Bahrak, and H. Najjaran. "Mixing Solutions in Bitcoin and Ethereum Ecosystems: A Review and Tutorial". In: *arXiv* (Oct. 2023). DOI: 10.48550/arXiv.2310.04899. URL: https://doi.org/10.48550/arXiv.2310.04899 (cit. on p. 9).

[11] D. Arthur and S. Vassilvitskii. "k-means++: The Advantages of Careful Seeding". In: *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)* (2007), pp. 1027–1035. DOI: 10.1145/1283383.1283494 (cit. on p. 42).

[12] M. Bartoletti, B. Pes, and S. Serusi. "Data Mining for Detecting Bitcoin Ponzi Schemes". In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 75–84. DOI: 10.1109/CVCBT.2018.00014 (cit. on p. 10).

[13] C. Borgelt. "An Implementation of the FP-Growth Algorithm". In: *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*. ACM, 2005, pp. 1–5. DOI: 10.1145/1133905.1133907. URL: https://doi.org/10.1145/1133905.1133907 (cit. on p. 45).

[14] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (cit. on p. 10).

[15] T. Caliński and J. Harabasz. "A Dendrite Method for Cluster Analysis". In: *Communications in Statistics* 3.1 (1974), pp. 1–27. DOI: 10.1080/03610927408827101. URL: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101 (cit. on p. 42).

[16] R. J. Campello, D. Moulavi, and J. Sander. "Density-based clustering based on hierarchical density estimates". In: *Advances in Knowledge Discovery and Data Mining* (2013), pp. 160–172 (cit. on p. 39).

[17] Chainalysis Team. *The Chainalysis 2024 Crypto Crime Report*. Tech. rep. Accessed: February 21, 2025. Chainalysis, Jan. 2024 (cit. on p. 3).

[18] Chainalysis Team. *Zero Knowledge Rollups & Optimistic Rollups: An Overview*. Accessed: February 21, 2025. 2024. URL: https://www.chainalysis.com/blog/zero-knowledge-rollups-optimistic-rollups-overview/ (cit. on p. 19).

[19] S. Chen, Z. Qian, W. Siu, X. Hu, J. Li, S. Li, Y. Qin, T. Yang, Z. Xiao, W. Ye, Y. Zhang, Y. Dong, and Y. Zhao. "PyOD 2: A Python Library for Outlier Detection with LLM-powered Model Selection". In: *arXiv preprint arXiv:2412.12154* (2024). https://arxiv.org/abs/2412.12154 (cit. on p. 9).

[20] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785 (cit. on pp. 10, 35).

[21] Coin Dance. *Coin Dance Portal*. Accessed: February 21, 2025. 2025. URL: https://coindance.com (cit. on p. 3).

[22] CoinGecko Research Team. *Global Cryptocurrency Market Cap Charts*. Available at: https://www.coingecko.com/en/global-charts (Accessed: February 21, 2025). 2024 (cit. on p. 1).

[23] M. Conti, S. Kumar, C. Lal, and S. Ruj. "A Survey on Security and Privacy Issues of Bitcoin". In: *arXiv preprint arXiv:1706.00916* (2017). Version 3. DOI: 10.48550/arXiv.1706.00916. URL: https://arxiv.org/abs/1706.00916 (cit. on pp. 1, 4, 8).

[24] L. Cunha, M. Brito, D. Oliveira, and A. Martins. "Active Learning in the Detection of Anomalies in Cryptocurrency Transactions". In: *Machine Learning and Knowledge Extraction* 5 (2023), pp. 1717–1745 (cit. on p. 10).

[25] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909. URL: https://ieeexplore.ieee.org/document/4766909 (cit. on pp. 42, 43).

[26] Elliptic, www.elliptic.co. *Elliptic Data Set*. https://www.kaggle.com/datasets/ellipticco/elliptic-data-set. 2019 (cit. on pp. 19, 21, 24).

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press. 1996, pp. 226–231. URL: https://dl.acm.org/doi/10.5555/3001460.3001507 (cit. on p. 39).

[28] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)* (1996), pp. 226–231. DOI: 10.5555/3001460.3001507 (cit. on p. 42).

[29] European Union. *Markets in Crypto-Assets Regulation (MiCA)*. Accessed: February 21, 2025. 2023. URL: https://www.europarl.europa.eu/doceo/document/A-9-2022-0345_EN.html (cit. on pp. 1, 5, 17).

[30] FBI's Internet Crime Complaint Center. *2023 IC3 Cryptocurrency Crime Report*. 2023. URL: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf (cit. on p. 3).

[31] Financial Action Task Force. *Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers*. Tech. rep. FATF, June 2023. URL: https://www.fatf-gafi.org/ (cit. on pp. 5, 18).

[32] Financial Crimes Enforcement Network. *FinCEN Cryptocurrency Guidelines*. Accessed: February 21, 2025. 2023. URL: https://www.fincen.gov/resources/statutes-regulations/guidance (cit. on p. 18).

[33] M. Harlev, H. Yin, K. Langenheldt, R. Mukkamala, and R. Vatrapu. "Breaking Bad: De-anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning". In: *Proceedings of the 51st Hawaii International Conference on System Sciences*. 2018 (cit. on p. 13).

[34] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd. Springer, 2009. ISBN: 978-0-387-84857-0. URL: https://www.springer.com/gp/book/9780387848570 (cit. on p. 10).

[35] S. Haykin. *Neural Networks and Learning Machines*. 3rd. Pearson, 2009. ISBN: 978-0-131-47139-9. URL: https://www.pearson.com/en-us/subject-catalog/p/neural-networks-and-learning-machines/P200000003595/9780131471399 (cit. on p. 10).

[36] S. Hisham, M. Makhta, and A. Aziz. "A Comprehensive Review of Significant Learning for Anomalous Transaction Detection Using a Machine Learning Method in a Decentralized Blockchain Network". In: *International Journal of Advanced Technology and Engineering Exploration* 9.95 (2022), pp. 1366–1396. DOI: 10.19101/IJATEE.2021.876322. URL: https://doi.org/10.19101/IJATEE.2021.876322 (cit. on p. 8).

[37] Horizen Academy. *What are Cross Chain Transactions?* Accessed: February 21, 2025. 2024. URL: https://www.horizen.io/academy/cross-chain-transactions/ (cit. on p. 19).

