NETWORK ANALYTICS FOR ANTI-MONEY LAUNDERING – A SYSTEMATIC LITERATURE REVIEW AND EXPERIMENTAL EVALUATION

A PREPRINT

Bruno Deprez[*]
KU Leuven
University of Antwerp - imec

Toon Vanderschueren
KU Leuven
University of Antwerp - imec

Bart Baesens
KU Leuven
University of Southampton

Tim Verdonck
University of Antwerp - imec
KU Leuven

Wouter Verbeke
KU Leuven

ABSTRACT

Money laundering presents a pervasive challenge, burdening society by financing illegal activities. The use of network information is increasingly being explored to effectively combat money laundering, given it involves connected parties. This led to a surge in research on network analytics for anti-money laundering (AML). The literature is, however, fragmented and a comprehensive overview of existing work is missing. This results in limited understanding of the methods to apply and their comparative detection power. This paper presents an extensive and unique literature review, based on 97 papers from *Web of Science* and *Scopus*, resulting in a taxonomy following a recently proposed fraud analytics framework. We conclude that most research relies on expert-based rules and manual features, while deep learning methods have been gaining traction. This paper also presents a comprehensive framework to evaluate and compare the performance of prominent methods in a standardized setup. We compare manual feature engineering, random walk-based, and deep learning methods on two publicly available data sets. We conclude that (1) network analytics increases the predictive power, but caution is needed when applying GNNs in the face of class imbalance and network topology, and that (2) care should be taken with synthetic data as this can give overly optimistic results. The open-source implementation facilitates researchers and practitioners to extend this work on proprietary data, promoting a standardized approach for the analysis and evaluation of network analytics for AML.

Keywords Fraud Analytics · Anti-Money Laundering · Network Analytics · Literature Review

1 Introduction

Money laundering is the process of concealing illegally obtained funds by passing them through a complex sequence of transactions, so the money can be used to fund further criminal activities, e.g., drug trafficking, and terrorism [75]. The United Nations Office on Drugs and Crime [126] estimates that around 2% to 5% of global GDP is laundered worldwide, amounting yearly to USD 2 trillion.

In contrast to credit card fraud, money laundering occurs over a longer duration and requires analysis of multiple transactions for detection. It is actively concealed, since sustained laundering is needed to process continuous money streams. Credit card fraud, on the other hand, aims to get as much money out as fast as possible. Hence, the specific characteristics of money laundering merit dedicated research attention.

[*]Correspondening author: bruno.deprez@kuleuven.be

Since this money needs to enter the financial system at some point [75], legislators have put stringent rules on financial institutions [15] for reporting suspicious transactions. Despite record fines and mounting pressure by regulators, money laundering represents a growing problem [135].

Money laundering typically involves three stages [126, 75]: placement, where illicit funds enter the financial system; layering, where transactions obscure their origin; and integration, where the funds re-enter the economy through seemingly legitimate means, completing the laundering process.

Anti-money laundering practices require constant monitoring of clients and transactions. These practices are, however, largely ineffective [35], with Europol estimating only 1.1% of criminal profits in the EU being confiscated [38].

Effective money laundering involves multiple parties, with individual transactions appearing normal [15]. Therefore, network analytics is essential [15, 10]. This need was already mentioned three decades ago by Senator et al. [113], where the authors used visualisations to support expert investing.

The increasing attention of the past years has resulted in a growing body of literature on network analytics for anti-money laundering, but simultaneously resulted in a lack of comprehensive overviews. Therefore, as a first step, this paper aims to supplement the literature by providing a comprehensive literature review of network analytics (NA) for anti-money laundering, covering 97 papers analysed according to multiple dimensions.

Due to the fragmented nature of the literature there is limited insight into which methods perform best. This fragmentation and lack of comparison is apparent in the fact that many of the introduced methods are not tested against any baselines. Furthermore, researchers are slow to adopt the latest network analytics methods and graph neural network in anti-money laundering. Therefore, the second goal of this work is to present a structured experimental set-up to evaluate the most prominent methods.

In summary, our main contributions are four-fold:

- We provide an extensive and critical literature review on network analytics for anti-money laundering, to gain comprehensive and deep insights, and provide directions for future research.
- We benchmark a range of state-of-the-art network analytics methods for anti-money laundering, comparing the performance on two open-source data sets.
- We provide insights into the specific challenges of network methods based on the topology of the network and the data generating process.
- We implement the methods in a uniform manner and facilitate replication of the presented results by providing public access to the code at `https://github.com/B-Deprez/AML_Network`, aiming to promote a standardized approach towards the analysis and adoption of NA for AML and to encourage further research.

This paper is structured as follows. We present the methodology for the literature review in Section 2, with Section 3 presenting the results and analysis. Section 4 describes the the empirical evaluation, with the results presented in Section 5. Overall conclusions and directions for future work are presented in Section 6.

2 Literature Review

The first part of this work covers the current literature. A summary is given according to multiple categories. Additional analyses are provided for the top-cited papers. The aim is to provide answers to the following research questions:

- What are the most prominent learning methods in the literature and how have these evolved over time?
- How are methods evaluated, using what data sets and evaluation metrics?
- What is the setting, objectives and challenges discussed in the top-cited papers?

2.1 Methodology

We conducted the literature search using the queries *"graph analy*" AND "money launder*"* and *"network analy*" AND "money launder*"* on WoS and Scopus, chosen for their high completeness and research credibility [96, 14]. The search is limited to English-language papers published before 2023, excluded recent publications to ensure adequate time for community evaluation.

The selection process included a title and abstract scan, followed by a full-text review. Papers were excluded if they focused on legal aspects or lacked substantial application of network analytics to anti-money laundering (AML). The

final set of papers is classified according to seven criteria as summarised in Table 1: publication data, method type, modelling approach, evaluation metrics, main objectives, data type, and network characteristics.

Our review provides both a broad analysis of trends across all selected papers and an in-depth focus on top-cited works (top 10% annually and overall, per Google Scholar). This dual approach highlights general trends and the state-of-the-art (SOTA) in the field. Finally, we examine relevant review papers to contextualize our findings and demonstrate how our work extends existing literature.

Table 1: The (sub-)categories with explanation for the classification of the literature.

Category	**Sub-Category**	**Definition**
Publication Data	Title	The full title of the paper
	Journal/Conference	Where the paper is published
	Year	The year of publication
	Review Paper	Indicating if it is a review paper
	Bitcoin/Crypto	Indicating if it deals with crypto currency
Method Type	Supervised	Application of supervised method
	Unsupervised	Application of unsupervised method
	Semi-Supervised	Application of semi-supervised method
	Mixed	Methods presented across different learning methods
	Visualisation	Application of visualisation method
Modelling Method	Rule-based	Application of specific rules or cut-offs
	Manual features	Network features having a exact definition
	Walk-based	Network analysis based on (random) walks
	Shallow representation	Embeddings typically through matrix factorization or random walks, without employing deep neural networks
	Deep representation	Embeddings based on deep neural networks, e.g., GNNs
	Correlation-based	Analysis of correlation of network features with target
	Logistic regression	Application of logistic regression
	Tree-based	Application of tree-based methods
	SVM-based	Application of support vector machines
	Neural networks	Learning using neural networks
	Anomaly detection	Application of anomaly or outlier detection
	Clustering	Features based on community detection or clustering of features
Evaluation Metric	Accuracy	Accuracy
	Precision	The precision
	Recall	The recall
	F1	The F1 or micro-F1 score
	TPR	The true positive rate
	FPR	The false positive rate
	AUC-ROC	The area under the ROC curve
	AUC-PR	The area under the precision-recall curve
	Time	The execution time
Objective	Client classification	Detection of suspicious entities
	Transaction classification	Detection of suspicious transactions
	Community detection	Detection of suspicious groups of clients
	Flow/Chain detection	Detection of suspicious combination of transactions
Data	Proprietary	Usage of confidential data
	Open-source	Usage of freely available data
	Synthetic	Usage of synthetically generated data
Network	Multiple	Usage of multiple networks
	Inverse/undirected	Additional use of network with directions removed or reverted

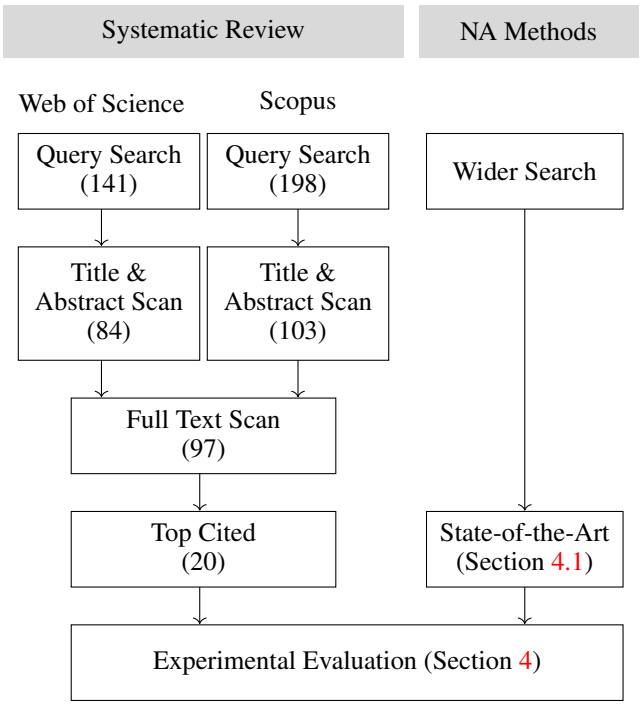


Figure 1: The search and filtering method with the number of remaining papers between brackets.

2.2 Results

Figure 1 summarizes the initial search results. After filtering and removing duplicates, 97 papers remained, of which 11 were review papers. Classification based on Table 1 is detailed in Table 2 and Table 3, with review papers excluded from category-specific analysis due to their broader scope.

Table 2: **Literature classification.** Classification of the papers according to different categories: publication data, learning method, evaluation metric, objective, and data.

	Publication Data			**Evaluation Metric**									**Objective**				**Data**			**Network**	
Reference	*Year*	*Review Paper*	*Bitcoin/Crypto*	*Accuracy*	*Precision*	*Recall*	*F1*	*TPR*	*FPR*	*AUC-ROC*	*AUC-PR*	*Time*	*Client Classification*	*Transaction Classification*	*Community Detection*	*Flow/Chain Detection*	*Proprietary*	*Open-Source*	*Synthetic*	*Inverse/undirected network*	*Multiple networks*
Total		11	26	9	15	18	17	4	5	10	1	11	37	27	13	16	32	40	11	5	18
Drężewski et al. [33]	2015	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
Ngai et al. [99]	2011	✓	✗																		
Malm and Bichler [87]	2013	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Fronzetti Colladon and Remondi [41]	2017	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓
Zhou et al. [151]	2017	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
Didimo et al. [29]	2011	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗
McGinn et al. [88]	2016	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
Senator et al. [113]	1995	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
Wu et al. [137]	2021	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓
Gao and Ye [42]	2007	✓	✗																		
Joaristi et al. [68]	2018	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓
Irwin and Turner [63]	2018	✓	✓																		
Michalak and Korczak [93]	2011	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗
Wu et al. [138]	2021	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Li et al. [78]	2020	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗
Phetsouvanh et al. [105]	2018	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓

Continued on next page

Table 2 (continued)

Reference	Year	Publication Data: Review Paper	Publication Data: Bitcoin/Crypto	Evaluation Metric: Accuracy	Evaluation Metric: Precision	Evaluation Metric: Recall	Evaluation Metric: F1	Evaluation Metric: TPR	Evaluation Metric: FPR	Evaluation Metric: AUC-ROC	Evaluation Metric: AUC-PR	Evaluation Metric: Time	Objective: Client Classification	Objective: Transaction Classification	Objective: Community Detection	Objective: Flow/Chain Detection	Data: Proprietary	Data: Open-Source	Data: Synthetic	Network: Inverse/undirected network	Network: Multiple networks
Zhou et al. [154]	2017	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Didimo et al. [31]	2019	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Joaristi et al. [69]	2019	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓
Camino et al. [19]	2017	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗
Velasco et al. [131]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Nandhini and Das [97]	2016	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
Zhdanova et al. [150]	2014	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
Imanpour et al. [62]	2019	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
Turner et al. [124]	2019	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
Ovelgönne et al. [101]	2012	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Magalingam et al. [84]	2015	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✓
Helmy et al. [59]	2016	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
Xueshuo et al. [145]	2021	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗
Plaksiy et al. [107]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓
Garcia-Bedoya et al. [44]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗
Mehmet and Wijesekera [90]	2013	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
Xia et al. [140]	2021	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗
Kute et al. [74]	2021	✓	✗																		
Mahootiha et al. [86]	2021	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Shaikh et al. [115]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗
Turner et al. [125]	2020	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
Ricci [110]	2020	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
Bahulkar et al. [11]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗
Chmielewski and Stkapor [24]	2017	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Zhou et al. [152]	2022	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓
Gasanova et al. [46]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Magomedov et al. [85]	2018	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Tharani et al. [121]	2021	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓
Jin and Qu [67]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
Mehmet et al. [91]	2014	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Xiao et al. [141]	2018	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Mohan et al. [95]	2022	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Alarab et al. [4]	2020	✓	✓																		
Shen et al. [116]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓
Prado-Romero and Gago-Alonso [108]	2017	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
Zhang et al. [149]	2021	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Garcia-Bernardo et al. [45]	2022	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
He et al. [58]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Micale et al. [92]	2019	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
Bhalerao et al. [13]	2019	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓
Omar et al. [100]	2014	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Winiecki et al. [136]	2020	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓	✓
Starnini et al. [118]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Hamid [55]	2017	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Gulino et al. [53]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
Adriaens et al. [1]	2019	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗
Chang et al. [22]	2007	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Kurshan and Shen [73]	2020	✓	✗																		
Jin et al. [65]	2022	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
Bellomarini et al. [12]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Sousa Lima et al. [117]	2022	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗
Çağlayan and Bahtiyar [17]	2022	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Jin et al. [66]	2022	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Kurniawan et al. [72]	2015	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Sun et al. [119]	2022	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Gerbrands et al. [47]	2022	✓	✗																		
Dumitrescu et al. [34]	2022	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Yu et al. [148]	2022	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
García and Mateos [43]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
Wang et al. [133]	2022	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗
Gong et al. [48]	2022	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
Li et al. [79]	2022	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Li et al. [76]	2022	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗

Continued on next page

Table 2 (continued)

Reference	Publication Data			Evaluation Metric									Objective				Data			Network	
	Year	Review Paper	Bitcoin/Crypto	Accuracy	Precision	Recall	F1	TPR	FPR	AUC-ROC	AUC-PR	Time	Client Classification	Transaction Classification	Community Detection	Flow/Chain Detection	Proprietary	Open-Source	Synthetic	Inverse/undirected network	Multiple networks
Assumpção et al. [8]	2022	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Semenov et al. [112]	2017	✓	✗																		
Didimo et al. [30]	2012	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Cardoso et al. [20]	2022	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Day [27]	2021	✓	✓																		
Alarab et al. [5]	2020	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Badalov et al. [9]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Korczak and Łuszczyk [71]	2011	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Lokanan [82]	2022	✓	✗																		
Rathore et al. [109]	2022	✗	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Liu et al. [81]	2012	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Xiong and Zhong [143]	2020	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Shaikh and Nazir [114]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Yang et al. [146]	2022	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Xiong and Xiao [142]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Mehmet and Wijesekera [89]	2013	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗
Cheong and Si [23]	2010	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Han et al. [57]	2021	✓	✓																		

Table 3: **Literature classification.** Classification of the papers according to the different method building blocks.