[38] D. Hovstadius. "Anti-Money Laundering with Unreliable Labels". Master's thesis. Uppsala University, June 2024. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-533771 (cit. on p. 19).

[39] Y. Hu, S. Seneviratne, K. Thilakarathna, K. Fukuda, and A. Seneviratne. "Characterizing and Detecting Money Laundering Activities on the Bitcoin Network". In: (2019). arXiv:1912.12060. URL: https://arxiv.org/abs/1912.12060 (cit. on p. 13).

[40] L. Hubert and P. Arabie. "Comparing Partitions". In: *Journal of Classification* 2 (1985), pp. 193–218. DOI: 10.1007/BF01908075. URL: https://link.springer.com/article/10.1007/BF01908075 (cit. on pp. 42, 43).

[41] Immutable. *Zero-Knowledge vs. Optimistic Rollups Explained: Which One is Better for Blockchain Games?* Accessed: February 21, 2025. 2023. URL: https://www.immutable.com/blog/zero-knowledge-vs-optimistic-rollups-explained-which-one-is-better-for-blockchain-games (cit. on p. 19).

[42] J. H. W. Jr. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244. DOI: 10.1080/01621459.1963.10500845. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845 (cit. on p. 42).

[43] M. Jullum, A. Løland, R. Huseby, G. Ånonsen, and J. Lorentzen. "Detecting Money Laundering Transactions with Machine Learning". In: *Journal of Money Laundering Control* 23.1 (Jan. 2020), pp. 173–186. DOI: 10.1108/JMLC-07-2019-0055. URL: https://doi.org/10.1108/JMLC-07-2019-0055 (cit. on p. 13).

[44] P. Kamuangu. "A Review on Financial Fraud Detection using AI and Machine Learning". In: *Journal of Economics, Finance and Accounting Studies* 6.1 (Feb. 2024), pp. 67–77. DOI: 10.32996/jefas.2024.6.1.7. URL: https://doi.org/10.32996/jefas.2024.6.1.7 (cit. on p. 13).

[45] K. Karadag. *Validity Fraud Proofs in Layer 2 Solutions*. Accessed: February 21, 2025. 2024. URL: https://medium.com/@keremkaradag/validity-fraud-proofs-in-layer-2-solutions-53c39dd5e86e (cit. on p. 19).

[46] R. Karim, F. Hermsen, S. Chala, P. Perthuis, and A. Mandal. "Scalable Semi-Supervised Graph Learning Techniques for Anti Money Laundering". In: *IEEE Access* 12 (2024), pp. 50012–50029. DOI: 10.1109/ACCESS.2024.3383784 (cit. on p. 19).

[47] J. Kim, M. Nakashima, W. Fan, S. Wuthier, X. Zhou, I. Kim, and S. Chang. "Anomaly Detection Based on Traffic Monitoring for Secure Blockchain Networking". In: *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. Sydney, Australia: IEEE, 2021, pp. 1–9. DOI: 10.1109/ICBC51069.2021.9461119. URL: https://doi.org/10.1109/ICBC51069.2021.9461119 (cit. on p. 14).

[48] C. Lee, S. Maharjan, K. Ko, and J. Hong. "Toward Detecting Illegal Transactions on Bitcoin Using Machine-Learning Methods". In: *Blockchain and Trustworthy Systems*. Ed. by Z. Zheng, H.-N. Dai, M. Tang, and X. Chen. Vol. 1156. Communications in Computer and Information Science. Springer Singapore, 2020, pp. 520–533. DOI: 10.1007/978-981-15-2777-7_42. URL: https://doi.org/10.1007/978-981-15-2777-7_42 (cit. on p. 13).

[49] N. Li, M. Qi, Z. Xu, X. Zhu, W. Zhou, S. Wen, and Y. Xiang. "Blockchain Cross-Chain Bridge Security: Challenges, Solutions, and Future Outlook". In: *Distributed Ledger Technologies: Research and Practice* 4.1 (Mar. 2025), pp. 1–34. DOI: 10.1145/3696429. URL: https://doi.org/10.1145/3696429 (cit. on pp. 18, 19).

[50] J. Liu, H. Zhao, Y. Lyu, and X. Yue. "The Provision Strategy of Blockchain Service under the Supply Chain with Downstream Competition". In: *Annals of Operations Research* 327.1 (Aug. 2023), pp. 375–400. DOI: 10.1007/s10479-022-05034-2. URL: https://doi.org/10.1007/s10479-022-05034-2 (cit. on p. 14).

[51] S. Lloyd. "Least Squares Quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137. ISSN: 0018-9448. DOI: 10.1109/TIT.1982.1056489. URL: http://ieeexplore.ieee.org/document/1056489/ (cit. on pp. 39, 42).

[52] J. Lorenz, M. Silva, D. Aparício, J. Ascensão, and P. Bizarro. "Machine Learning Methods to Detect Money Laundering in the Bitcoin Blockchain in the Presence of Label Scarcity". In: *arXiv preprint arXiv:2005.14635* (2020). https://arxiv.org/abs/2005.14635 (cit. on pp. 4, 13).

[53] S. Lundberg and S.-I. Lee. "A Unified Approach to Interpreting Model Predictions". In: (2017). Core SHAP theory, pp. 4765–4774. DOI: 10.48550/arXiv.1705.07874. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (cit. on p. 56).

[54] J. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. University of California Press, 1967, pp. 281–297. URL: https://digicoll.lib.berkeley.edu/record/113015?v=pdf (cit. on p. 39).

[55] L. McInnes and J. Healy. "hdbscan: Hierarchical density based clustering". In: *The Journal of Open Source Software* 2.11 (2017), p. 205. DOI: 10.21105/joss.00205 (cit. on p. 42).

[56] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. Voelker, and S. Savage. "A fistful of bitcoins: Characterizing payments among men with no names". In: *Proceedings of the 2013 Conference on Internet Measurement Conference*. IMC '13. ACM, 2013, pp. 127–140. DOI: 10.1145/2504730.2504747 (cit. on p. 4).

[57] A. Mohan, K. P., P. Sankar, M. K., and A. Peter. "Improving anti-money laundering in bitcoin using evolving graph convolutions and deep neural decision forest". In: *Data Technologies and Applications* 57 (Nov. 2022), pp. 1–17. DOI: 10.1108/DTA-06-2021-0167. URL: https://doi.org/10.1108/DTA-06-2021-0167 (cit. on p. 14).

[58] P. Monamo, V. Marivate, and B. Twala. "Unsupervised Learning for Robust Bitcoin Fraud Detection". In: *2016 Information Security for South Africa (ISSA)*. IEEE, 2016, pp. 129–134 (cit. on p. 12).

[59] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Bitcoin.org* (2008). https://bitcoin.org/bitcoin.pdf (cit. on p. 1).

[60] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun. "The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and an Academic Review of Literature". In: *Decision Support Systems* 50.3 (2011), pp. 559–569 (cit. on p. 12).

[61] A. Olutimehin. "The Synergistic Role of Machine Learning, Deep Learning, and Reinforcement Learning in Strengthening Cyber Security Measures for Crypto Currency Platforms". In: *Asian Journal of Research in Computer Science* 18.3 (Feb. 2025), pp. 190–212. DOI: 10.9734/ajrcos/2025/v18i3586. URL: https://doi.org/10.9734/ajrcos/2025/v18i3586 (cit. on p. 16).

[62] M. Patel. "Fraud on the Crypto Market". In: *SSRN Electronic Journal* (2022). DOI: 10.2139/ssrn.4278973. URL: https://doi.org/10.2139/ssrn.4278973 (cit. on p. 3).

[63] T. Pham and S. Lee. "Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods". In: *arXiv* (Feb. 2017). DOI: 10.48550/arXiv.1611.03941. URL: https://doi.org/10.48550/arXiv.1611.03941 (cit. on p. 12).

[64] J. Quinlan. "Induction of Decision Trees". In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251 (cit. on p. 10).

[65] E. Ruiz and J. Angelis. "Combating Money Laundering with Machine Learning – Applicability of Supervised-Learning Algorithms at Cryptocurrency Exchanges". In: *Journal of Money Laundering Control* 25.4 (Oct. 2022), pp. 766–778. DOI: 10.1108/JMLC-09-2021-0106. URL: https://doi.org/10.1108/JMLC-09-2021-0106 (cit. on p. 14).

[66] M. Sabbaghi. *Which crypto coins are quantum resistant.* Accessed: February 21, 2025. 2024. URL: https://www.uniblock.dev/blog/which-crypto-coins-are-quantum-resistant (cit. on p. 17).

[67] B. Sakız and A. Gencer. "Blockchain Technology and Its Impact on the Global Economy". In: *Proceedings of International Conference of Eurasian Economies.* Famagusta, Turkish Republic of Northern Cyprus, 2019, pp. 98–105. DOI: 10.36880/C11.02258 (cit. on p. 1).

[68] T. Sathya, N. Keertika, S. Shwetha, D. Upodhyay, and M. Hasanov. "Bitcoin Heist Ransomware Attack Prediction Using Data Science Process". In: *E3S Web of Conferences* 399 (2023). Ed. by V. Vijayan and T. S. Kumar, p. 04056. DOI: 10.1051/e3sconf/202339904056. URL: https://doi.org/10.1051/e3sconf/202339904056 (cit. on p. 19).

[69] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), 19:1–19:21. DOI: 10.1145/3068335 (cit. on p. 42).

[70] A. Shojaeinasab. "Decoding Illicit Bitcoin Transactions: A Multi-Methodological Approach for Anti-Money Laundering and Fraud Detection in Cryptocurrencies". PhD dissertation. Victoria, Canada: University of Victoria, 2024 (cit. on pp. 9, 14).

[71] P. Singh, D. Agrawal, and S. Pandey. "Anomaly Detection and Analysis in Blockchain Systems". In: *In Review* (Jan. 2023). DOI: 10.21203/rs.3.rs-2414745/v1. URL: https://doi.org/10.21203/rs.3.rs-2414745/v1 (cit. on p. 10).

[72] A. Strehl and J. Ghosh. "Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions". In: *Journal of Machine Learning Research* 3 (2002), pp. 583–617. DOI: 10.1162/153244303321897735. URL: https://www.jmlr.org/papers/volume3/strehl02a/strehl02a.pdf (cit. on pp. 42, 44).

[73] Stripe. *A Primer on Machine Learning for Fraud Protection*. Tech. rep. Accessed: February 21, 2025. Stripe, June 2023. URL: https://stripe.com/guides/primer-on-machine-learning-for-fraud-protection (cit. on p. 4).

[74] Q. Umer, J. Li, M. Ashraf, R. Bashir, and H. Ghous. "Ensemble Deep Learning-Based Prediction of Fraudulent Cryptocurrency Transactions". In: *IEEE Access* 11 (2023), pp. 95213–95224. DOI: 10.1109/ACCESS.2023.3310576. URL: https://doi.org/10.1109/ACCESS.2023.3310576 (cit. on p. 10).

[75] United Nations Office on Drugs and Crime. *Estimating Illicit Financial Flows Resulting from Drug Trafficking and Other Transnational Organized Crimes*. Accessed: February 21, 2025. United Nations, 2011. URL: https://www.unodc.org/documents/data-and-analysis/Studies/Illicit_financial_flows_2011_web.pdf (cit. on p. 1).

[76] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998. ISBN: 978-0-471-03003-4. URL: https://www.wiley.com/en-us/Statistical+Learning+Theory-p-9780471030034 (cit. on p. 10).

[77] D. Vassallo, V. Vella, and J. Ellul. "Application of Gradient Boosting Algorithms for Anti-Money Laundering in Cryptocurrencies". In: *SN Computer Science* 2.3 (May 2021), p. 143. DOI: 10.1007/s42979-021-00558-z. URL: https://doi.org/10.1007/s42979-021-00558-z (cit. on p. 14).

[78] M. Weber, G. Domeniconi, J. Chen, D. Weidele, C. Bellei, T. Robinson, and C. Leiserson. *Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics*. Tech. rep. arXiv preprint, 2019. URL: https://arxiv.org/abs/1908.02591 (cit. on pp. 1, 13, 19).

[79] M. Weber, J. Chen, T. Suzumura, A. Pareja, T. Ma, H. Kanezashi, T. Kaler, C. Leiserson, and T. Schardl. "Scalable Graph Learning for Anti-Money Laundering: A First Look". In: *arXiv* (Nov. 30, 2018). DOI: 10.48550/arXiv.1812.00076. URL: https://github.com/IBM/AMLSim (cit. on pp. 19, 21).

[80] D. Yaga, P. Mell, N. Roby, and K. Scarfone. *Blockchain Technology Overview*. Tech. rep. National Institute of Standards and Technology, 2018. DOI: 10.6028/NIST.IR.8202 (cit. on p. 2).