Reference	Publication Data		Method Type					Modelling Method											
	Year	Bitcoin/Crypto	Supervised	Unsupervised	Semi-Supervised	Mixed	Visualisation	Rule-Based	Manual Features	Walk-based	Shallow Repr.	Deep Repr.	Correlation-Based	Tree-Based	SVM-Based	Neural Networks	Anomaly Detection	Clustering	Logistic Regression
Total		26	26	40	6	3	20	20	45	15	6	12	3	10	4	6	10	24	6
Dreżewski et al. [33]	2015	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Malm and Bichler [87]	2013	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Fronzetti Colladon and Remondi [41]	2017	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Zhou et al. [151]	2017	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Didimo et al. [29]	2011	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
McGinn et al. [88]	2016	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Senator et al. [113]	1995	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Wu et al. [137]	2021	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Joaristi et al. [68]	2018	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✗
Michalak and Korczak [93]	2011	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Wu et al. [138]	2021	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Li et al. [78]	2020	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Phetsouvanh et al. [105]	2018	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
Zhou et al. [154]	2017	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✓	✓
Didimo et al. [31]	2019	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Joaristi et al. [69]	2019	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Camino et al. [19]	2017	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗
Velasco et al. [131]	2021	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Nandhini and Das [97]	2016	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Zhdanova et al. [150]	2014	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Imanpour et al. [62]	2019	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Turner et al. [124]	2019	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Ovelgönne et al. [101]	2012	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Magalingam et al. [84]	2015	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Helmy et al. [59]	2016	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Xueshuo et al. [145]	2021	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗
Plaksiy et al. [107]	2018	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Garcia-Bedoya et al. [44]	2021	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mehmet and Wijesekera [90]	2013	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Xia et al. [140]	2021	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗

Continued on next page

Table 3 (continued)

	Publication Data		Method Type					Modelling Method											
Reference	*Year*	*Bitcoin/Crypto*	*Supervised*	*Unsupervised*	*Semi-Supervised*	*Mixed*	*Visualisation*	*Rule-Based*	*Manual Features*	*Walk-based*	*Shallow Repr.*	*Deep Repr.*	*Correlation-Based*	*Tree-Based*	*SVM-Based*	*Neural Networks*	*Anomaly Detection*	*Clustering*	*Logistic Regression*
Mahootiha et al. [86]	2021	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Shaikh et al. [115]	2021	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Turner et al. [125]	2020	✓	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗
Ricci [110]	2020	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Bahulkar et al. [11]	2018	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Chmielewski and Stkapor [24]	2017	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Zhou et al. [152]	2022	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Gasanova et al. [46]	2018	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Magomedov et al. [85]	2018	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Tharani et al. [121]	2021	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Jin and Qu [67]	2018	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mehmet et al. [91]	2014	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Xiao et al. [141]	2018	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Mohan et al. [95]	2022	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗
Shen et al. [116]	2021	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Prado-Romero and Gago-Alonso [108]	2017	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Zhang et al. [149]	2021	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
Garcia-Bernardo et al. [45]	2022	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
He et al. [58]	2021	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Micale et al. [92]	2019	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Bhalerao et al. [13]	2019	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓
Omar et al. [100]	2014	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Winiecki et al. [136]	2020	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Starnini et al. [118]	2021	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Hamid [55]	2017	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Gulino et al. [53]	2021	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Adriaens et al. [1]	2019	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Chang et al. [22]	2007	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Jin et al. [65]	2022	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Bellomarini et al. [12]	2021	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗
Sousa Lima et al. [117]	2022	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Çağlayan and Bahtiyar [17]	2022	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	✗
Jin et al. [66]	2022	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓
Kurniawan et al. [72]	2015	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Sun et al. [119]	2022	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Dumitrescu et al. [34]	2022	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
Yu et al. [148]	2022	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
García and Mateos [43]	2021	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Wang et al. [133]	2022	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Gong et al. [48]	2022	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Li et al. [79]	2022	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗
Li et al. [76]	2022	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Assumpção et al. [8]	2022	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Didimo et al. [30]	2012	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Cardoso et al. [20]	2022	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗
Alarab et al. [5]	2020	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Badalov et al. [9]	2018	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Korczak and Łuszczyk [71]	2011	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Rathore et al. [109]	2022	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Liu et al. [81]	2012	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Xiong and Zhong [143]	2020	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Shaikh and Nazir [114]	2018	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Yang et al. [146]	2022	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
Xiong and Xiao [142]	2021	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mehmet and Wijesekera [89]	2013	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Cheong and Si [23]	2010	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Figure 2 illustrates the increasing trend in AML-related research. Table 4 lists journals and conferences with more than one relevant publication, covering 28 papers; the remaining 69 papers were distributed across other venues with a single publication each. The journal with most publications is the Journal of Money Laundering Control, with seven published papers. On second place is the ASONAM conference with three papers on network analytics for AML.

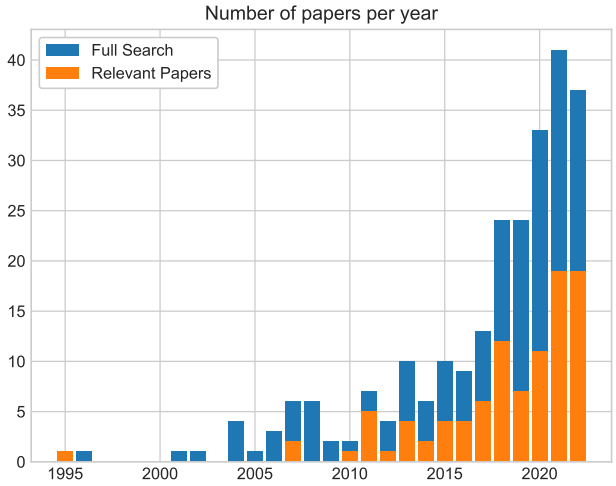


Figure 2: The number of papers incorporated in the full query and after filtering per year.

Table 4: Journals/conferences that have two or more publications in scope of this review.

Journal/Conference Title	**Number**
Journal of Money Laundering Control	7
International Conference on Advances in Social Network Analysis and Mining, ASONAM	3
CEUR Workshop Proceedings	2
IEEE International Conference on Data Mining Workshops	2
IEEE International Conference on Big Data (Big Data)	2
Journal of Physics: Conference Series	2
Federated Conference on Computer Science and Information Systems (FedCSIS)	2
EPJ Data Science	2
Information Sciences	2
IEEE Access	2
International Conference on Machine Learning Technologies	2

Crypto-related research accounts for 26 papers (26.5% of the total), analysed separately in Appendix A due to its distinct AML setting, data characteristics, and anonymity challenges. These differences are evident in the corresponding plots.

Appendix B presents results for the 20 top-cited papers, summarized in Table 6. The category distribution aligns with the overall literature, with 25% focusing on cryptocurrency, closely matching the full-scope percentage.

3 Discussion on the Literature

3.1 Review Papers

We identified 11 review papers, which we discuss first to illustrate how our work complements and extends the existing literature.

Ngai et al. [99], Kurshan and Shen [73] and Lokanan [82] cover general financial fraud, with less emphasis on anti-money laundering or graph learning. Ngai et al. [99] construct a classification framework, but include only one paper on money laundering, which is the only paper on network analytics. Kurshan and Shen [73] give a high-level overview of the general difficulties of machine learning solutions for fighting financial crime. They noted that network-based methods have a lot of potential to process many transactions handled by financial institutions. The paper is closer to a discussion paper than to a systematic literature review. The scope of the review by Lokanan [82] is limited, covering only visualisation methods for outlier detection.

Other review papers deal specifically with AML [74, 112, 42]. Kute et al. [74] give an overview of the methods applied in the literature, but their main topic is interpretability. Semenov et al. [112] discuss the different methods applied to AML. It touches upon different network methods, but it is not a systematic literature review. The framework by Gao and Ye [42] again deals with a wider array of methods, with the authors concluding that network analytics is the most promising. However, the number of papers on network analytics for AML included is limited.

Gerbrands et al. [47] look at the effects of money laundering policies, by measuring the change in network features when a major AML law is introduced. It analyses cluster size and the degree, closeness and betweenness centrality. Hence, the scope is different, since we focus on enhancing anti-money laundering using network features.

A significant number of review papers deal with crypto currency [63, 27, 4, 57]. Irwin and Turner [63] describe challenges in detecting money laundering on the blockchain. This view results in a narrow scope with a total of 45 unique references. These different challenges are partially addressed by Alarab et al. [4] and Han et al. [57]. The experiment by Alarab et al. [4] compares different supervised machine learning methods, with the statistics of the ego-network of the different nodes in the Bitcoin transaction network [37, 134] as features.

Other papers cover a specific money laundering method and propose tailor-made solutions. Han et al. [57] discuss the application of anomaly detection, based on crime-specific transaction patterns. The paper also covers Ponzi schemes and blackmail campaigns. Their scope is more limited, with a total of 33 references. Similarly, Day [27] gives an overview of different characteristics of financial crime, and uses these to construct knowledge graphs for building legal cases.

We extend on prior work by conducting the first systematic review of network analytics methods for anti-money laundering, including both supervised and unsupervised methods. Our scope spans traditional centrality measures to advanced graph neural networks, and addresses money laundering in both fiat and crypto currency.

3.2 Full Scope of Papers

This section summarises the 83 non-review papers. Although network analytics for anti-money laundering has been researched for a long time, Figure 2 illustrates this increasing research interest only after 2010. The average increase in output is 61% year-on-year for the papers in scope.

3.2.1 Methods.

The methods in the literature are supervised, unsupervised, semi-supervised and visualisations. The most popular method is unsupervised learning, followed by supervised learning and visualisations, as shown in Table 3. The popularity of unsupervised learning may be due to money laundering being very uncommon, resulting in highly skewed data sets with very few labelled observations [3]. Due to limited resources to investigate the massive amounts of transactions, most transactions are left unlabelled [129, 106]. In addition, criminals try to evade detection by continuously changing tactics [10, 106], making historically patterns less relevant for future predictions [15]. The performance of (supervised) models trained on historical data can therefore decrease over time [14].

Table 3 indicates that half of the papers rely on manual feature engineering. These manual features include classic network centrality measures (degree, betweenness etc.) and summary statistics of the transactions in the nodes' ego-network. The second most popular method is clustering, which is used broadly in two ways. First, clustering for visualisations mitigates visual cluttering and allows the the selection detail shown. Second, clusters are used for feature engineering. Next to manual feature engineering and clustering, rule-based methods are still frequently used. This illustrates that the current literature heavily relies on expert knowledge. Hence, there is an opportunity to further develop data-driven machine learning methods in anti-money laundering.

The evolution of methods used over time, as presented in Figure 3, shows that the earliest research relied almost exclusively on rule-based systems and manual feature engineering. These remain popular approaches to this day. From 2018 onward, we see the application of deep learning methods based on shallow representations and graph neural networks. Especially the use of GNNs has seen a spike in 2022.

The downstream methods used for classification have also evolved. Popular methods are tree-based, clustering, anomaly detection and logistic regression. SVM-based methods were popular until 2019, while neural networks were only used in AML as of 2020. These are often recurrent neural networks to incorporate the temporal aspect [140, 149, 95, 79].

Key to these methods is how the authors define the network(s) used. The construction of multiple networks, e.g., same entities but using different relations, to capture richer data is present in 18 papers. Of these papers, five used the undirected network or the one with direction reversed in tandem with the original network. It allows for information propagate from, e.g., the transaction's receiver back to the sender.

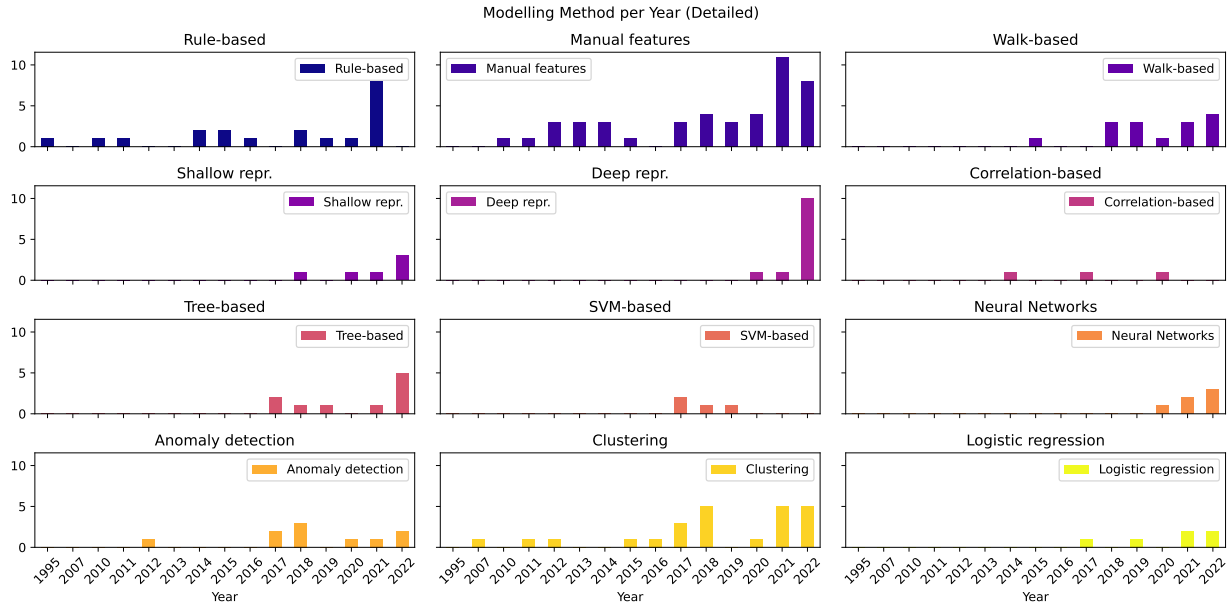


Figure 3: The evolution of the model building blocks over the years. Since there are many modelling methods, the graphs are split.

3.2.2 Objectives.

Table 2 shows that the AML method's objective is most often detecting suspicious clients, followed by detecting suspicious transactions, money flows and communities. Client classification combines client characteristics with payments made over a longer period. Since money laundering is done over a longer period, with the aim to make the individual transaction appear normal, models trained on a series of transactions, leveraging an individual's behaviour, can obtain higher performance.

Figure A3 shows that methods for crypto-currency are most often introduced for suspicious transaction detection. Due to the pseudo-anonymity, it is much harder to know which wallets belong to what person [152, 145], making it less feasible to build profiles. Therefore, researchers in AML have also looked at de-anonymisation of the wallets [152, 145, 48]. They use network features to cluster wallets together to see if they belong to the same person or organization.

In recent years, more studies have been done on flow/chain detection, tracking payments over multiple steps/people. Although this requires more computing power, it more closely resembles reality [126, 75]. Hence, these methods are becoming more widely used.

3.2.3 Evaluation.

Table 2 shows that research mostly relies on the recall, $F1$-score, precision and time. Figure 4 illustrates that training/prediction time was a popular method in earlier studies, but relatively few papers use time in more recent work, which might be a consequence of increased computing power. Additionally, the popularity of the precision, recall and $F1$-score is a recent and increasing trend in the literature.

We remark two surprising results. First, only one paper used the area under the precision-recall curve (AUC-PR). Due to the high label imbalanced, this metric is shown to be more appropriate than the area under the ROC curve (AUC-ROC) [26, 102]. Therefore, we would expect it to be more widely used. Second, some papers solely reported accuracy [31, 141, 48, 143]. Given the high class imbalance, accuracy fails in providing an appropriate performance assessment.

3.2.4 Data.

Table 2 shows that most studies use open-source data, with proprietary data used in almost 50% of the papers. Only a minority of papers report their findings using synthetic data. There is, however, a major difference between crypto- and other research. Figure A1 shows that crypto-research is almost exclusively done on open-source data. This results from

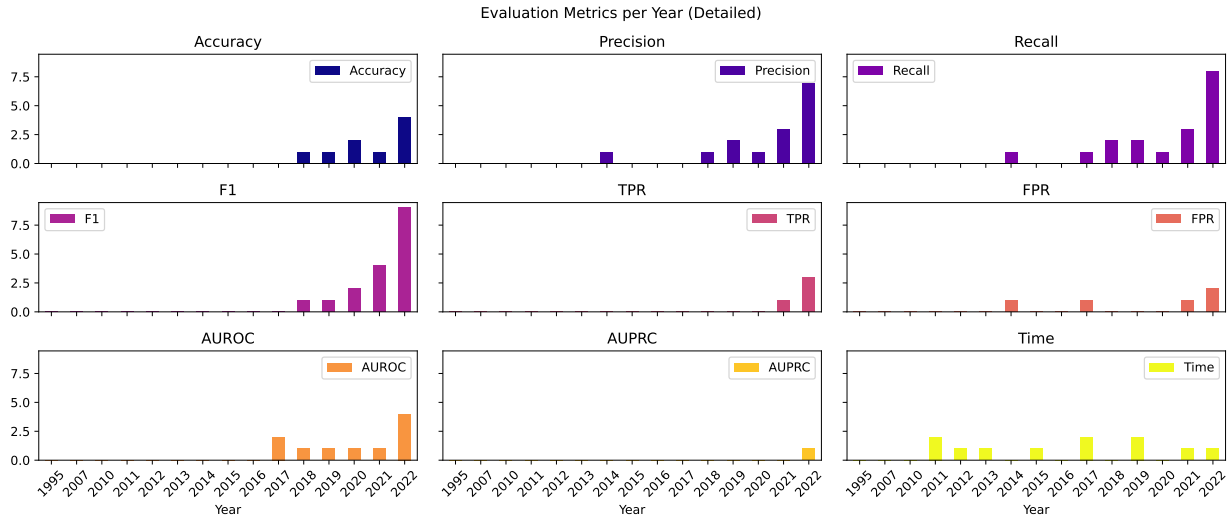


Figure 4: The evolution of the evaluation metrics over the years. Since there are many metrics, the graphs are split.

the blockchain's freely accessible distributed ledger [134]. On the other hand, *classic* transaction data cannot be shared, due to privacy reasons, resulting in half the data sets being proprietary.