[81] X. Yang, C. Zhang, Y. Sun, K. Pang, L. Jing, S. Wa, and C. Lv. "FinChain-BERT: A High-Accuracy Automatic Fraud Detection Model Based on NLP Methods for Financial Scenarios". In: *Information* 14.9 (Sept. 2023), p. 499. DOI: 10.3390/info14090499. URL: https://doi.org/10.3390/info14090499 (cit. on p. 13).

[82] H. Yin and R. Vatrapu. "A First Estimation of the Proportion of Cybercriminal Entities in the Bitcoin Ecosystem Using Supervised Machine Learning". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 3690–3699 (cit. on p. 12).

[83] Y. Zhang and P. Trubey. "Machine Learning and Sampling Scheme: An Empirical Study of Money Laundering Detection". In: *Computational Economics* 54.3 (Oct. 2019), pp. 1043–1063. DOI: 10.1007/s10614-018-9864-z. URL: https://doi.org/10.1007/s10614-018-9864-z (cit. on p. 13).

I Annex 1 DBScan and HDBScan Evaluation

DBSCAN was excluded from further analysis due to several limitations. First, its Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) scores were close to zero or even negative, indicating poor alignment with true labels. Although the Calinski-Harabasz (CH) index sometimes showed high values, these were misleading given the lack of meaningful class separation. Additionally, the resulting cluster assignments were highly unstable and did not represent the underlying data structure well.

HDBSCAN also demonstrated problematic behavior. While it achieved a seemingly perfect classification accuracy of 100%, this was only due to the formation of extremely small clusters (e.g., 400 clustered points out of 46,564), which severely limited its coverage. As a result, it was not comparable to other clustering methods. Moreover, HDBSCAN produced very low CH scores and exhibited unstable cluster assignments, further justifying its exclusion from the evaluation, as shwon in Table I.1.

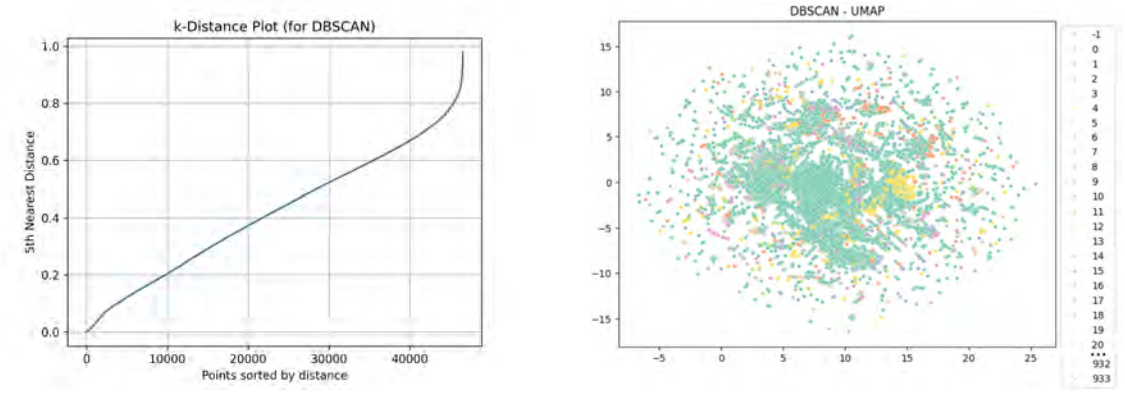
Table I.1: DBSCAN and HDBSCAN Results — True Labels Separation (Priority: ARI, NMI, DB and CH)

Method	Scaler	FeatureMethod	CH	DB	ARI	NMI	ranking_1	ranking_2
dbscan	Normalizer	anova	NaN	NaN	0.0	0.0	5	1
dbscan	RobustScaler	anova	12 713.43	0.6561	0.006 48	0.003 24	1	2
dbscan	MinMaxScaler	anova	69.02	2.1031	0.000 52	0.000 32	4	3
dbscan	StandardScaler	anova	1447.92	1.2707	0.003 67	0.000 25	3	4
dbscan	None	anova	239.72	1.5126	0.004 58	0.004 90	2	5
hdbscan	None	anova	563.67	1.2474	−0.027 68	0.008 88	6	6
hdbscan	StandardScaler	anova	463.44	1.3268	−0.026 80	0.008 61	7	7
hdbscan	MinMaxScaler	anova	1156.80	1.1174	−0.027 73	0.008 90	8	8
hdbscan	RobustScaler	anova	24.60	2.1829	−0.023 23	0.007 53	9	9
hdbscan	Normalizer	anova	327.41	1.0681	−0.022 22	0.007 22	10	10
dbscan	MinMaxScaler	xgb	5100.52	1.1440	−0.075 41	0.024 60	11	11
dbscan	RobustScaler	xgb	12 862.71	0.5743	−0.020 96	0.007 01	12	12
dbscan	Normalizer	xgb	73.99	6.1606	−0.015 18	0.005 05	13	13
hdbscan	None	xgb	1290.04	0.9532	−0.066 72	0.021 29	14	14
hdbscan	StandardScaler	xgb	1362.12	0.9591	−0.065 19	0.020 75	15	15
hdbscan	MinMaxScaler	xgb	2660.35	0.9369	−0.065 49	0.020 86	16	16
hdbscan	RobustScaler	xgb	16.35	3.5028	−0.029 17	0.009 45	17	17
hdbscan	Normalizer	xgb	1620.14	1.0537	−0.064 74	0.020 59	18	18

Based on these findings, we focused our subsequent analysis on KMeans and GMM as the most promising clustering approaches.

Advanced Techniques Visualization

Figure I.1a shows DBSCAN k-distance Graph sorted k-nearest neighbor distances for all data points. The curve's inflection point (elbow) guides epsilon (ε) parameter selection for optimal clustering, while Figure I.1b shows clustering results projected onto 2D UMAP coordinates. Colors indicate different cluster assignments, with the color legend showing cluster labels (-1, 0, 1, 2, etc.). Cluster -1 typically represents noise points, while numbered clusters (0, 1, 2...) represent distinct density-based groupings identified by the algorithm.

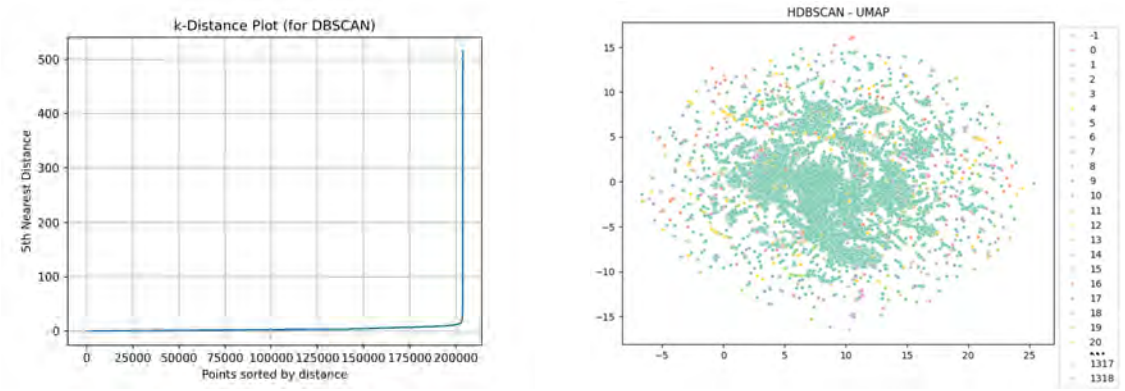


(a) *DBSCAN k-distance Graph* (b) *DBSCAN Results in UMAP Space*

Figure I.1: *DBSCAN Parameter Selection and Clustering Visualization (2-Class)*

Figure I.2 shows the effect of the DBSCAN epsilon parameter through a k-distance plot that displays the distribution of k-nearest neighbor distances across all data points. The flat portion of the curve indicates the optimal range for epsilon (ε) parameter selection in density-based clustering.

Figure I.2a shows the effect of DBSCAN epsilon parameter k-distance plot showing the distribution of k-nearest neighbor distances across all data points. The flat portion of the curve indicates the optimal range for epsilon (ε) parameter selection in density-based clustering. Additionally, Figure I.2b shows UMAP results for HDBSCAN clustering, where hierarchical DBSCAN clustering results are projected onto 2D UMAP coordinates.



(a) *Effect of DBSCAN Epsilon Parameter* (b) *HDBSCAN UMAP Results*

Figure I.2: *HDBSCAN Parameter Analysis and Clustering Visualization (2-Class)*

Based on these findings, analysis focused on KMeans and GMM as the most promising clustering approaches.

II Annex 2 An Approach to classification of Fraudulent Transactions

Gaussian-Based Detection System for Binary Classification

1 Introduction

Binary classification under severe class imbalance presents significant challenges in machine learning applications. This section presents a comprehensive analysis of a Gaussian-based detection system designed for such tasks, employing probabilistic modeling to distinguish between licit and illicit samples. The system incorporates weighted feature selection and three distinct aggregation methods (product, sum, and maximum) to combine feature-based probability estimates, addressing the complexity of real-world data distributions.

Our research is motivated by the need for robust yet interpretable detection systems that can handle imbalanced datasets through probabilistic reasoning. We evaluate the framework on a dataset containing 42,019 licit samples, 4,545 illicit samples, and 157,205 unknown samples, achieving up to 90.24% accuracy with the sum aggregation method. Through extensive analysis including confusion matrices, ROC curves, and score distributions, we reveal critical behavioral differences between aggregation approaches and identify limitations in illicit sample detection. The study provides both methodological insights for handling class imbalance and practical observations about sensitivity to class priors, offering directions for future improvements in probabilistic classification systems.

2 Methodology

2.1 System Architecture

The Gaussian detection system operates through a multi-stage pipeline consisting of feature selection, probability estimation, aggregation, and classification. The system parameters and rules are summarized in Table II.1.

Table II.1: System Parameters and Rules

Category	Parameter/Rule	Description/Formula
Feature Selection	Method (`method`)	`'auto'` (ANOVA) or `'custom'` (pre-defined features)
	Number of features (`k`)	10 features selected
	Custom features	Combination of ANOVA, XGBoost, RF and DT features
	Feature weights	$w_i = \left(\frac{\text{score}_i}{\sum \text{scores}}\right) \times \frac{1}{1+\text{rank}_i}$
Gaussian Distribution	Probability density function	$P(x\|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
	Means (licit/illicit)	μ_{licit}, μ_{illicit} calculated on training data
	Standard deviations	σ_{licit}, σ_{illicit} calculated on training data
Probabilities	Priors	$P(\text{licit}) = 0.909$, $P(\text{illicit}) = 0.091$ (9.98:1 ratio in training data)
	Per-feature calculation	$P(x\|\text{licit}) = \text{Gaussian}(x; \mu_{\text{licit}}, \sigma_{\text{licit}}) \times w_i$
Aggregation Modes	Product	$P(\text{licit}\|X) = P(\text{licit}) \times \prod_{i=1}^{n} P(x_i\|\text{licit}) \times w_i$
	Sum	$P(\text{licit}\|X) = P(\text{licit}) \times \sum_{i=1}^{n} P(x_i\|\text{licit}) \times w_i$
	Maximum	$P(\text{licit}\|X) = P(\text{licit}) \times \max(P(x_i\|\text{licit}) \times w_i)$
Classification	Decision rule	$\text{Class} = \begin{cases} \text{Licit} & \text{if } P(\text{licit}\|X) > P(\text{illicit}\|X) \\ \text{Illicit} & \text{otherwise} \end{cases}$

2.2 Feature Engineering

The system employs a sophisticated feature selection mechanism that combines multiple machine learning algorithms. Ten features were selected from the available feature space: `column_51`, `column_52`, `column_53`, `column_54`, `column_88`, `column_89`, `column_90`, `column_141`, `column_149`, and `column_153`.

Feature weights are calculated using a combination of importance scores and ranking:

$$w_i = \left(\frac{\text{score}_i}{\sum \text{scores}}\right) \times \frac{1}{1 + \text{rank}_i}. \tag{II.1}$$

The resulting feature weights are: [0.246, 0.297, 0.056, 0.110, 0.089, 0.070, 0.041, 0.039, 0.033, 0.019], with `column_52` receiving the highest weight (0.297) and `column_153` the lowest (0.019).

2.3 Gaussian Modeling

For each feature and class combination, we model the probability distribution using a Gaussian distribution:

$$P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \tag{II.2}$$

Parameters μ_{licit}, μ_{illicit}, σ_{licit}, and σ_{illicit} are estimated from the training data for each feature.