Following this last remark, one would expect to have (almost) no research done on open-source data sets for fiat currencies. Figure A1 shows that almost 30% of non-crypto related papers use open-source data. These papers can be categorised in four groups:

- Methods that are evaluated on non-financial network data, whether or not next to proprietary/synthetic money laundering data: Zhou et al. [151], Ovelgönne et al. [101], Micale et al. [92], Prado-Romero and Gago-Alonso [108]
- Methods evaluated on government data, assumed or stated to be open-source: Didimo et al. [29], Imanpour et al. [62], Velasco et al. [131], Bahulkar et al. [11], Malm and Bichler [87]
- Methods constructed to analyse leaked documents/scandals, e.g., the Panama papers: Joaristi et al. [68, 69], Magalingam et al. [84], Adriaens et al. [1], Winiecki et al. [136], Helmy et al. [59], Cheong and Si [23]
- Other: Nandhini and Das [97], Bhalerao et al. [13], Çağlayan and Bahtiyar [17]

For the *Other* category, we have the following. Nandhini and Das [97] say that the data is publicly available, without giving any more information. Bhalerao et al. [13] scraped their data from Hack Forums. Finally, Çağlayan and Bahtiyar [17] use a Kaggle data set based on synthetic data. We categorise it as open-source, since it is made publicly available.

Table 5 summarises the open-source data sets. The most widely used are almost all crypto-currency data sets. Raw Bitcoin data is used by ten papers. This raw data is often enhanced using data from other platforms. One such platform is WalletExplorer, which is used by three papers. It is less relevant today, since the data has not been maintained since 2016. Another recent data set on Bitcoin is the Elliptic data sets [37, 134]. Although only published in 2019, it has already been used in six experiments. After Bitcoin, Ethereum seems to be the second most popular crytpo-currency in AML.

The most used data set for fiat currency is the Czech Financial Data set (CFD). This data set does not provide any labels. Papers using it introduce additional, synthetic money laundering patterns to test their methods [78, 133]. This is not ideal, as this hinders comparison across research and can introduce detection bias.

As mentioned above, AML methods are also being evaluated on leaked data like the ICIJ Offshore Leaks Database (including, e.g., the Panama-papers) and the Enron data set.

3.3 Top Cited Papers

A final step in our discussion is an in-depth analysis of the top-cited papers. These papers are well received by the scientific community and are likely of high quality. We select the 10% most cited papers per year, extended with the 10% most cited papers overall (see Section 2.1). This results in 20 papers as listed in Table 6. The analysis is done in

Table 5: The open-source data sets used in the different papers.

Data Set	Number	References
Raw bitcon	10	[88], [137], [105], [138], [125], [110], [145], [121], [109], [146]
Elliptic Data set	6	[5], [140], [95], [119], [76], [79]
Ethereum XBlock.pro	4	[121], [152], [65], [66]
ICIJ Offshore Leaks Database	4	[69], [44], [68], [136]
WalletExplorer	3	[137], [125], [145]
Czech Financial Data set (CFD)	2	[78], [133]
Enron Data set	2	[84], [1]
RenRenDai	1	[141]
Bloxy	1	[66]
iCOV	1	[47]
Secret Presidential Funds scandal	1	[23]
Tencent QQ	1	[154]
Caviar data set	1	[11]
BitcoinTalk	1	[138]
Ripple Network	1	[19]
2015 Ashley Madison extortion scam	1	[105]
FinCEN	1	[29]
BlockCypher	1	[109]

three stages to gain deeper insights. First, we revisit the categories as presented in Table 1. Second, we analyse the task-level objectives by discussing the data and methods present in these literature. Third, the papers are classified according to the processing steps and challenges, as introduced by Bockel-Rickermann et al. [14].

3.3.1 General Classification.

Five out of 11 review papers are selected. This is to be expected, since review papers have a higher likelihood of being cited [32]. Plots summarising the classification according to Table 1 are given in Appendix B. Similar results are observed for the top-cited papers as for the literature as a whole; unsupervised learning is by far the most applied method, and research relies heavily on manual and rule-based features. Only time and threshold-dependent metrics were used for evaluation. This indicates that threshold-independent metrics are not mainstream when comparing AML models using networks analytics. We observe that, compared to the wider literature, flow/chain detection methods are better cited as well as studies using open-source data. As mentioned before, flow detection better captures the intricate patterns used in money laundering. The use of open-source data makes comparison with and replication of the studies possible, fostering research adoption.

Additionally, five papers deal with crypto-related money laundering, indicating a strong research interest. On the one hand, crypto currencies are widely used by criminals because of the absence of regulation [40] and the anonymity it provides [63, 134]. On the other hand, a full history of ownership is available on the blockchain for each *coin*, facilitating the discovery of transaction patterns in payment networks [63, 134].

3.3.2 Task-Level Objectives.

Table 7 summarises the data set used with information therein, the model proposed by the authors to capture a certain money laundering phenomenon, and the baselines to which the proposed model is compared. Additionally, Table 8 summarises the network properties of these papers.

We again see a strong reliance on *basic* centrality measures for anti-money laundering. A couple of papers apply very basic analysis. Fronzetti Colladon and Remondi [41] and Malm and Bichler [87] purely consider correlations to extract conclusions for their AML features.

Next to the typical objectives of money laundering detection, some research illustrates how network analytics can be used to support investigators to find novel leads and other people involved in previously-reported cases.

Table 7 points to limitations in the literature. Many of the proposed methods are only tested on proprietary data, limiting the reproducibility of their results. The performance is often illustrated using case studies, without the inclusion of other baselines. This shows that there is a lack of a consensus on baselines and state-of-the-art methods in the field.

Table 6: Summary of the top-cited papers, with indication of whether it is a review and/or crypto-related.

Reference	Title	Year	Citations	Review	Crypto
Ngai et al. [99]	The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature	2011	1307	✓	
Fronzetti Colladon and Remondi [41]	Using social network analysis to prevent money laundering	2017	200		
Chang et al. [22]	WireVis: Visualization of Categorical, Time-Varying Data From Financial Transactions	2007	171		
Senator et al. [113]	Financial Crimes Enforcement Network AI System (FAIS) Identifying Potential Money Laundering from Reports of Large Cash Transactions	1995	152		
Drężewski et al. [33]	The application of social network analysis algorithms in a system supporting money laundering detection	2015	141		
McGinn et al. [88]	Visualizing Dynamic Bitcoin Transaction Patterns	2016	122		✓
Gao and Ye [42]	A framework for data mining-based anti-money laundering research	2007	106	✓	
Wu et al. [137]	Detecting Mixing Services via Mining Bitcoin Transaction Network With Hybrid Motifs	2021	94		✓
Malm and Bichler [87]	Using friends for money: the positional importance of money-launderers in organized crime	2013	66		
Zhou et al. [151]	A Local Algorithm for Structure-Preserving Graph Cut	2017	63		
Li et al. [78]	FlowScope: Spotting Money Laundering Based on Graphs	2020	54		
Irwin and Turner [63]	Illicit Bitcoin transactions: challenges in getting to the who, what, when and where	2018	47	✓	✓
Phetsouvanh et al. [105]	EGRET: Extortion Graph Exploration Techniques in the Bitcoin Network	2018	38		✓
Kute et al. [74]	Deep Learning and Explainable Artificial Intelligence Techniques Applied for Detecting Money Laundering–A Critical Review	2021	38	✓	
Zhdanova et al. [150]	No Smurfs: Revealing Fraud Chains in Mobile Money Transfers	2014	32		
Bhalerao et al. [13]	Mapping the underground: Supervised discovery of cybercrime supply chains	2019	24		
Ovelgönne et al. [101]	Covertness Centrality in Networks	2012	20		
Cheong and Si [23]	Event-based approach to money laundering data analysis and visualization	2010	12		
Gerbrands et al. [47]	The effect of anti-money laundering policies: an empirical network analysis	2022	11	✓	
Zhou et al. [152]	Behavior-Aware Account De-Anonymization on Ethereum Interaction Graph	2022	7		✓

The above observations motivate the construction of the experimental set-up of Section 4. Our aim is to come to a common framework to test methods for network analytics in a uniform way.

In addition to the objectives of the studies, Table 8 presents the network construction in these papers. It indicates that most of the papers use homogeneous networks with transactions forming the edges between accounts. Only a few use heterogeneous networks, either by having both accounts and transactions as nodes, or having nodes that represent different characteristics or feature values.

3.3.3 Processing steps.

Next, the papers are classified according to the framework presented by Bockel-Rickermann et al. [14]. Table 9 summarises the papers over the processing steps. The most prevalent pre-processing steps are feature engineering and exploration. Feature engineering is done to define different risk profiles or to enhance GNN models. This is either based on neighbourhood and centrality metrics [41, 33, 101] or on the aggregation of local (node-specific) information [152, 150]. For exploration, some research is done on how to construct meaningful visualisations [22, 88], while other papers rely on clustering the nodes as additional support for their findings [87, 41].

Table 7: Summary of the objective of the paper, where the paper uses [Data set] with [information] to introduce a [model] capturing [phenomenon] and is compared to [baselines].

Paper	Data set	Information	Model	Phenomenon	Baselines
Fronzetti Colladon and Remondi [41]	Italian factoring data	Transaction invoices, economic sector, geographical area	Correlation of centrality metrics with label	Smurfing	-
Chang et al. [22]	Proprietary bank data set	Transactions, accounts, keywords, personal information	Visualisation of keywords-accounts, account using transactions, and evolution transactions of (groups of) accounts	Suspicious behaviour	-
Senator et al. [113]	Reports of large cash transactions according to Bank Secrecy Act	People, accounts, businesses, locations and transactions	Graphical user interface	identifying high-value leads	-
Drężewski et al. [33]	Proprietary bank statements, national court register	Connections	Rule-based role definition via centrality metrics	Roles in criminal organisations	-
McGinn et al. [88]	Bitcoin	Transactions, wallets	Visualisation	Unexpected high-frequency transaction patterns	-
Wu et al. [137]	Bitcoin	Transactions and wallets	Account, transaction and network feature extraction + z-score based on hybrid motives + logistic regression	Mixing	-
Malm and Bichler [87]	Police intelligence reports	Person under investigation in the report and relations, demographic characteristics of each person, nature of illicit drug trade involvement, and the types of relationships that existed among individuals	Correlation of betweenness and eigen-vector centrality with known roles	Role of money-launderers in illicit markets	-
Zhou et al. [151]	Proprietary bank data set	Bank accounts, names, emails, addresses, and phone number	HOSPLOC (local graph clustering)	Synthetic identities and money laundering	Local clustering (Nibble, NPR, LS-OSC), global (NMF, TSC)
Li et al. [78]	Proprietary and Czech Financial Data set	Bank accounts, transactions	FlowScope (objective maximisation to find fast money flow through bank in multi-partite network)	Anomalous money flow indicative for money laundering	SpokEn, D-Cube, Fraudar, HoloScope, RRCF
Phetsouvanh et al. [105]	Bitcoin blockchain and Ashley Madison blackmail	Transactions and wallets	Average length and confluence analysis	Suspicious bitcoin flow and discover other wallets owned by suspected perpetrators	-
Zhdanova et al. [150]	Synthetic MMT data	Transaction between phones	PSA@R (event-driven process analysis)	Smurfing	PART, C4.5, Random Forest
Bhalerao et al. [13]	Forum threads (English and Russian)	Users, Posts and replies	Automatic classification using XGBoost, LogReg, SVM ans FastText. Construction supply chain based on interaction graph. Visualisation via alluvial plots	Money laundering and other fraudulent behaviour	-
Ovelgönne et al. [101]	Youtube and Universitat Rovira I Virgili data sets	Network	Covertness centrality	Hiding behaviour	-
Cheong and Si [23]	Criminal relations database	Transactions and relationships	Rule-based degree of suspicion and visualisation of relations	Find additional suspects in money laundering cases	-
Gerbrands et al. [47]	iCOV	Transactions, family and professional relations of people investigated	Clustering (Louvain and temporal) + node level centralities	Interactions over time	-
Zhou et al. [152]	Ethereum (Eth-ICO, Eth-Mining, Eth-Exchange, Eth-Pish&Hack)	Transactions and wallets	Pipeline based on sampling, Hierarchical Graph ATtention Encoder and sub-graph contrastive learning	Account identification (phishing, exchange etc.)	Manual, Deepwalk, node2vec, trans2vec, graph2vec + LR, RF, LGBM, and GCN, GAT, GIN, I2BGNN-A, I2BGNN-T

Most papers apply unsupervised learning in their processing step. This involves detection of suspicious transaction flows [151, 78, 105]. Another important part is the use of centrality metrics to define specific roles for entities in the network [33, 87, 101]. Finally, some methods create heuristics to define outlying behaviour [113, 22, 23, 150].

For the post-processing step, most research covers statistical evaluation, implementation and interpretation. Statistical evaluation is used to obtain a quantitative evaluation of the performance. Implementation consists of analysing the efficiency of the methods. It deals solely with computation time and scalability. For interpretation, researchers use specific interpretable features.

More detailed trends in the literature are discussed below according to three main categories, that mainly follow the processing steps; unsupervised learning, (semi-)supervised learning and visualisation methods.

Unsupervised learning is divided into two main streams. The first assigns and identifies specific roles of actors in criminal networks, based on police reports of criminal cases [87, 33]. The second covers methods to track money flows through the network [151, 78, 105]. It is important to integrate these transaction chains, since money laundering is a process involving multiple steps and actors [75]. Next to these two main streams, the remaining paper [101] introduces a new covertness measure.

Table 8: Summary of the construction of the network.

Paper	Nodes	Edges
Fronzetti Colladon and Remondi [41]	Factoring companies	Economic sector geographical data transaction amount tacit link (same owners/resources)
Chang et al. [22]	People	Transactions
Senator et al. [113]	Accounts People, businesses, accounts and location	Connections
Drężewski et al. [33]	People (via court register)	Transactions (via bank statements)
McGinn et al. [88]	Transactions, inputs and outputs	Input/output part of same transaction, inputs that belong to the same address
Wu et al. [137]	Address Address and transactions	Transactions If they are related
Malm and Bichler [87]	People (from police reports)	Co-appearance in the report (family, client-lawyer etc.)
Zhou et al. [151]	All items of interest (bank account, name, email, address, phone number)	Link bank account to other items
Li et al. [78]	Bank accounts	Transfers
Phetsouvanh et al. [105]	Bitcoin wallet address	Transactions
Ovelgönne et al. [101]	-	-
Cheong and Si [23]	People	Relations
Zhou et al. [152]	Accounts	Transactions
Zhdanova et al. [150]	mWallets	Transactions

Table 9: Papers per method cluster, based on Bockel-Rickermann et al. [14].