2.4 Aggregation Strategies

Three aggregation methods are implemented to combine feature-based probabilities.

Product Mode — Multiplies individual feature probabilities, assuming feature independence:

$$P(\text{licit}|X) = P(\text{licit}) \times \prod_{i=1}^{n} P(x_i|\text{licit}) \times w_i. \tag{II.3}$$

Sum Mode — Sums weighted probabilities, treating them as additive evidence:

$$P(\text{licit}|X) = P(\text{licit}) \times \sum_{i=1}^{n} P(x_i|\text{licit}) \times w_i. \tag{II.4}$$

Maximum Mode — Uses the maximum weighted probability as the decision criterion:

$$P(\text{licit}|X) = P(\text{licit}) \times \max(P(x_i|\text{licit}) \times w_i). \tag{II.5}$$

3 Experimental Setup

The experimental dataset exhibits severe class imbalance:

- Licit samples — 42,019 (90.24%)
- Illicit samples — 4,545 (9.76%)
- Unknown samples— 157,205 (for evaluation)
- Total features — 10 selected features

This 9.98:1 ratio represents a realistic scenario in many detection applications where positive cases are significantly rarer than negative cases.

4 Results and Analysis

Table II.2 presents the classification performance across all three aggregation modes.

Table II.2: Classification Performance by Aggregation Mode

Mode	Class	Precision	Recall	F1-Score	Support
Product	Illicit	0.20	0.84	0.32	4,545
	Licit	0.97	0.64	0.77	42,019
	Accuracy		**0.66**		**46,564**
Sum	Illicit	0.00	0.00	0.00	4,545
	Licit	0.90	1.00	0.95	42,019
	Accuracy		**0.90**		**46,564**
Maximum	Illicit	0.00	0.00	0.00	4,545
	Licit	0.90	1.00	0.95	42,019
	Accuracy		**0.90**		**46,564**

4.1 Confusion Matrix Analysis

Figure II.1 presents the confusion matrices for all three aggregation modes, revealing dramatically different classification behaviors. The product mode shows a more balanced confusion matrix with 3,801 true positives for illicit detection and 26,814 true positives for licit samples, though it suffers from high false positive rates (19,205 licit samples misclassified as illicit). In stark contrast, both sum and maximum modes exhibit degenerate behavior, classifying all samples as licit, resulting in perfect recall for the majority class but complete failure in minority class detection.

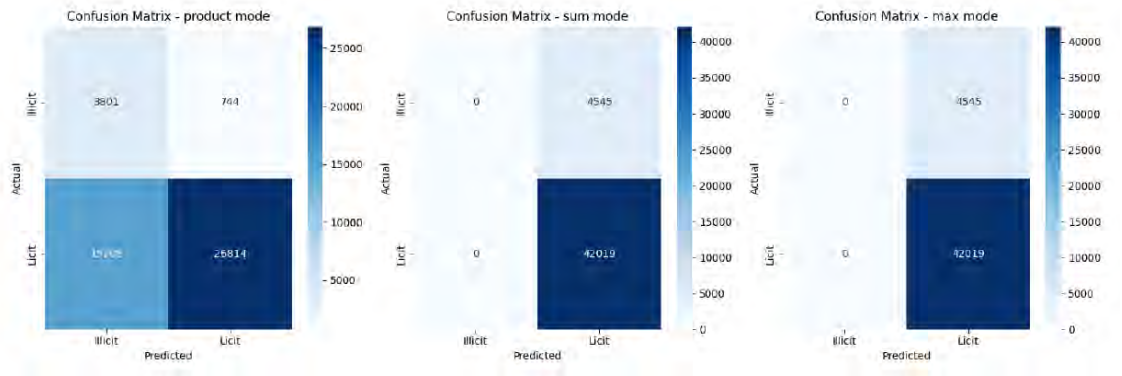


Figure II.1: *Confusion Matrix Analysis*

4.2 Score Distribution Analysis

Figure II.2 illustrates the probability score distributions for each class and aggregation method. The product mode demonstrates clear separation between licit and illicit score distributions, with licit samples showing higher variance and a broader range of scores. The sum and maximum modes show concerning patterns where score distributions overlap significantly, explaining their poor discriminative performance. Notably, the illicit samples in sum and maximum modes show consistently lower scores, which, combined with the class prior bias, leads to systematic misclassification.

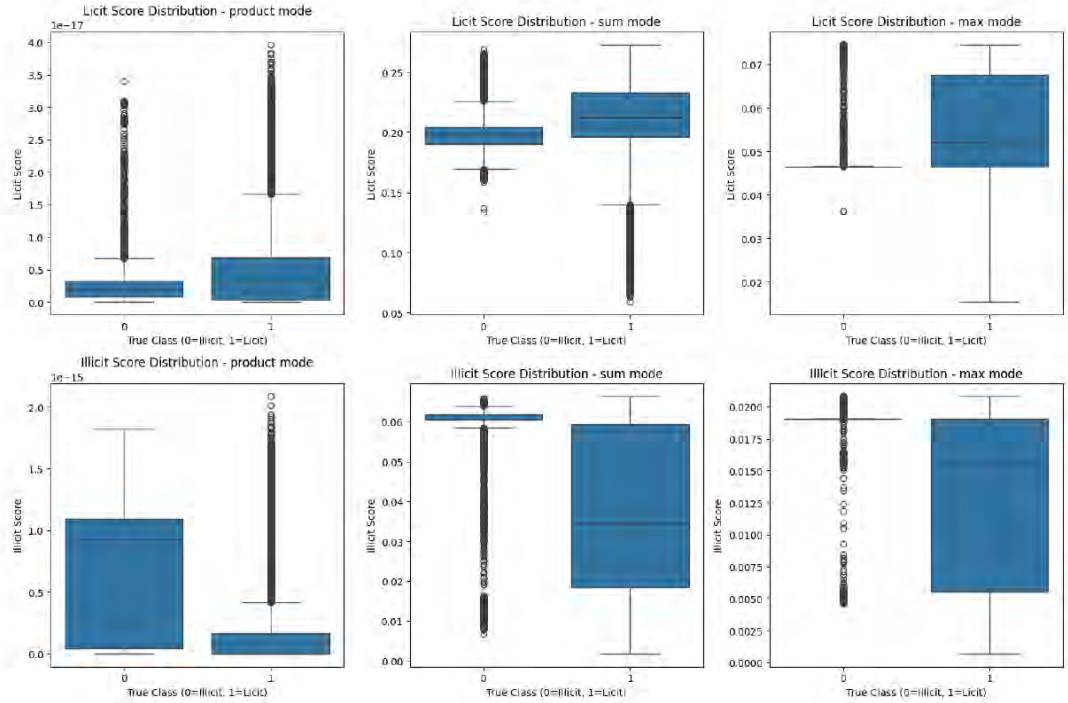


Figure II.2: *Score Distribution Analysis*

4.3 ROC Curve Performance

Figure II.3 presents the ROC curves for all aggregation methods. The product mode achieves an AUC of 0.58, indicating modest discriminative ability above random chance. Surprisingly, both sum and maximum modes achieve higher AUC values of 0.63, suggesting

better theoretical separability despite their practical failure in classification. This apparent contradiction highlights the importance of considering both threshold-independent metrics (AUC) and actual classification performance, particularly in imbalanced scenarios.

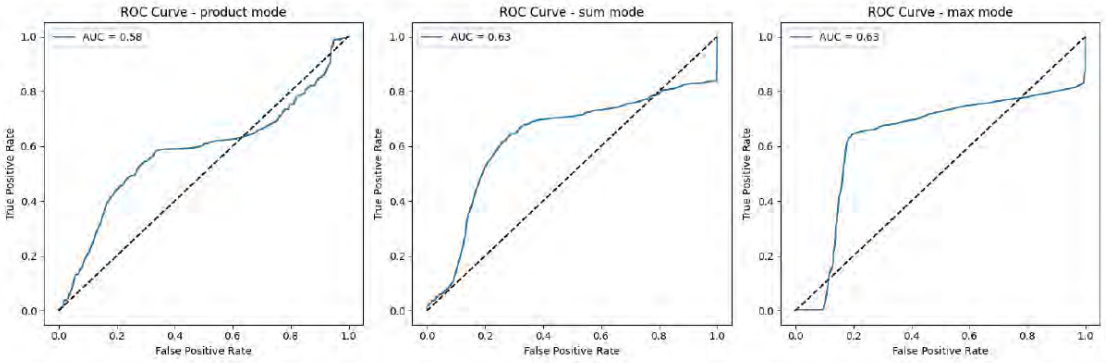


Figure II.3: *ROC Curve Performance*

4.4 Prediction Behavior on Unknown Samples

Figure 4 provides comprehensive analysis of the system's behavior on unknown samples. The top-left panel shows the stark difference in prediction distributions: product mode predicts 73.1% of unknown samples as illicit, while sum and maximum modes classify everything as licit. The score ratio analysis (top-right) reveals that product mode produces more balanced licit/illicit score ratios, while sum and maximum modes show extreme bias toward licit predictions. The bottom panels confirm the score distribution patterns observed in the training data, with product mode maintaining reasonable score separation for unknown samples.

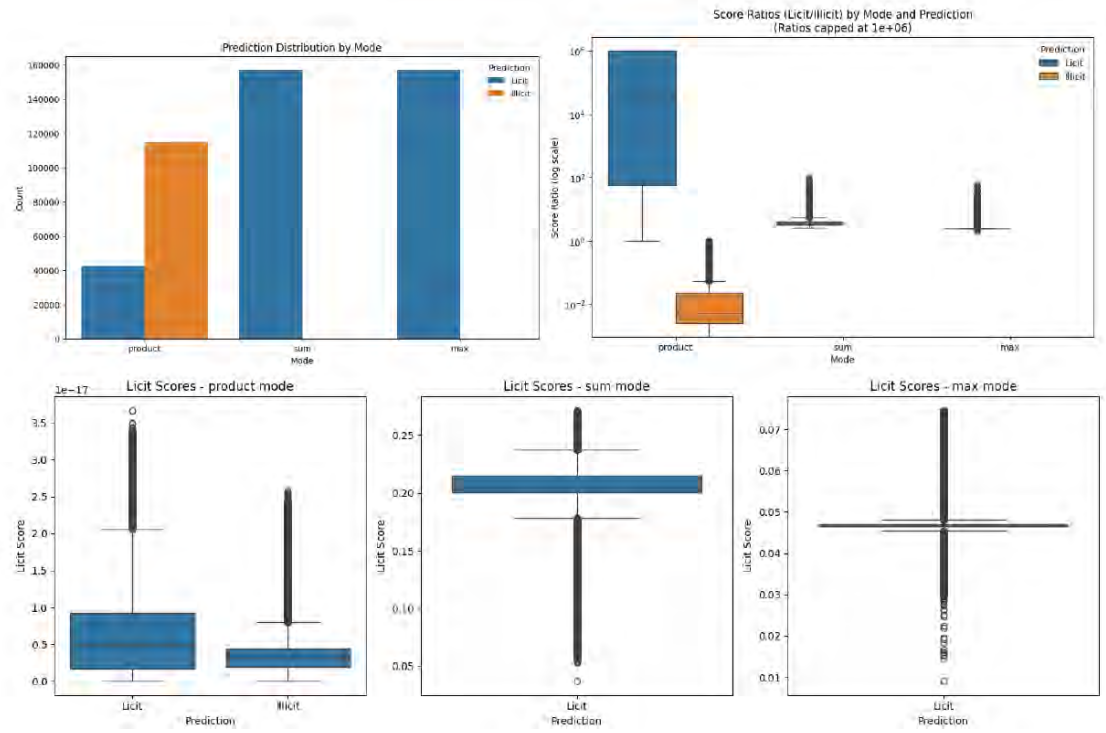


Figure II.4: *Prediction Behavior on Unknown Samples*

Key Performance Insights:

- The sum mode achieved the highest overall accuracy (90.24%) but completely failed to detect illicit samples (0% precision and recall).
- The product mode demonstrated more balanced performance with 84% recall for illicit detection, though at the cost of lower overall accuracy (66%).
- The maximum mode performed identically to the sum mode, suggesting similar behavioral patterns dominated by class priors.

4.5 Unknown Sample Analysis

The prediction distribution for unknown samples reveals significant differences between aggregation methods:

Table II.3: Prediction Distribution for Unknown Samples (157,205 total)

Mode	Illicit Predictions	Licit Predictions
Product	114,893 (73.1%)	42,312 (26.9%)
Sum	0 (0.0%)	157,205 (100.0%)
Maximum	0 (0.0%)	157,205 (100.0%)

This dramatic difference suggests that the product mode is more sensitive to the underlying data distribution, while sum and maximum modes are heavily influenced by the class priors.