Process Step	Sub-Steps	References
Pre-processing	Sampling	[41].
	Exploration	[41],[22],[88], [87].
	Missing Value Treatment	[41].
	Outlier Detection and Treatment	-
	Categorization, standardization & segmentation	[33], [13].
	Feature Engineering	[41], [33], [101],[152], [150].
	Variable Selection	-
Processing	Unsupervised Learning	[22], [113], [33], [87], [151], [78], [105], [101], [23],[150].
	Supervised Learning	[41], [13].
	Semi-Supervised Learning	[137].
	Hybrid Learning	[152].
Post-processing	Statistical Evaluation	[41], [137], [151], [78], [13], [152], [150].
	Interpretation	[33], [137], [13], [152].
	Economical Evaluation	[150].
	Implementation	[33], [88], [151], [78], [101],[152], [150].

Although detecting money laundering is in principle trying to find anomalous behaviour, very few studies deal with (unsupervised) anomaly detection [85, 108, 34], leaving possibilities for future work.

(Semi-)Supervised learning methods deal with different aspects of money laundering. Their strength lies in a combination of (1) the methods used and (2) tailoring the network to the problem at hand. The obtained network features are put into downstream classifier. To maintain interpretability, network metrics are recalculated for multiple networks, each representing a specific aspect of the data, to extract predictive features [41, 137]. Promising methods are based on deep learning [152], but these are mostly black boxes. Research on the interpretability of deep network representation methods is still scarce [74].

Table 10: Papers per fraud detection challenge, based on Bockel-Rickermann et al. [14].

	Challenge	Papers
Original	Automation	[113], [23], [152].
	Class Imbalance	[137], [13].
	Concept Drift	[41], [13].
	Data Availability	[41], [22], [137], [87], [101], [150].
	Feature Construction	[101].
	Noisy Data	[113], [137], [87].
	Stream Data	[113], [88], [152].
	Real-Time Execution	[41].
	Scalability	[151], [101], [152].
	Unlabelled Data	[113], [137], [13], [152].
	Verification Latency	[150].
New	Bias in data	[87], [150].
	Generalisation	[152].
	Robustness	[78].
	Anonymity	[88], [137], [13].

Visualisation of transaction networks is developed to support the experts in during investigations. This is either done by having a overview of the flow of money over different transactions [88, 22] or by finding specific relations among the persons involved [113, 29, 23]. Hence, these methods are mostly intended to be used after suspicion has been raised. None of them include an evaluation of the added value in money laundering detection.

3.3.4 Challenges.

Table 10 summarises the papers according to the challenges identified by Bockel-Rickermann et al. [14]. It only contains those challenges mentioned in the literature. The main challenge is data availability, followed by unlabelled data. Due to privacy issues, only a limited number of anonymised data is available. Additionally, since transaction data sets are often huge and resources are limited, it is hard to provide all instances with a label.

Few papers mention feature construction, real-time execution, or verification latency as a challenge. Anti-money laundering is less reliant on execution time, since the investigation is done after the fact, based on a longer history of transactions [75].

We introduce four additional challenges observed in the anti-money laundering literature, supplementing Bockel-Rickermann et al. [14]. We view this as an extension of the framework since these challenges can be important to fraud research in general.

- **Bias in data** affects the performance of the model and manifests itself in different ways. First, there is *bias in the missing values* [87]. Second, *detection bias* [150] can be present in synthetic data when specific patterns are present that the model is trained to look for.
- **Generalisation** is mainly used in two ways. One way refers to the ability for methods developed for anti-money laundering to be used for other fraud domains as well [152]. Another and important use of the term refers to the need for methods to generalise to unseen data [49], often referred to as *inductive* methods (compared to *transductive* ones). This is an important consideration when selecting network-based methods [56, 127, 128].
- **Robustness** of a method is important since money laundering can be seen as an adversarial attack [78], as perpetrators try to avoid detection by the methods.
- **Anonymity** of the account data gives rise to two main challenges. First, the identity of the account holder is unknown in crypto data sets, which makes it almost impossible to know if different accounts - also called wallets - belong to the same person [137]. Second, financial institution possess customer data, but due to privacy reasons, researchers cannot publish these. Hence, published articles avoid using Personally Identifiable Information all together [13].

3.4 Further Considerations

Based on the observations made above, we give an intermediate conclusion and perspective on future work. The methods in the literature still rely heavily on expert-based methods, i.e., manual feature engineering and rule-based

systems, but deep learning methods are gaining traction. A more detailed analysis of these methods is needed to evaluate their performance properly, especially given that the current literature barely applies threshold-independent metrics meant for highly skewed data, like the AUC-PR. Moreover, the top-cited papers barely compare their methods to other baselines. This detailed analysis is the topic for the second part of this paper.

Different gaps in the literature exist. A first gap was also identified by Kute et al. [74], namely the lack of interpretation tools. Future work should interpretation methods to try and explain the output of graph neural networks.

A second gap concerns the under-representation of classic unsupervised methods [21, 3]. Future work should analyse the effectiveness of unsupervised learning in networks to find anomalous behaviour. This can be beneficial since many clients and transactions are effectively unlabelled, and the modi operandi for money laundering keep evolving. The second part of this paper will introduce a first way of applying such unsupervised learning methods, but a more extensive study is needed.

A third is that most research studies static networks. Some research is being done to incorporate the time-dynamic nature of transaction networks using recurrent neural networks, but we believe that much more can still be done.

A fourth gap is related to the inherent incompleteness of labels in AML. As criminals try to cover their tracks and financial institutions have only limited resources [129, 106], it is a given that there are still many money laundering cases that stay unnoticed. These transactions are seen as having label 0 when training the models, resulting in a positive and unlabelled (PU) problem. Specific PU-learning methods for networks should be applied when training a model.

A fifth gap is related to the limited resources as well. Since not all clients and transactions can be investigated, the predictions should take this into account. One way of dealing with this is learning-to-rank [28, 130], which has already been shown to be effective in fraud detection. The main benefit is that the goals is to correctly rank the most suspicious clients, while less importance is given to have a correct ranking for less suspicious clients. Although this is highly relevant, it has not yet been applied to the AML literature using network analytics.

4 Experimental Set-Up

To the best of our knowledge, a benchmark study comparing the state-of-the-art network learning methods for anti-money laundering is still missing. This section describes the experimental framework to fill this gap, including model specification (Section 4.1), data (Section 4.2), hyperparameter tuning (Section 4.3) and performance metrics (Section 4.4). The focus is on supervised learning, but unsupervised learning is also explored. Given challenges in open-source data availability, we present results on two data sets and provide code to facilitate reproducible experiments. We hope this framework will support and expand future research, and allows uniform publishing and comparing of results on proprietary data without the need to disclose sensitive information.

4.1 Model Specification

To set up the experimental evaluation, we start from the wider literature on network representation. Extensive overviews are given by Cai et al. [18], Hamilton et al. [56] and Goyal and Ferrara [50]. A sub-branch of representation learning applies deep learning methods [153, 139]. Based on these papers, we select the methods that are most prominently used.

We refined the selection by considering that, for AML, transaction data contains networks with millions of nodes and edges, making scalability an important selection criterion. Table 11 provides an overview of the methods selected. They are grouped into three categories [18, 56, 50]: manual feature engineering, shallow representation learning and deep representation learning. These methods will be tested against a baseline model that only includes the intrinsic features, denoted by IF.

Manual feature engineering includes both local and global metrics. The first is the density of the ego-network. A node's ego-network consists of the node, its direct neighbours and all connections between the neighbours. The density is the relative number of connections in this ego-network compared to the theoretical maximum. Summary statistics, i.e., minimum, mean and maximum values, of the density of the node's neighbours are also included.

A second type consists of centrality metrics, quantifying the global position of the node in the network [98]. The most popular centrality metrics are the betweenness, closeness and eigenvector centrality. The betweenness quantifies the number of shortest paths between any two nodes in the network passing through that node. The closeness, being the inverse of the average distance, measures how close a node is to all other nodes. The eigenvector centrality is high for nodes that are connected to important nodes in the network. The importance of the node itself is quantified using the PageRank [103].

Table 11: The methods used to generate network features for the experimental evaluation.

Category	Features	Definition
Manual feature engineering	Density	Baesens et al. [10]: Number of edges in the egonet relative to the maximal number possible.
	Centrality measures	Closeness, betweenness and eigenvector centrality.
	PageRank	Page et al. [103]: Score indicating the importance of a node.
Shallow representation learning	DeepWalk	Perozzi et al. [104]: Random walk through the network. Embedding via word2vec.
	Node2vec	Grover and Leskovec [52]: Truncated random walks (breadth-first vs. depth-first). Embedding via word2vec.
Deep representation learning	GCN	Kipf and Welling [70]: Graph convolutional network
	GraphSAGE	Hamilton et al. [56]: Aggregation based on fixed-size sample of neighbours
	GAT	Veličković et al. [132]: Graph attention network
	GIN	Xu et al. [144]: Graph isomorphism network

Next to the manual feature engineering, automatic network features are constructed using network embedding methods [56, 50, 18]. Although there are many different methods to construct a network embedding [18], we use those based on deep learning, called network representation learning [127, 129].

Shallow network representation learning can be viewed as an "embedding lookup" [56, 129]. We apply methods based on random walks. DeepWalk [104] was the first method to try to give nodes that are close in the network, i.e., that co-occur on short random walks, a similar embedding. Each random walk is seen as a sentence of words. The nodes are embedded into a Euclidean latent space using NLP methods, often skip-gram [94].

Node2vec [52] extends DeepWalk by introducing two hyperparameters, p and q. The method samples each node according to the following unnormalized transition probability, α, at node v, when coming from node t:

$$\alpha(c_{i+1} = x \mid c_i = v, c_{i-1} = t) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases},$$

where d_{tx} denotes the distance between node t and x. These probabilities are defined a link exists between v and x. Otherwise, it is set to 0.

The main drawbacks of shallow representation learning methods are that (1) they are transductive, so need full retraining when a new node is added to the network, and (2) retraining on the same network results in a different embedding. This second point can be understood intuitively by viewing two embeddings of the same network, with one being a rotation of the other. Both are equally good, but a downstream classifier trained on one cannot be used to make predictions on the other, as the coordinates of the nodes are different.

In the experiments below, we construct the embedding on all nodes in the training and validation set, while only training on the labels from the training set. Afterwards, the embedding is constructed for the full network, and we train a new classifier using the labels of the train and validation set.

This mimics the applications of these methods in reality, since a bank will have the full transaction network up to the current time, with some historical labels. Based on this, they need to determine which accounts to investigate next for money laundering.

Deep network representation learning adapts deep learning methods to be directly applicable on networks. We apply different graph neural network (GNN) architectures.

The most popular methods for node embedding calculation are graph convolutional networks (GCNs), based on convolutional neural networks [70]. It relies on neighbourhood aggregation to update the embedding as follows:

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

with $H^{(l)}$ the embedding at step l, σ a non-linear (activation) function, $\tilde{A} = A + I$ the adjacency matrix with self-loops added, the diagonal matrix $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $W^{(l)}$ the learnable weights in layer l. The node embedding for the

individual node is updated via

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \tilde{\mathcal{N}}_i} \frac{1}{\sqrt{\deg(i)\deg(j)}} W^{(l)} h_j^{(l)}\right).$$

Neighbourhood aggregation has seen multiple extensions, two of which this paper will evaluate. The first is GraphSAGE [56], which samples a fixed number of neighbours for each node, to make the calculations scalable to large graphs. The representations of the sampled neighbours are aggregated, with the authors proposing the mean, LSTM and pooling aggregators [56]. Then, the aggregated neighbourhood representation is concatenated with the node's representation. Finally, the representation is updated using the weights and activation function. For a given node v, this is calculated as:

$$\begin{aligned} h_{\mathcal{N}_v}^{(l)} &= \texttt{AGGREGATE}_l\left(\{h_u^{(l-1)}, \forall u \in \mathcal{N}_v\}\right) \\ h_v^{(l)} &= \sigma\left(W^{(l)} \cdot \left[h_v^{(l-1)} \parallel h_{\mathcal{N}_v}^{(l)}\right]\right), \end{aligned}$$

where $\parallel$ represents the concatenation operator.

The second extension is graph attention network (GAT) [132], which adds an attention mechanism. Our work uses the modified version of the attention mechanism, introduced to fix the static attention problem [16]:

$$\begin{aligned} e_{ij} &= a^T \texttt{LeakyReLU}\left(W^{(l)}\left[h_v^{(l-1)} \parallel h_{\mathcal{N}_v}^{(l)}\right]\right) \\ \alpha_{ij} &= \texttt{softmax}_j(e_{ij}) \\ h_i^{(l)} &= \sigma\left(\sum_{j \in \tilde{\mathcal{N}}_i} \alpha_{ij} W^{(l)} h_j^{(l-1)}.\right) \end{aligned}$$

As mentioned by Veličković et al. [132], having multiple mechanisms—also called heads—can improve stability. We follow Veličković et al. [132], where the intermediate steps concatenate the results over the heads, while the final layer averages the results.

The final method is the Graph Isomorphism Network (GIN) [144]. The authors used the Weisfeiler-Lehman graph isomorphism test to define the necessary conditions a GNN must satisfy to achieve maximal discriminative power. This discriminative power is denoted by how well the training data is fitted.

Using the universal approximation theorem [61, 60], GIN updates the representation using a multilayer perceptron (MLP).

$$h_v^{(l)} = \texttt{MLP}^{(l)}\left(\left(1 + \epsilon^{(l)}\right) h_v^{(l-1)} + \sum_{j \in \mathcal{N}_i} h_j^{(l-1)}\right)$$

Although this increases the risk of overfitting, the authors show that GIN has satisfactory generalisation capabilities. The main problem with other GNNs is that they tend to underfit the data [144]. Recent research showed GIN's potential to outperform other GNN architectures [36, 123].

There are some key differences between the three categories presented in Table 11. The manual features need to be specifically defined, while representation learning tries to find meaningful embeddings automatically. Additionally, the manual features and the shallow representations are added to the intrinsic features, which are used in a down-stream classifier (as illustrated in Figure 5). The GNNs incorporate the intrinsic features and directly learn the embeddings on the classification task.

Different decoders are used for the final classification in the supervised setting. A neural network with two hidden layers, each of dimension ten is used for the manual features and shallow representation methods. For the graph neural networks, the GNN layers are followed by a single linear layer for making predictions.

For the unsupervised model, we apply isolation forest [80], as it achieves state-of-the-art performance for outlier/anomaly detection [122, 64]. This will be applied to the intrinsic and manual features, and the shallow representations, since it requires a tabular input. A larger experiment comparing unsupervised methods for network analytics is left for future work.

One of the objectives of this research is to provide the code for all methods in a clean and simple way and make these methods easily accessible to researchers and practitioners. The methods are implemented in Python. The manual

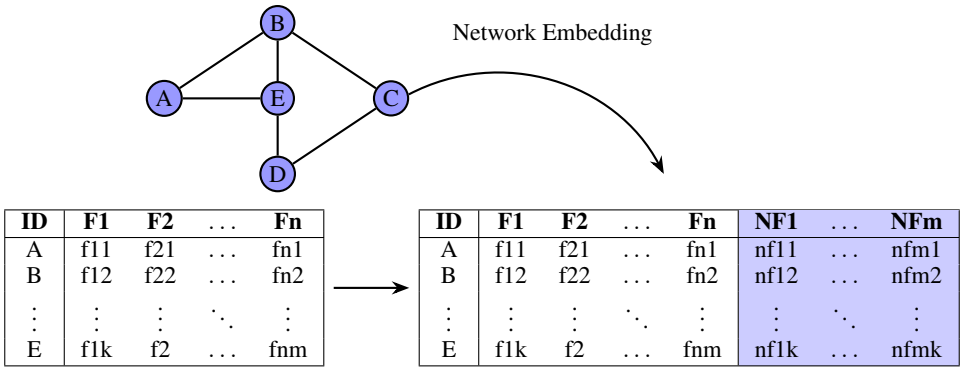


Figure 5: For the feature engineering and shallow representation methods, the obtained network features/embedding is added to the intrinsic ones.

features are constructed using NetworkX [54] and NetworKit [7], and the representation learning is implemented in PyTorch Geometric [39]. The code is made available on GitHub[2].

4.2 Data

The methods are compared on two data sets; the Elliptic [37, 134] and IBM money laundering [6, 36] data sets. These cover the two main lines of research in the literature. The Elliptic data set concerns real-world crypto-transactions, while the IBM data set deals with simulated transactions in fiat currencies.

4.2.1 Elliptic.

The Elliptic data set [37, 134] contains Bitcoin transactions for 49 time steps, each being around two weeks long. The network has 203 769 nodes and 234 355 edges. The nodes correspond to transactions, and an edge represents that the output of one transaction is the input of the next. Additionally, the data set contains 166 pre-calculated features. These are split into 94 transaction-specific, i.e., local features and 72 aggregated features, summarizing the local features of a node's neighbours.

All features are numerical and have already been standardised. The local features will be used for all methods. Aggregated features are not included for the GNNs, since these methods aggregate the neighbourhood information, including the node features. This was also addressed by Weber et al. [134], stating that graph neural networks are better to address the heterogeneity of the neighbourhoods than the aggregated features.

Around 33% of transactions are labelled, 4 545 of which as illicit and 42 019 as licit. Hence, the classification problem is imbalanced with 2% of all nodes labelled *illicit*. We note that an illicit label does not automatically mean that the transaction is used to launder money. However, criminals will try to obscure the source of these illicit funds, so we believe that it is still informative. Section 3.2 also illustrated that this data set is widely used in AML research.

We use the Elliptic data set included in the Pytorch Geometric library [25, 39]. We split the periods into a train (period 1-30), validation (period 31-40) and test set (period 41-49). Only nodes having a label are used for performance evaluation. Transactions with label 2 (unknown) are included for the construction of the network, but they do not contribute to training or loss calculation [70, 147, 134].

4.2.2 IBM Money Laundering.

The second data set is the HI-Small data set, taken from Kaggle, consisting of simulated transactions in a virtual world between individuals, companies and banks [6, 36]. A fraction of these transactions are labelled as money laundering. Given that the labels are at transaction level, the network is constructed with the transactions as nodes. Edges are added from transaction v_i to v_j if the receiver of transaction v_i is the sender of transaction v_j, transaction v_j happened before v_i, and the difference in time is smaller than Δt, i.e., $\Delta t \geq t_j - t_i \geq 0$. This is inspired by the work by Tariq and Hassani [120], and allows to capture the money flow. We take a 60-20-20 train-validation-test split based on transaction time.

[2] https://github.com/B-Deprez/AML_Network

Table 12: Network features for the Elliptic and IBM-AML data set.

Network	Nodes	Edges	Percentage Fraud	Avg. Degree	Avg. Path Length	Clustering
Elliptic	203,769	234,355	2%	2.3	15.96	0.0117
IBM-AML	500,000	1,278,952	0.27%	5.1	3.78	0.0000

Since this results in a massive data set and hyperparameter tuning for DeepWalk and node2vec would take too long, we only take the last 500 000 transactions, and set Δt to four hours. This results in a network with 500 000 nodes and 1 278 952 edges. Just 1356 transactions, i.e., 0.27%, are labelled as money laundering.

Each transaction has a couple of features, of which we use the following: 'Amount Received', 'Receiving Currency', 'Amount Paid', 'Payment Currency', 'Payment Format'. Next to that, we extract the day of the week, hour and minute out of a transaction's date stamp. The categorical features, i.e., 'Receiving Currency', 'Payment Currency', 'Payment Format', are transformed using one-hot-encoding. In the remainder of this paper, we denote this data set as IBM-AML.

An analysis of the topology of both networks is given in Table 12, based on well-known network features. IBM-AML seems to be much more connected, compared to the Elliptic data set, having higher average degree and lower average path length. This is corroborated by the degree distribution and the distributions of the closeness centrality, in Figure 6 and Figure 7, respectively. The Elliptic data set more closely presents a scale-free distribution, while the IBM-AML data set deviates from this with fewer nodes with low degree and more nodes with higher degree (hubs). The hubs resulted in a much lower average path length, and higher closeness centrality values.

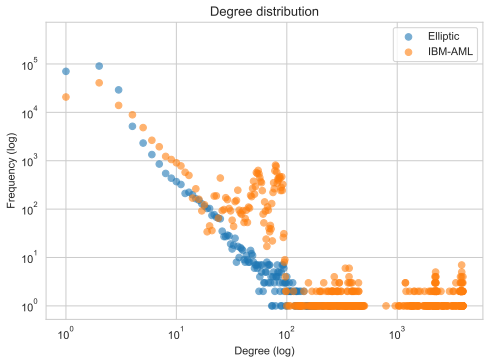


Figure 6: Comparison of the degree distributions of the Elliptic and IBM-AML data sets.

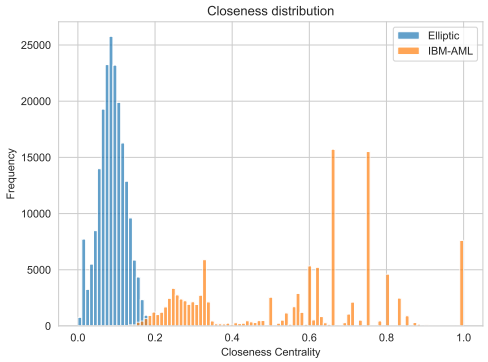


Figure 7: Comparison of the distributions of the closeness centrality of the Elliptic and IBM-AML data sets.

4.3 Hyperparameter Tuning

The hyperparameters are tuned to maximises the AUC-PR on the validation set. However, it is not feasible to tune these using a grid search. As the methods often have many hyperparameters, the number of combinations increases

Table 13: The overview of the hyperparameter values for the supervised learning models. The tuned hyperparameter value is added between parentheses next to the name of the model.

Hyperparameter	**Tuning Range**	**Model (hyperparameter value Elliptic | IBM)**
α: random jump parameter	$[0.1, 0.9]$	PageRank^{S} (0.593 | 0.646)
Number of walks per node	$[1, 3] \cap \mathbb{N}$	DeepWalk^{S} (2 | 2), node2vec^{S} (1 | 3)
Walk length	$[3, 10] \cap \mathbb{N}$	DeepWalk^{S} (3 | 7), node2vec^{S} (9 | 9)
Word2vec context window size	$[2, 10] \cap \mathbb{N}$	DeepWalk^{S} (2 | 2), node2vec^{S} (5 | 9)
Latent/Embedding dimension	$[2, 64] \cap \mathbb{N}$	DeepWalk^{S} (5 | 52), node2vec^{S} (47 | 9)
Latent/Embedding dimension	$[32, 128] \cap \mathbb{N}$	GCN (108 | 97), GraphSAGE (48 | 53), GAT (99 | 51), GIN (74 | 47)
p: return parameter	$[0.5, 2]$	node2vec^{S} (1.17 | 0.537)
q: in-out parameter	$[0.5, 2]$	node2vec^{S} (1.60 | 1.394)
Number of negative samples	$[1, 5] \cap \mathbb{N}$	DeepWalk^{S} (1 | 1), node2vec^{S} (1 | 4)
GNN hidden dimensions	$[64, 256] \cap \mathbb{N}$	GCN (NA | 88), GraphSAGE (NA | 175), GAT (NA | NA), GIN (NA | NA)
GNN layers	$[1, 3] \cap \mathbb{N}$	GCN (1 | 2), GraphSAGE (1 | 2), GAT (1 | 1), GIN (1 | 1)
Learning rate	$[0.01, 0.1]$	IF^{S} (0.0163 | 0.0770), Manual^{S} (0.0166 | 0.0401), DeepWalk^{S} (0.0554 | 0.0526), node2vec^{S} (0.0159 | 0.0708), GCN (0.0225 | 0.0514), GraphSAGE (0.0248 | 0.0660), GAT (0.0421 | 0.0313), GIN (0.0345 | 0.0166)
Aggregator	{min, mean, max}	GraphSAGE (mean | max)
Number of attention heads	$[1, 5] \cap \mathbb{N}$ | $[1, 2] \cap \mathbb{N}$	GAT (4 | 2)
Dropout rate	$[0, 0.5]$	GCN (0.288 | 0.163), GraphSAGE (0.350 | 0.244), GAT (0.186 | 0.188), GIN (0.271 | 0.273)
Number of layers decoder	$[1, 3] \cap \mathbb{N}$	IF^{S} (1 | 2), Manual^{S} (1 | 2)
Hidden dimension decoder	$[5, 20] \cap \mathbb{N}$	IF^{S} (5 | 14), Manual^{S} (6 | 17)
Number of epochs decoder	$[5, 500] \cap \mathbb{N}$	IF^{S} (497 | 450)
Number of epochs decoder	$[5, 100] \cap \mathbb{N}$	Manual^{S} (64 | 59), DeepWalk^{S} (80 | 75), node2vec^{S} (93 | 75)
Number of epochs	$[5, 500] \cap \mathbb{N}$ | $[5, 100] \cap \mathbb{N}$	DeepWalk^{S} (176 | 75), node2vec^{S} (222 | 84), GCN (483 | 325), GraphSAGE (498 | 399), GAT (280 | 341), GIN (218 | 89)

exponentially. Additionally, transaction networks are very large, which results in a relative long training time for each individual combination of hyperparameters. Therefore, we tune the hyperparameters efficiently via Optuna [2], whereby we give a range of possible values to select from. We use optuna's default `TPESampler` sampler.

The ranges are summarised in Table 13 and 14, for supervised and unsupervised learning, respectively. We add superscript S and U to indicate that tuning is done for supervised and unsupervised learning, respectively. A model only based on the intrinsic features (IF) is included. The number of trials is set to 50 for DeepWalk and node2vec, and 100 for the GNNs.

Since the IBM data set is much larger, scalability in both time and memory are very important. Especially DeepWalk and node2vec take much time to run. The higher average degree also makes transition probability calculations more expensive. That is why the number of epochs for calculating the embedding are capped at 100. The attention mechanism of GAT is conditional on the query node, meaning that attention is learned for each node pair. To be able to train the model in memory, we limit the attention head to a maximum of two.

All decoders used for classification have a fixed architecture (Section 4.1). For the GNNs, the training of the decoder weights happens simultaneously with the other weights during training. For the manual and shallow representation features, the decoder/classifier needs to be trained afterwards. The only hyperparameter tuned is the number of epochs.

The tuned hyperparameters are presented in parentheses in Table 13 and Table 14.

Table 14: The overview of the hyperparameter values for the unsupervised learning models. The tuned hyperparameter value is added between parentheses next to the name of the model.

Hyperparameter	**Tuning Range**	**Model (hyperparameter value Elliptic | IBM)**
α: random jump parameter	$[0.1, 0.9]$	PageRank^{U} (0.187 | 0.117)
Number of walks per node	$[1, 3] \cap \mathbb{N}$	DeepWalk^{U} (3 | 2), node2vec^{U} (3 | 2)
Walk length	$[3, 10] \cap \mathbb{N}$	DeepWalk^{U} (6 | 7), node2vec^{U} (4 | 6)
Word2vec context window size	$[2, 10] \cap \mathbb{N}$	DeepWalk^{U} (4 | 6), node2vec^{U} (2 | 4)
Latent/embedding dimension	$[2, 64] \cap \mathbb{N}$	DeepWalk^{U} (64 | 6), node2vec^{U} (63 | 47)
p: return parameter	$[0.5, 2]$	node2vec^{U} (0.78 | 1.52)
q: in-out parameter	$[0.5, 2]$	node2vec^{U} (1.58 | 1.60)
Number of negative samples	$[1, 5] \cap \mathbb{N}$	DeepWalk^{U} (4 | 5), node2vec^{U} (4 | 1)
Learning rate	$[0.01, 0.1]$	DeepWalk^{U} (0.0948 | 0.0541), node2vec^{U} (0.0414 | 0.062)
Number of epochs	$[5, 500] \cap \mathbb{N}$ | $[5, 100] \cap \mathbb{N}$	DeepWalk^{U} (17 | 46), node2vec^{U} (59 | 50)
Number of estimators	$[50, 200] \cap \mathbb{N}$	IF^{U} (60 | 119), Manual^{U} (70 | 66), DeepWalk^{U} (69 | 169), node2vec^{U} (60 | 100)
Max. number of samples	$[0.1, 1]$	IF^{U} (0.484 | 0.111), Manual^{U} (0.333 | 0.559), DeepWalk^{U} (0.533 | 0.386), node2vec^{U} (0.824 | 0.354)
Max. number of features (%)	$\{10 * n \mid n \in \mathbb{N}_{10}\}$	IF^{U} (0.1 | 0.9), Manual^{U} (0.8 | 0.3), DeepWalk^{U} (0.8 | 0.2), node2vec^{U} (0.8 | 0.4)
Bootstrap	{True, False}	IF^{U} (True | False), Manual^{U} (True | True), DeepWalk^{U} (True | False), node2vec^{U} (True | True)

4.4 Performance Metrics

We tackle money laundering as a binary classification problem. We adopt the most popular performance metrics from the literature, namely the precision, recall and F1-score (Section 3.2). These require a classification threshold. Since investigation resources are limited [129], the thresholds will be set relatively high, e.g., by classifying the top 0.1%, 1% or 10% of scores as money laundering. This is supplemented by a threshold equal to the relative prevalence of money laundering in the data sets.

The selection of these thresholds is ad-hoc, and can influence the conclusions. Therefore, we also include threshold-independent metrics. The most popular one for binary classification is the area under the ROC curve (AUC-ROC). The ROC curve plots the true positive rate against the false positive rate. However, in highly imbalanced data sets with many negative samples, the false positive rate can appear low even when the absolute number of false positives is high, making it less sensitive to performance issues in detecting the minority class [26, 111]. The area under the precision-recall curve (AUC-PR) is better suited to compare these models [26, 102, 111]. This also adds to the literature, since we showed in Section 3.2 that few papers base their conclusions on the AUC-PR.

To test model stability, the models are trained 10 times on a randomised version of the training set. The manual network features and shallow representation embeddings result in classic tabular features (cf. Figure 5). For these, we apply a bootstrap method, where a new train set is constructed of the same size as the original, where the observations are sampled with replacement. The deep representation methods apply train masks. We initialise new train masks by randomly selecting half the train masks from the original ones.

5 Results and Discussion

5.1 Supervised Learning

The results for the threshold-independent metrics, i.e., AUC-ROC and AUC-PR, on the test set are shown in Table 15, including the standard deviation calculated as described in Section 4.4. Additionally, the threshold-dependent results, i.e., precision, recall and F1-score, are given in Table 16 and Table 17 for the top-1% results. To analyse the sensitivity to the threshold, we report the results on the other three thresholds in Appendix C.

Table 15: Threshold-independent metrics: AUC-ROC and AUC-PR values over the different methods for the two data sets, based on the test set. The standard deviation is also reported.

Supervised	**Elliptic**		**IBM-AML**	
Methods	**AUC-ROC**	**AUC-PR**	**AUC-ROC**	**AUC-PR**
Intrinsic features	0.8607 ± 0.0079	0.5730 ± 0.0958	0.7560 ± 0.0075	0.0186 ± 0.0008
Egonet features	0.8648 ± 0.0090	0.5956 ± 0.0328	$\mathbf{0.7579 \pm 0.0033}$	$\mathbf{0.0187 \pm 0.0004}$
DeepWalk	0.8483 ± 0.0083	0.5823 ± 0.0265	0.7558 ± 0.0033	0.0185 ± 0.0004
Node2vec	0.8497 ± 0.0044	0.5938 ± 0.0251	0.7551 ± 0.0031	0.0184 ± 0.0004
GCN	0.8465 ± 0.0110	0.5948 ± 0.0164	0.5657 ± 0.0904	0.0113 ± 0.0029
GraphSAGE	$\mathbf{0.8712 \pm 0.0122}$	$\mathbf{0.6392 \pm 0.0336}$	0.6068 ± 0.0793	0.0120 ± 0.0028
GAT	0.8579 ± 0.0161	0.6376 ± 0.0270	0.4203 ± 0.1889	0.0103 ± 0.0041
GIN	0.8213 ± 0.0203	0.5079 ± 0.0593	0.5028 ± 0.2533	0.0119 ± 0.0065

Table 16: Threshold-dependent metrics: Precision, recall and F1-score values for the top 1% scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

	Elliptic Supervised		
Methods	**Precision**	**Recall**	**F1-score**
Intrinsic features	0.8570 ± 0.2706	0.1840 ± 0.1173	0.2932 ± 0.1451
Egonet features	0.9339 ± 0.0463	0.1645 ± 0.0082	0.2797 ± 0.0139
DeepWalk	0.9116 ± 0.0464	$\mathbf{0.1881 \pm 0.0895}$	$\mathbf{0.3049 \pm 0.1054}$
Node2vec	0.9321 ± 0.0465	0.1642 ± 0.0082	0.2791 ± 0.0139
GCN	0.9911 ± 0.0094	0.1728 ± 0.0047	0.2942 ± 0.0070
GraphSAGE	0.9822 ± 0.0188	0.1721 ± 0.0077	0.2929 ± 0.0113
GAT	$\mathbf{0.9964 \pm 0.0075}$	0.1753 ± 0.0067	0.2981 ± 0.0099
GIN	0.9286 ± 0.0709	0.1633 ± 0.0129	0.2777 ± 0.0214

Table 17: Threshold-dependent metrics: Precision, recall and F1-score values for the top 1% scores over the different methods for the IBM-AML data set, based on the test set. The standard deviation is also reported.