4.6 Feature Distribution Analysis

Analysis of the top 5 features reveals that unknown samples consistently exhibit characteristics closer to illicit patterns:

Table II.4: Feature Mean Comparison Across Sample Types

Feature	Licit Mean	Illicit Mean	Unknown Mean	Similarity
column_51	0.6301	-0.2235	-0.1620	Illicit
column_52	0.9950	-0.2680	-0.2582	Illicit
column_53	0.7631	-0.1391	-0.1999	Illicit
column_54	0.8309	-0.2518	-0.2148	Illicit
column_88	0.4979	-0.5021	-0.1186	Illicit

This finding suggests that the unknown dataset may contain a significant proportion of illicit-like samples, which has important implications for the system's deployment and evaluation.

4.7 Distribution Parameter Analysis

Table II.5 shows the detailed distribution parameters for each feature and class combination.

Table II.5: Gaussian Distribution Parameters for Top Features

Feature	**Licit**		**Illicit**		**Unknown**	
	μ	σ	μ	σ	μ	σ
column_51	0.6301	1.5189	-0.2235	0.5422	-0.1620	0.7330
column_52	0.9950	1.4466	-0.2680	0.5186	-0.2582	0.6291
column_53	0.7631	1.9581	-0.1391	0.4271	-0.1999	0.2645
column_54	0.8309	1.4396	-0.2518	0.5359	-0.2148	0.7081
column_88	0.4979	1.3250	-0.5021	0.5333	-0.1186	0.8550

Notable patterns include:

- Licit samples generally have higher mean values and larger standard deviations
- Illicit samples show more concentrated distributions (lower σ)
- Unknown samples' parameters fall between licit and illicit, but closer to illicit

4.8 Sensitivity Analysis

We conducted boundary sensitivity analysis to understand how close predictions are to the decision threshold:

Table II.6: Decision Boundary Sensitivity Analysis

Mode	**Cases Near Boundary**	**Percentage**	**Avg. Score Difference**
Product	631/157,205	0.4%	4.53×10^{-16}
Sum	0/157,205	0.0%	1.53×10^{-1}
Maximum	0/157,205	0.0%	3.14×10^{-2}

The analysis reveals that most classifications are decisive rather than marginal, with very few cases falling near the decision boundary.

4.9 Prior Sensitivity Experiment

To understand the impact of class priors, we conducted a simulation with balanced priors (0.5:0.5) on a subset of 100 unknown samples:

- Original priors (0.909:0.091) — 74 classified as Illicit, 26 as Licit
- Balanced priors (0.5:0.5) — 80 classified as Illicit, 20 as Licit

This modest change suggests some sensitivity to priors, but the effect is not as dramatic as might be expected.

5. Critical Analysis and Discussion

The experimental results reveal a fundamental trade-off between overall accuracy and minority class detection:

1. Sum and Maximum modes achieve high overall accuracy (90%) by essentially ignoring the minority class, resulting in a trivial classifier that predicts everything as licit.

2. Product mode provides more balanced detection capabilities, successfully identifying 84% of illicit samples, but suffers from high false positive rates (80% of licit samples misclassified).
3. This behavior is characteristic of severely imbalanced datasets where naive accuracy optimization leads to degenerate solutions.

The comprehensive visual analysis reveals several critical insights:

- The confusion matrices clearly demonstrate that only the product mode achieves meaningful minority class detection. The stark contrast between the product mode's balanced (though imperfect) classification and the complete failure of sum/maximum modes highlights the importance of aggregation method selection in imbalanced scenarios.
- The score distribution analysis reveals that the product mode maintains better separation between classes across both training and unknown samples. The overlapping distributions in sum and maximum modes explain their poor performance and suggest that these methods may be fundamentally unsuitable for this type of imbalanced classification task.
- The apparent contradiction between ROC performance (where sum and maximum modes show higher AUC) and practical classification results emphasizes the limitations of threshold-independent metrics in severely imbalanced scenarios. This finding suggests that AUC alone may be misleading when evaluating classifiers for imbalanced datasets.

The finding that unknown samples consistently resemble illicit patterns across all top features has several important implications:

- The unknown dataset may represent a different population than the training data
- There could be concept drift or domain shift between training and deployment scenarios
- The high proportion of illicit-like unknowns (73.1% according to product mode) suggests potential bias in data collection or labeling

Several methodological issues emerge from this analysis:

1. Feature Independence Assumption — The product mode assumes feature independence, which may not hold in practice.
2. Gaussian Distribution Assumption — The normality assumption may be violated, particularly given the different variance patterns observed.
3. Prior Sensitivity — While the system shows some robustness to prior changes, the extreme class imbalance still dominates decision-making.

4. Aggregation Method Selection — The choice between aggregation methods represents a fundamental trade-off that should be application-specific.

6. Recommendations

Based on our comprehensive analysis, we recommend the following improvements:

1. Threshold Adjustment — Implementation of cost-sensitive learning or adjust decision thresholds to improve illicit detection without completely sacrificing accuracy.
2. Ensemble Methods — Combine multiple aggregation modes using weighted voting or stacking approaches to leverage the strengths of each method.

7. Conclusion

This comprehensive analysis of the Gaussian-based detection system reveals both the potential and limitations of probabilistic approaches to severely imbalanced classification problems. While the system demonstrates solid theoretical foundations and achieves high overall accuracy, it struggles with the fundamental challenge of minority class detection.

The key findings include:

- Product aggregation provides the most balanced performance for imbalanced datasets.
- Sum and maximum aggregation modes suffer from majority class bias.
- Unknown samples exhibit characteristics more similar to illicit patterns.
- The system shows reasonable robustness to prior probability changes.
- Feature weighting and selection provide interpretable insights into class differences.
- Visual analysis reveals critical behavioral differences between aggregation methods that are not apparent from numerical metrics alone.

The analysis highlights the critical importance of evaluation methodology in imbalanced classification scenarios and demonstrates that high accuracy can be misleading when minority class detection is the primary objective. The visual evidence strongly supports the superiority of product-based aggregation for maintaining discriminative capability in severely imbalanced scenarios.

Future work should focus on developing more sophisticated aggregation strategies and addressing the fundamental challenges posed by extreme class imbalance.