	IBM-AML Supervised		
Methods	**Precision**	**Recall**	**F1-score**
Intrinsic features	0.0190 ± 0.0040	0.0210 ± 0.0044	0.0200 ± 0.0042
Egonet features	$\mathbf{0.0192 \pm 0.0034}$	0.0213 ± 0.0038	0.0202 ± 0.0036
DeepWalk	0.0174 ± 0.0053	0.0193 ± 0.0059	0.0183 ± 0.0056
Node2vec	0.0181 ± 0.0047	0.0200 ± 0.0052	0.0190 ± 0.0049
GCN	0.0100 ± 0.0045	0.8807 ± 0.3119	0.0197 ± 0.0089
GraphSAGE	0.0119 ± 0.0027	$\mathbf{0.9742 \pm 0.0298}$	$\mathbf{0.0236 \pm 0.0052}$
GAT	0.0065 ± 0.0070	0.1805 ± 0.3695	0.0104 ± 0.0139
GIN	0.0085 ± 0.0087	0.0166 ± 0.0087	0.0092 ± 0.0089

For both data sets, the models with intrinsic features give already good result. Adding egonet features is beneficial for both data sets. We do, however, see a drop in performance for DeepWalk and node2vec. It is possible that some of the feature values, i.e., the coordinates in the latent space, are correlated or noisy. Neural networks, used here as the decoder for classification, are known to suffer under correlated and noisy features [51]. GIN also performs poorly on both data sets. One explanation is that the neural network in the aggregation step is not optimal, and should be tuned further. Another explanation is that the inclusion of neural network aggregation makes GIN more sensitive to the high class imbalance. Hence, GIN seems less suited for anti-money laundering. It should be noted that good performance is observed for the precision indicating that GIN reduces false positives in the top-percentiles of predictions.

When looking at the Elliptic data set, we see that also the GNNs appear to underperform, except for GraphSAGE. Additionally, GAT has good AUC-PR performance. We see in Tables 16, C1, C3 and C5 that for the top percentages the GNNs have remarkably higher precision, but lower recall. Although fewer cases are detected, those with highest predicted money laundering propensity contain far fewer false positives. This is highly important when adopting these models in practice.

The results on the IBM data set in Table 15 show a different picture. Only the egonet features improve performance. The threshold-dependent metrics show a similar picture.

One explanation for the poor performance of GNNs might be related to the network structure (Section 4.2), with very short average paths and many high-degree nodes. This results in strong over-smoothing [77], resulting in very similar embeddings for the different nodes with insufficient discriminatory power.

Moreover, a deeper analysis reveals that the training loss, when training on the train and validation set, is highly unstable. Depending on the epoch and specific weight initialization, the results vary widely, with the models sometimes just predicting the majority class. For this specific data set, GNNs seem to face instability due to the extreme class imbalance.

The supervised learning experiments on both data sets demonstrate that the network structure provides additional insights for anti-money laundering. The observed performance improvements are limited, but even a small relative increase in performance has material implications for the business. The efficiency of AML practices increases significantly since investigators have more high quality predictions, and spend less time on the investigation of false positives.

However, depending on the network topology and the extent of the class imbalance, GNNs might be too unstable. Additional experiments are needed to quantify the sensitivity of GNNs to the imbalance, but this is outside the scope of this research.

5.2 Unsupervised Learning

The results of the unsupervised methods using isolation forests are given in Tables 18. None of the results on the Elliptic data set are particularly good, with AUC-ROC values below 0.5. For completeness, we also include the threshold-dependent metric in Table 19, where most models fail to detect any money laundering cases.

These results are in line with a previous study performed by Lorenz et al. [83] who tested anomaly detection methods on the Elliptic data set. The authors did not include any additional network analytics methods and only reported the F1 score. However, they found that across the methods, the anomaly detection results were much worse than for supervised learning, with reported F1-scores for the isolation forest equal to 0.

We perform additional analyses on these results. Figure 8 shows the anomaly scores for the intrinsic features for the Elliptic data set. Here, a very negative score signifies anomalies. All money laundering cases have scores near zero, within the bulk of the distribution, indicating that the isolation forest does not recognize these cases as anomalies.

Conversely, the results for IBM-AML are more in line with expectations. Network information captured by DeepWalk, results in the best performing model here. Comparing these results to those in Table 15, we see that DeepWalk even outperforms the supervised learning methods in terms of AUC-PR.

The most plausible explanation in our opinion lies in the assumptions underlying isolation forest and the data generating process. Two specific assumptions by Liu et al. [80]:

- Anomalies yield fewer partitions since anomalies occupy regions with lower density;
- Instances with distinct attributes are separated earlier in the tree, contributing to shorter paths.

Therefore, the nature of the data sets explains the difference in performance. The Elliptic data set is a real-life data set, containing criminals that actively attempt to cover their tracks [10]. They try to mimic the 'average behaviour', but no real transaction/person is average on all attributes. This is why it seems that classifying the least suspicious

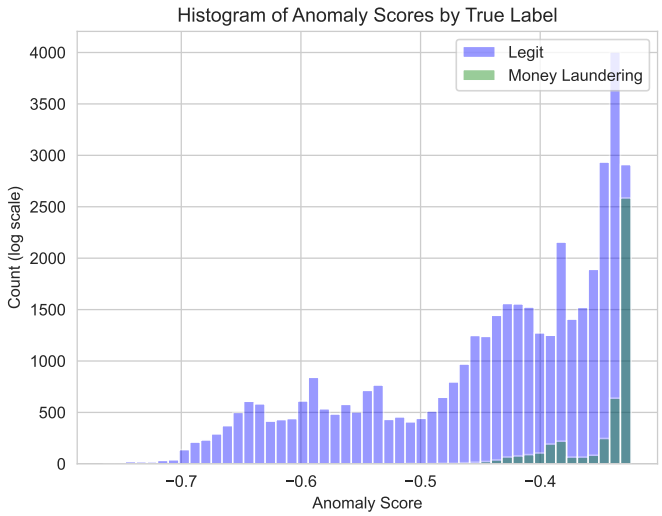


Figure 8: The anomaly scores for the legitimate and money laundering cases. More negative is more anomalous.

Table 18: Threshold-independent metrics: AUC-ROC and AUC-PR values over the different methods for the two data set, based on the test set. The standard deviation is also reported.

Unsupervised Methods	**Elliptic**		**IBM-AML**	
	AUC-ROC	**AUC-PR**	**AUC-ROC**	**AUC-PR**
Intrinsic features	0.134 ± 0.007	0.058 ± 0.001	0.710 ± 0.015	0.017 ± 0.003
Egonet features	0.217 ± 0.010	0.013 ± 0.000	0.700 ± 0.033	0.011 ± 0.005
DeepWalk	0.182 ± 0.008	0.056 ± 0.001	$\mathbf{0.744 \pm 0.023}$	$\mathbf{0.024 \pm 0.008}$
Node2vec	0.190 ± 0.008	0.056 ± 0.0005	0.632 ± 0.031	0.005 ± 0.001

transactions as money launderers, we would achieve an AUC-ROC around 80%. Hence, transactions that appear 'too normal' are actually most suspicious. In contrast, the IBM-AML data set is synthetic, with fraudulent patterns and behaviours deliberately introduced. It is likely that camouflage behaviour is not present in this data set.

These insight highlights the importance of knowing the exact assumptions underlying the data sets used. Synthetic data sets might give overly optimistic results, since complex behaviour inherent to the problem at hand are not captured.

5.3 Ablation Study

To better understand the effect of the network structure on the predictive power of the models, we analyse the performance of the different models without intrinsic features. For the models, we take the hyperparameters as determines in Table 13 and Table 14. The GNNs require feature values, so we give each node a dummy feature with value 1 [128]. Additionally, for the Elliptic data set, we have no control on the summary features, so we discard these as well, although they represent some network structure.

Table 19: Threshold-dependent metrics: Precision, recall and F1-score values for the top 1% scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

Methods	**Elliptic Unsupervised**		
	Precision	**Recall**	**F1-score**
Intrinsic features	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Egonet features	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
DeepWalk	0.005 ± 0.005	0.001 ± 0.001	0.001 ± 0.001
Node2vec	0.003 ± 0.004	0.000 ± 0.000	0.001 ± 0.001

Table 20: Threshold-dependent metrics: Precision, recall and F1-score values for the top 1% scores over the different methods for the IBM-AML data set, based on the test set. The standard deviation is also reported.

Methods	IBM-AML Unupervised Precision	Recall	F1-score
Intrinsic features	0.042 ± 0.005	0.156 ± 0.020	0.067 ± 0.008
Egonet features	0.027 ± 0.013	0.098 ± 0.048	0.042 ± 0.021
DeepWalk	$\mathbf{0.055 \pm 0.009}$	$\mathbf{0.204 \pm 0.034}$	$\mathbf{0.087 \pm 0.015}$
Node2vec	0.007 ± 0.002	0.025 ± 0.008	0.010 ± 0.003

Table 21: Threshold-independent metrics: AUC-ROC and AUC-PR values over the different methods for the two data sets, based on the test set. The standard deviation is also reported.

Supervised Methods	Elliptic AUC-ROC	Elliptic AUC-PR	IBM-AML AUC-ROC	IBM-AML AUC-PR
Egonet features	0.5172 ± 0.0607	0.0630 ± 0.0122	0.4758 ± 0.0187	0.0088 ± 0.0003
DeepWalk	0.5125 ± 0.0184	0.0597 ± 0.0021	0.5019 ± 0.0143	0.0092 ± 0.0004
Node2vec	0.4454 ± 0.0118	0.0511 ± 0.0021	0.4931 ± 0.0095	0.0091 ± 0.0003
GCN	0.5655 ± 0.0139	0.0696 ± 0.0039	0.4985 ± 0.0183	0.0092 ± 0.0007
GraphSAGE	0.5000 ± 0.0000	0.0565 ± 0.0025	0.5050 ± 0.0081	$\mathbf{0.0093 \pm 0.0003}$
GAT	0.5002 ± 0.0032	0.0566 ± 0.0018	0.5014 ± 0.0035	0.0091 ± 0.0002
GIN	$\mathbf{0.6177 \pm 0.0110}$	$\mathbf{0.0739 \pm 0.0049}$	$\mathbf{0.5061 \pm 0.0035}$	0.0092 ± 0.0003

The results for supervised learning are given in Table 21. The AUC values indicate that most methods based purely on the network structure are close to random.

One notable exception is GIN, which is specifically designed to better distinguish certain graph structures [144]. Graph structure is the only information available in this part of the experiments. The results for GIN indicate that there are some structural difference in the network for fraudulent and non-fraudulent nodes in the Elliptic data set. These structural difference are complex, since they are not picked up by the egonet features or the random-walk-based ones.

The other GNN architectures struggle, since they rely heavily on the node features. Especially GAT—which performed well with features—leverages node features for the attention calculations.

For the IBM data set, the differences are less pronounced that for the Elliptic data set. The network structure alone might not be enough to uncover money laundering. The data generation process of this synthetic data set creates a direct correlation between the money laundering pattern and feature values. Therefore, these feature values are more important than the network structure.

The results for the isolation forest in Table 22 show an improvement for the Elliptic data set, compared to the isolation forest with intrinsic features. Here, however, the models mostly make random predictions. When including the features, a better distinction can be made between money and non-money laundering. As mentioned before, the features of the money laundering transactions are so average (not anomalous according to the isolation forest) that they become suspicious again. The inverse predictions were in that case very good predictors. When only considering the network structure, it is harder to find meaningful anomalies. Hence, a higher AUC in this case points to a less useful model, as the model is as good as random.

The results on the IBM data set are in line with what we have seen before. The model makes almost random predictions, with deepwalk giving the best AUC-ROC and AUC-PR values.

This ablation study showed that it seems that network structure alone is not enough to come to a performant AML model. There is an important interplay between the network and the node features, pointing to a need to combine the two. Additionally, it highlighted the strong dependence of GNNs on these node features—except for the GIN in some cases—and on the data generation process. The explicit correlation between the money laundering cases and feature values for the IBM data set resulted in sub-optimal results when only considering the network structure.

5.4 Model Stability

As a final analysis, we look at the range of the results to assess model stability. Figure 9 present the box plots of the results for the models on the ten different training sets.

Table 22: Threshold-independent metrics: AUC-ROC and AUC-PR values over the different methods for the two data sets, based on the test set. The standard deviation is also reported.

Supervised Methods	**Elliptic**		**IBM-AML**	
	AUC-ROC	**AUC-PR**	**AUC-ROC**	**AUC-PR**
Egonet features	$\mathbf{0.5254 \pm 0.0120}$	0.0226 ± 0.0008	0.4737 ± 0.0057	0.0027 ± 0.0000
DeepWalk	0.4337 ± 0.0064	$\mathbf{0.0834 \pm 0.0013}$	$\mathbf{0.5380 \pm 0.0081}$	$\mathbf{0.0031 \pm 0.0001}$
Node2vec	0.4231 ± 0.0050	0.0815 ± 0.0010	0.4910 ± 0.0092	0.0026 ± 0.0001

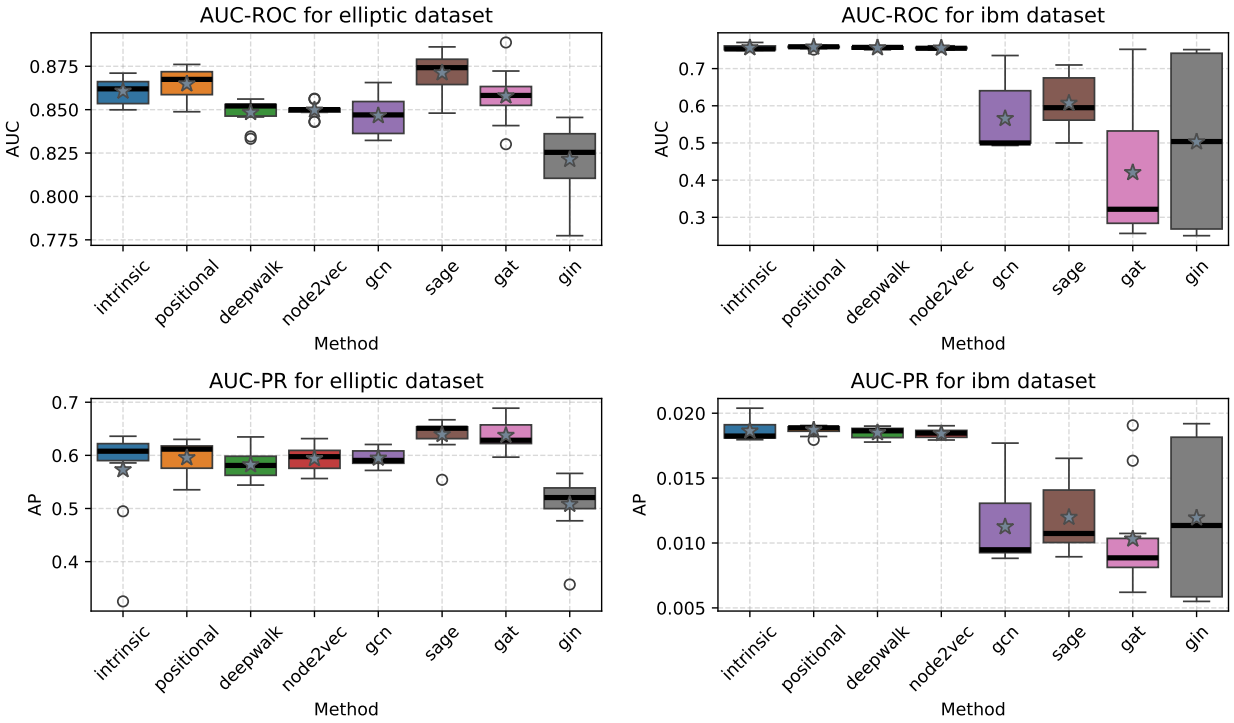


Figure 9: Box plots of the AUC-ROC and AUC-PR for the ten different training sets.

The results are quite stable on the Elliptic data set, although the AUC-ROC shows a bit more variability for the GNNs. The values on the IBM data set show strong instability for all GNNs, while the other models are relatively stable. The GNNs are probably less stable on the IBM data set due to the larger class imbalance.

6 Conclusion

6.1 Literature Review

This paper present a literature review on network analytics for anti-money laundering. The 97 selected papers were processed in two steps to achieve (1) a comprehensive understanding of the broader literature and (2) detailed insight into research that has been well received by the community. The comprehensive analysis classified all papers according to seven categories, each having different sub-characteristics. The detailed analysis was performed on the 10% most-cited papers in scope, which were further categorized using the framework recently proposed by Bockel-Rickermann et al. [14]. Information was given on the different processing steps taken, the challenges faced or addressed in the papers, the experiments performed, and the definition of nodes and edges in the networks.

A large part of the literature concerns methods that rely heavily on expert-knowledge by applying manual feature engineering and rule-based methods. Most of these are based on basic network centrality metrics.

The analysis and comparison of the models is mostly done using threshold-dependent metric, most notably accuracy, precision and recall. Given the class imbalance, we strongly advise against the use of accuracy as an evaluation metric

for AML models. Instead, the AML research should adopt the threshold-independent AUC-PR as a standard evaluation metric, as this is well suited for model comparison in the presence of high class imbalance.

Our analysis of the literature identified six critical gaps that future research must address:

1. **Benchmark**: There is a lack of comparison of novel methods with other models in the literature. This shows that there is a need for a unified approach to perform test including standard open-source data sets on which to test as well as a consensus on which methods can be seen as baseline models and which as state-of-the-art.
2. **Interpretability**: Few studies have explored tools to explain the outputs of graph neural networks (GNNs), which is essential for regulatory and operational transparency.
3. **Unsupervised Methods**: Limited exploration of classic unsupervised methods restricts their potential to detect anomalies, a key requirement as money laundering tactics evolve.
4. **Dynamic Network Analysis**: Most approaches rely on static networks, with insufficient research into dynamic methods that could better capture the temporal nature of transactions.
5. **Fraud Specific Methods**: The inherent incompleteness of AML labels, due to undetected cases and resource constraints, highlights the need for specialized PU-learning techniques, while limited resources necessitate actionable, prioritized predictions, which could be achieved through learning-to-rank methods.
6. **Adoption State-of-the-Art:** AML is often done using basic network metrics, while new developments in the field are adopted slowly. There is a need for standardising the use of heterogeneous and dynamic networks in AML. Future work should study how this can be achieved by comparing the latest state-of-the-art in continual graph learning, graph contrastive learning, heterogeneous and temporal GNNs, and graph transformers for AML, to name a few.

6.2 Experimental Evaluation

To extend upon the existing body of knowledge and to partially address the gaps in the literature, we implemented an extensive experimental evaluation covering manual feature engineering, shallow and deep representation learning, both in a supervised and unsupervised setting. We can that network features bring additional predictive power to AML models, when combined with the node features. However, the GNNs struggled when faced with extreme class imbalance and with a network with many hubs.

The observed performance improvements were limited, but even a small relative increase in performance has material implications for the business. The efficiency of AML practices increases significantly since investigators have more high quality predictions, and spend less time on the investigation of false positives.

In the unsupervised learning experiments, we employed isolation forests and observed notable differences between the real-world Elliptic data set and the synthetic IBM-AML data set. Although on synthetic data, the methods performed well, the observations from the Elliptic data seemed to exhibit strong camouflaging tactics. The anomaly scores of the latter were so low that investigating the least suspicious transactions for money laundering became the better strategy. We conclude from this that care should be taken when testing methods on synthetic data, since it can give overly optimistic results.

Our experiments on unsupervised learning are, however, limited. We only applied an isolation forest, which is a global anomaly detection method. Future work should focus on extending this to a larger benchmark, in line with the work of Lorenz et al. [83], to analyse if the same conclusions hold for other global and local anomaly detection methods.

The additional ablation study highlighted the need to incorporate both the network and feature information. When excluding the node features, the model predictions are as good as random. While this approach is common in the literature, we recommend against relying solely on the network structure.

6.3 Limitations

There are limitations to our work that should be addressed in future research.

First, conclusions are drawn based on only two data sets. To generalise our conclusions, additional tests need to be performed on a broader range of data sets. This can be done by including more open-source data sets when these become available, and rely on practitioners to use our code to test the methods on their proprietary data and publish the results.

Second, the decoders were always based on neural networks. The application and impact analysis of applying different machine learning methods as down-stream classifiers is another interesting area of study to extend the experiments presented here.

Third, the hyperparameter tuning process was limited. While Optuna provided an efficient method for hyperparameter search, the 50 to 100 tuning rounds used in this study may be insufficient. The choices in this paper were made to keep everything tractable and feasible given the limited resources available. Future work could expand the parameter ranges and number of tuning rounds.

Acknowledgments

This work was supported by the Research Foundation – Flanders (FWO research project 1SHEN24N) and by the BNP Paribas Fortis Chair in Fraud Analytics. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References

[1] Florian Adriaens, Cigdem Aslay, Tijl De Bie, Aristides Gionis, and Jefrey Lijffijt. Discovering interesting cycles in directed graphs. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1191–1200, 2019.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015. doi:10.1007/s10618-014-0365-y. URL https://doi.org/10.1007/s10618-014-0365-y.

[4] Ismail Alarab, Simant Prakoonwit, and Mohamed Ikbal Nacer. Comparative analysis using supervised learning methods for anti-money laundering in bitcoin. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, ICMLT '20, page 11–17, Beijing, China, 2020. Association for Computing Machinery. ISBN 9781450377645. doi:10.1145/3409073.3409078. URL https://doi.org/10.1145/3409073.3409078.

[5] Ismail Alarab, Simant Prakoonwit, and Mohamed Ikbal Nacer. Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, ICMLT '20, page 23–27, Beijing, China, 2020. Association for Computing Machinery. ISBN 9781450377645. doi:10.1145/3409073.3409080. URL https://doi.org/10.1145/3409073.3409080.

[6] Erik Altman, Jovan Blanuša, Luc von Niederhäusern, Beni Egressy, Andreea Anghel, and Kubilay Atasu. Realistic synthetic financial transactions for anti-money laundering models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 29851–29874. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5f38404edff6f3f642d6fa5892479c42-Paper-Datasets_and_Benchmarks.pdf.

[7] Eugenio Angriman, Alexander van der Grinten, Michael Hamann, Henning Meyerhenke, and Manuel Penschuck. *Algorithms for Large-Scale Network Analysis and the NetworKit Toolkit*, pages 3–20. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-21534-6. doi:10.1007/978-3-031-21534-6_1. URL https://doi.org/10.1007/978-3-031-21534-6_1.

[8] Henrique S Assumpção, Fabrício Souza, Leandro Lacerda Campos, Vinícius T de Castro Pires, Paulo M Laurentys de Almeida, and Fabricio Murai. Delator: Money laundering detection via multi-task learning on large transaction graphs. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 709–714. IEEE, 2022. doi:10.1109/BigData55660.2022.10021010.

[9] LAa Badalov, SDa Belov, and ISa Kadochnikov. Checking foreign counterparty companies using big data. In *CEUR Workshop Proceedings*, volume 2267, pages 523–527, 2018.

[10] Bart Baesens, Veronique Van Vlasselaer, and Wouter Verbeke. *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection*. John Wiley & Sons, Inc, 2015. ISBN 9781119133124. doi:10.1002/9781119146841.

[11] Ashwin Bahulkar, N Orkun Baycik, Thomas Sharkey, Yeming Shen, Boleslaw Szymanski, and William Wallace. Integrative analytics for detecting and disrupting transnational interdependent criminal smuggling, money, and money-laundering networks. In *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6. IEEE, 2018. doi:10.1109/THS.2018.8574121.

[12] Luigi Bellomarini, Davide Magnanimi, Markus Nissl, and Emanuel Sallinger. Neither in the programs nor in the data: Mining the hidden financial knowledge with knowledge graphs and reasoning. In *Mining Data for Financial Applications: 5th ECML PKDD Workshop, MIDAS 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers 5*, pages 119–134. Springer, 2021.

[13] Rasika Bhalerao, Maxwell Aliapoulios, Ilia Shumailov, Sadia Afroz, and Damon McCoy. Mapping the underground: Supervised discovery of cybercrime supply chains. In *2019 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–16. IEEE, 2019. doi:10.1109/eCrime47957.2019.9037582.

[14] Christopher Bockel-Rickermann, Tim Verdonck, and Wouter Verbeke. Fraud analytics: A decade of research: Organizing challenges and solutions in the field. *Expert Systems with Applications*, 232:120605, 2023. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2023.120605. URL https://www.sciencedirect.com/science/article/pii/S0957417423011077.

[15] Richard J. Bolton and David J. Hand. Statistical Fraud Detection: A Review. *Statistical Science*, 17(3):235 – 255, 2002. doi:10.1214/ss/1042727940. URL https://doi.org/10.1214/ss/1042727940.

[16] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *CoRR*, abs/2105.14491, 2021. URL https://arxiv.org/abs/2105.14491.

[17] Mehmet Çağlayan and Şerif Bahtiyar. Money laundering detection with node2vec. *Gazi University Journal of Science*, 35(3):854–873, 2022.

[18] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018. doi:10.1109/TKDE.2018.2807452.

[19] Ramiro Daniel Camino, Radu State, Leandro Montero, and Petko Valtchev. Finding suspicious activities in financial transactions and distributed ledgers. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 787–796. IEEE, 2017. doi:10.1109/ICDMW.2017.109.

[20] Mário Cardoso, Pedro Saleiro, and Pedro Bizarro. Laundrograph: Self-supervised graph representation learning for anti-money laundering. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, pages 130–138, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393768. doi:10.1145/3533271.3561727. URL https://doi.org/10.1145/3533271.3561727.

[21] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. ISSN 0360-0300. doi:10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

[22] Remco Chang, Mohammad Ghoniem, Robert Kosara, William Ribarsky, Jing Yang, Evan Suma, Caroline Ziemkiewicz, Daniel Kern, and Agus Sudjianto. Wirevis: Visualization of categorical, time-varying data from financial transactions. In *2007 IEEE symposium on visual analytics science and technology*, pages 155–162. IEEE, 2007. doi:10.1109/VAST.2007.4389009.

[23] Tat-Man Cheong and Yain-Whar Si. Event-based approach to money laundering data analysis and visualization. In *Proceedings of the 3rd International Symposium on Visual Information Communication*, VINCI '10, Beijing, China, 2010. Association for Computing Machinery. ISBN 9781450304368. doi:10.1145/1865841.1865869. URL https://doi.org/10.1145/1865841.1865869.

[24] Mariusz Chmielewski and Piotr Stkapor. Money laundering analytics based on contextual analysis. application of problem solving ontologies in financial fraud identification and recognition. In *Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology–ISAT 2016–Part I*, pages 29–39. Springer, 2017.

[25] Elliptic Dataset. Source code for torch_geometric.datasets.elliptic. https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/elliptic.html. Accessed: 2024-01-31.

[26] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 233–240, Pittsburgh, Pennsylvania, USA, 2006. ISBN 1595933832. doi:10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

[27] Min-Yuh Day. Artificial intelligence for knowledge graphs of cryptocurrency anti-money laundering in fintech. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '21, page 439–446, Virtual Event, Netherlands, 2022. Association for Computing Machinery. ISBN 9781450391283. doi:10.1145/3487351.3488415. URL https://doi.org/10.1145/3487351.3488415.

[28] Floris Devriendt, Jente Van Belle, Tias Guns, and Wouter Verbeke. Learning to rank for uplift modeling. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4888–4904, 2022. doi:10.1109/TKDE.2020.3048510.

[29] Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Pietro Palladino. An advanced network visualization system for financial crime detection. In *2011 IEEE Pacific visualization symposium*, pages 203–210. IEEE, 2011. doi:10.1109/PACIFICVIS.2011.5742391.

[30] Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. Vis4aui: Visual analysis of banking activity networks. In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications - Volume 1: IVAPP, (VISIGRAPP 2012)*, pages 799–802. INSTICC, SciTePress, 2012. ISBN 978-989-8565-02-0. doi:10.5220/0003933407990802.

[31] Walter Didimo, Luca Grilli, Giuseppe Liotta, Fabrizio Montecchiani, and Daniele Pagliuca. Visual querying and analysis of temporal fiscal networks. *Information Sciences*, 505:406–421, 2019. ISSN 0020-0255. doi:https://doi.org/10.1016/j.ins.2019.07.097. URL https://www.sciencedirect.com/science/article/pii/S0020025519307182.

[32] Peng Dong, Marie Loh, and Adrian Mondry. The "impact factor" revisited. *Biomedical Digital Libraries*, 2(1):7, 2005. doi:10.1186/1742-5581-2-7. URL https://doi.org/10.1186/1742-5581-2-7.

[33] Rafał Drężewski, Jan Sepielak, and Wojciech Filipkowski. The application of social network analysis algorithms in a system supporting money laundering detection. *Information Sciences*, 295:18–32, 2015. ISSN 0020-0255. doi:https://doi.org/10.1016/j.ins.2014.10.015. URL https://www.sciencedirect.com/science/article/pii/S0020025514009979.

[34] Bogdan Dumitrescu, Andra Băltoiu, and Ştefania Budulan. Anomaly detection in graphs of bank transactions for anti money laundering applications. *IEEE Access*, 10:47699–47714, 2022. doi:10.1109/ACCESS.2022.3170467.

[35] The Economist. Losing the war, April 2021. URL https://www.economist.com/finance-and-economics/2021/04/12/the-war-against-money-laundering-is-being-lost.

[36] Béni Egressy, Luc von Niederhäusern, Jovan Blanusa, Erik Altman, Roger Wattenhofer, and Kubilay Atasu. Provably powerful graph neural networks for directed multigraphs, 2024.

[37] Elliptic. Elliptic. www.elliptic.co. Accessed: 2024-01-31.

[38] Europol. Does crime still pay? criminal asset recovery in the eu, 2016.

[39] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[40] Emily Fletcher, Charles Larkin, and Shaen Corbet. Countering money laundering and terrorist financing: A case for bitcoin regulation. *Research in International Business and Finance*, 56:101387, 2021. ISSN 0275-5319. doi:https://doi.org/10.1016/j.ribaf.2021.101387. URL https://www.sciencedirect.com/science/article/pii/S0275531921000088.

[41] Andrea Fronzetti Colladon and Elisa Remondi. Using social network analysis to prevent money laundering. *Expert Systems with Applications*, 67:49–58, 2017. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2016.09.029. URL https://www.sciencedirect.com/science/article/pii/S0957417416305139.

[42] Zengan Gao and Mao Ye. A framework for data mining-based anti-money laundering research. *Journal of Money Laundering Control*, 10(2):170–179, 2007. doi:10.1108/13685200710746875. URL https://doi.org/10.1108/13685200710746875.

[43] Ignacio González García and Alfonso Mateos. Use of social network analysis for tax control in spain. *Hacienda Publica Espanola*, (239):159–197, 2021.

[44] Olmer Garcia-Bedoya, Oscar Granados, and José Cardozo Burgos. Ai against money laundering networks: the colombian case. *Journal of Money Laundering Control*, 24(1):49–62, 2021.

[45] Javier Garcia-Bernardo, Joost Witteman, and Marilou Vlaanderen. Uncovering the size of the illegal corporate service provider industry in the netherlands: a network approach. *EPJ Data Science*, 11(1):23, 2022.

[46] Ayshan Gasanova, Alexander N Medvedev, Evgeny I Komotskiy, Kamen B Spasov, and Igor N Sachkov. On the use of data mining methods for money laundering detection based on financial transactions information. In *AIP Conference Proceedings*, volume 2040, page 050021. AIP Publishing LLC, 2018.

[47] Peter Gerbrands, Brigitte Unger, Michael Getzner, and Joras Ferwerda. The effect of anti-money laundering policies: an empirical network analysis. *EPJ Data Science*, 11(1):15, 2022.

[48] Yanan Gong, Kam-Pui Chow, Hing-Fung Ting, and Siu-Ming Yiu. Analyzing the error rates of bitcoin clustering heuristics. In *Advances in Digital Forensics XVIII: 18th IFIP WG 11.9 International Conference, Virtual Event, January 3–4, 2022, Revised Selected Papers*, pages 187–205. Springer, 2022.

[49] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[50] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018. ISSN 0950-7051. doi:https://doi.org/10.1016/j.knosys.2018.03.022. URL https://www.sciencedirect.com/science/article/pii/S0950705118301540.

[51] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

[52] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, San Francisco, California, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi:10.1145/2939672.2939754. URL https://doi.org/10.1145/2939672.2939754.

[53] Andrea Gulino, Stefano Ceri, Georg Gottlob, Emanuel Sallinger, and Luigi Bellomarini. Distributed company control in company shareholding graphs. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2637–2648. IEEE, 2021. doi:10.1109/ICDE51399.2021.00294.

[54] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[55] Oussama H Hamid. Breaking through opacity: A context-aware data-driven conceptual design for a predictive anti money laundering system. In *2017 9th IEEE-GCC conference and exhibition (GCCCE)*, pages 1–9. IEEE, 2017. doi:10.1109/IEEEGCC.2017.8448084.

[56] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

[57] HuaLong Han, YuPeng Chen, ChenYing Guo, and Yin Zhang. Blockchain abnormal transaction behavior analysis: a survey. In *Blockchain and Trustworthy Systems: Third International Conference, BlockSys 2021, Guangzhou, China, August 5–6, 2021, Revised Selected Papers 3*, pages 57–69. Springer, 2021.

[58] Jing He, Jiao Tian, Yuanyuan Wu, Xinyi Cia, Kai Zhang, Mengjiao Guo, Hui Zheng, Junfeng Wu, and Yimu Ji. An efficient solution to detect common topologies in money launderings based on coupling and connection. *IEEE Intelligent Systems*, 36(1):64–74, 2021. doi:10.1109/MIS.2021.3057590.

[59] Tamer Hossam Helmy, Mohamed Zaki, Tarek Salah, and Khaled Badran. Design of a monitor for detecting money laundering and terrorist financing. *Journal of Theoretical & Applied Information Technology*, 85(3), 2016.

[60] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi:https://doi.org/10.1016/0893-6080(91)90009-T. URL https://www.sciencedirect.com/science/article/pii/089360809190009T.

[61] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi:https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

[62] Maryam Imanpour, Stephanie Rosenkranz, Bastian Westbrock, Brigitte Unger, and Joras Ferwerda. A microeconomic foundation for optimal money laundering policies. *International Review of Law and Economics*, 60:105856, 2019. ISSN 0144-8188. doi:https://doi.org/10.1016/j.irle.2019.105856. URL https://www.sciencedirect.com/science/article/pii/S0144818818302643.

[63] Angela SM Irwin and Adam B Turner. Illicit bitcoin transactions: challenges in getting to the who, what, when and where. *Journal of money laundering control*, 21(3):297–313, 2018. URL https://doi.org/10.1108/JMLC-07-2017-0031.

[64] Qing-Xuan Jia, Chun-Xu Chen, Xin Gao, Xin-Peng Li, Bo Yan, Guan-Qun Ai, Jun-Liang Li, and Jian-Hang Xu. Anomaly detection method using center offset measurement based on leverage principle. *Knowledge-Based Systems*, 190:105191, 2020. ISSN 0950-7051. doi:https://doi.org/10.1016/j.knosys.2019.105191. URL https://www.sciencedirect.com/science/article/pii/S0950705119305301.

[65] Chengxiang Jin, Jie Jin, Jiajun Zhou, Jiajing Wu, and Qi Xuan. Heterogeneous feature augmentation for ponzi detection in ethereum. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(9):3919–3923, 2022. doi:10.1109/TCSII.2022.3177898.

[66] Hai Jin, Chenchen Li, Jiang Xiao, Teng Zhang, Xiaohai Dai, and Bo Li. Detecting arbitrage on ethereum through feature fusion and positive-unlabeled learning. *IEEE Journal on Selected Areas in Communications*, 40(12): 3660–3671, 2022. doi:10.1109/JSAC.2022.3213335.

[67] Yaqin Jin and Zhenxin Qu. Research on anti-money laundering hierarchical model. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pages 406–411. IEEE, 2018. doi:10.1109/ICSESS.2018.8663895.

[68] Mikel Joaristi, Edoardo Serra, and Francesca Spezzano. Inferring bad entities through the panama papers network. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 767–773. IEEE, 2018. doi:10.1109/ASONAM.2018.8508497.

[69] Mikel Joaristi, Edoardo Serra, and Francesca Spezzano. Detecting suspicious entities in offshore leaks networks. *Social Network Analysis and Mining*, 9:1–15, 2019.

[70] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[71] Jerzy Korczak and Walter Łuszczyk. Visual exploration of cash flow chains. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 41–46. IEEE, 2011.

[72] Aditya Kurniawan, Muhammad Yusuf Handeu, Bayu Alvian, and Ade Andriani Renouw. Information system for depicting relations between banking customer with risk: Study case in indonesia. In *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pages 125–128. IEEE, 2015. doi:10.1109/ICoICT.2015.7231409.

[73] Eren Kurshan and Hongda Shen. Graph computing for financial crime and fraud detection: Trends, challenges and outlook. *International Journal of Semantic Computing*, 14(04):565–589, 2020.

[74] Dattatray Vishnu Kute, Biswajeet Pradhan, Nagesh Shukla, and Abdullah Alamri. Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. *IEEE Access*, 9: 82300–82317, 2021. doi:10.1109/ACCESS.2021.3086230.

[75] Michael Levi and Peter Reuter. Money laundering. *Crime and justice*, 34(1):289–375, 2006. doi:10.1086/501508.

[76] An Li, Zhongshuai Wang, Minghao Yu, and Di Chen. Blockchain abnormal transaction detection method based on weighted sampling neighborhood nodes. In *2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 746–752. IEEE, 2022. doi:10.1109/ICBAIE56435.2022.9985815.

[77] Qimai Li, Zhichao Han, and Xiao-ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi:10.1609/aaai.v32i1.11604. URL https://ojs.aaai.org/index.php/AAAI/article/view/11604.

[78] Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. Flowscope: Spotting money laundering based on graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4731–4738, 2020. doi:10.1609/aaai.v34i04.5906. URL https://ojs.aaai.org/index.php/AAAI/article/view/5906.

[79] Ziyu Li, Yanmei Zhang, Qian Wang, and Shiping Chen. Transactional network analysis and money laundering behavior identification of central bank digital currency of china. *Journal of Social Computing*, 3(3):219–230, 2022. doi:10.23919/JSC.2022.0011.

[80] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1), mar 2012. ISSN 1556-4681. doi:10.1145/2133360.2133363. URL https://doi.org/10.1145/2133360.2133363.

[81] Xuan Liu, Jia Li, Zhigao Chen, and Pengzhu Zhang. Research on financial super-network model based on variational inequalities. In Michael J. Shaw, Dongsong Zhang, and Wei T. Yue, editors, *E-Life: Web-Enabled Convergence of Commerce, Work, and Social Life*, pages 66–76, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-29873-8.

[82] Mark E. Lokanan. Financial fraud detection: the use of visualization techniques in credit card fraud and money laundering domains. *Journal of Money Laundering Control*, 26(3):436–444, 2023. doi:10.1108/JMLC-04-2022-0058. URL https://doi.org/10.1108/JMLC-04-2022-0058.

[83] Joana Lorenz, Maria Inês Silva, David Aparício, João Tiago Ascensão, and Pedro Bizarro. Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. doi:10.1145/3383455.3422549. URL https://doi.org/10.1145/3383455.3422549.

[84] Pritheega Magalingam, Stephen Davis, and Asha Rao. Using shortest path to discover criminal community. *Digital Investigation*, 15:1–17, 2015. ISSN 1742-2876. doi:https://doi.org/10.1016/j.diin.2015.08.002. URL https://www.sciencedirect.com/science/article/pii/S1742287615000894. Special Issue: Big Data and Intelligent Data Analysis.

[85] Shamil Magomedov, Sergei Pavelyev, Irina Ivanova, Alexey Dobrotvorsky, Marina Khrestina, and Timur Yusubaliev. Anomaly detection with machine learning and graph databases in fraud management. *International Journal of Advanced Computer Science and Applications*, 9(11), 2018.

[86] Maryam Mahootiha, Alireza Hashemi Golpayegani, and Babak Sadeghian. Designing a new method for detecting money laundering based on social network analysis. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–7. IEEE, 2021. doi:10.1109/CSICC52343.2021.9420621.

[87] Aili Malm and Gisela Bichler. Using friends for money: the positional importance of money-launderers in organized crime. *Trends in Organized Crime*, 16:365–381, 2013. doi:10.1007/s12117-013-9205-5. URL https://doi.org/10.1007/s12117-013-9205-5.

[88] Dan McGinn, David Birch, David Akroyd, Miguel Molina-Solana, Yike Guo, and William J Knottenbelt. Visualizing dynamic bitcoin transaction patterns. *Big data*, 4(2):109–119, 2016. doi:10.1089/big.2015.0056.

[89] Murad Mehmet and Duminda Wijesekera. Data analytics to detect evolving money laundering. In *STIDS*, pages 71–78, 2013.

[90] Murad Mehmet and Duminda Wijesekera. Using dynamic risk estimation & social network analysis to detect money laundering evolution. In *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pages 310–315. IEEE, 2013. doi:10.1109/THS.2013.6699020.

[91] Murad Mehmet, Murat Güneştaş, and Duminda Wijesekera. Dynamic risk model of money laundering. In *Risk Assessment and Risk-Driven Testing: First International Workshop, RISK 2013, Held in Conjunction with ICTSS 2013, Istanbul, Turkey, November 12, 2013. Revised Selected Papers 1*, pages 3–20. Springer, 2014.

[92] Giovanni Micale, Alfredo Pulvirenti, Alfredo Ferro, Rosalba Giugno, and Dennis Shasha. Fast methods for finding significant motifs on labelled multi-relational networks. *Journal of Complex Networks*, 7(6):817–837, 03 2019. ISSN 2051-1329. doi:10.1093/comnet/cnz008. URL https://doi.org/10.1093/comnet/cnz008.

[93] Krzysztof Michalak and Jerzy Korczak. Graph mining approach to suspicious transaction detection. In *2011 Federated conference on computer science and information systems (FedCSIS)*, pages 69–75. IEEE, 2011.

[94] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[95] Anuraj Mohan, Karthika P. V., Parvathi Sankar, K. Maya Manohar, and Amala Peter. Improving anti-money laundering in bitcoin using evolving graph convolutions and deep neural decision forest. *Data Technologies and Applications*, 57(3):313–329, 2023. doi:10.1108/DTA-06-2021-0167. URL https://doi.org/10.1108/DTA-06-2021-0167.

[96] Philippe Mongeon and Adèle Paul-Hus. The journal coverage of web of science and scopus: a comparative analysis. *Scientometrics*, 106(1):213–228, 2016. doi:10.1007/s11192-015-1765-5. URL https://doi.org/10.1007/s11192-015-1765-5.

[97] M Nandhini and Bikram Bikash Das. An assessment and methodology for fraud detection in online social network. In *2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM)*, pages 104–108. IEEE, 2016. doi:10.1109/ICONSTEM.2016.7560932.

[98] Mark Newman. *Networks: An Introduction*. Oxford University Press, 03 2010. ISBN 9780199206650. doi:10.1093/acprof:oso/9780199206650.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199206650.001.0001.

[99] E.W.T. Ngai, Yong Hu, Y.H. Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support*

Systems, 50(3):559–569, 2011. ISSN 0167-9236. doi:https://doi.org/10.1016/j.dss.2010.08.006. URL https://www.sciencedirect.com/science/article/pii/S0167923610001302. On quantitative methods for detection of financial fraud.

[100] Normah Omar, Ismail bin Mohamed, Zuraidah Mohd Sanusi, and Hendi Yogi Prabowo. Understanding social network analysis (sna) in fraud detection. In *Proceedings of the International Congress on Interdisciplinary Behaviour and Social Sciences*, pages 543–548, 2014.

[101] Michael Ovelgönne, Chanhyun Kang, Anshul Sawant, and VS Subrahmanian. Covertness centrality in networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 863–870. IEEE, 2012. doi:10.1109/ASONAM.2012.156.

[102] Brice Ozenne, Fabien Subtil, and Delphine Maucort-Boulch. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *Journal of clinical epidemiology*, 68(8):855–859, 2015.

[103] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[104] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, New York, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi:10.1145/2623330.2623732. URL https://doi.org/10.1145/2623330.2623732.

[105] Silivanxay Phetsouvanh, Frédérique Oggier, and Anwitaman Datta. Egret: Extortion graph exploration techniques in the bitcoin network. In *2018 IEEE International conference on data mining workshops (ICDMW)*, pages 244–251. IEEE, 2018. doi:10.1109/ICDMW.2018.00043.

[106] Clifton Phua, Vincent C. S. Lee, Kate Smith-Miles, and Ross W. Gayler. A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119, 2010. URL http://arxiv.org/abs/1009.6119.

[107] Kirill Plaksiy, Andrey Nikiforov, and Natalia Miloslavskaya. Applying big data technologies to detect cases of money laundering and counter financing of terrorism. In *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 70–77. IEEE, 2018.

[108] Mario Alfonso Prado-Romero and Andrés Gago-Alonso. Community feature selection for anomaly detection in attributed graphs. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 21st Iberoamerican Congress, CIARP 2016, Lima, Peru, November 8–11, 2016, Proceedings 21*, pages 109–116. Springer, 2017.

[109] M Mazhar Rathore, Sushil Chaurasia, and Dhirendra Shukla. Mixers detection in bitcoin network: a step towards detecting money laundering in crypto-currencies. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5775–5782. IEEE, 2022. doi:10.1109/BigData55660.2022.10020982.

[110] Piergiorgio Ricci. How economic freedom reflects on the bitcoin transaction network. *Journal of Industrial and Business Economics*, 47(1):133–161, 2020.

[111] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[112] A Semenov, D Doropheev, A Mazeev, and T Yusubaliev. Survey of common design approaches in aml software development. In *CEUR Workshop Proceedings*, pages 1–9, 2017.

[113] Ted E Senator, Henry G Goldberg, Jerry Wooton, Matthew A Cottini, AF Umar Khan, Christina D Klinger, Winston M Llamas, Michael P Marrone, and Raphael WH Wong. Financial crimes enforcement network ai system (fais) identifying potential money laundering from reports of large cash transactions. *AI magazine*, 16(4): 21–21, 1995. doi:10.1609/aimag.v16i4.1169.

[114] Abdul K Shaikh and Amril Nazir. A model for identifying relationships of suspicious customers in money laundering using social network functions. In *Proceedings of the world congress on engineering*, volume 1, pages 4–7, 2018.

[115] Abdul Khalique Shaikh, Malik Al-Shamli, and Amril Nazir. Designing a relational model to identify relationships between suspicious customers in anti-money laundering (aml) using social network analysis (sna). *Journal of Big Data*, 8:1–22, 2021.

[116] Yeming Shen, Thomas C. Sharkey, Boleslaw K. Szymanski, and William (Al) Wallace. Interdicting interdependent contraband smuggling, money and money laundering networks. *Socio-Economic Planning Sciences*, 78:101068, 2021. ISSN 0038-0121. doi:https://doi.org/10.1016/j.seps.2021.101068. URL https://www.sciencedirect.com/science/article/pii/S0038012121000604.

[117] Rafael Sousa Lima, André Luiz Marques Serrano, Joshua Onome Imoniana, and César Medeiros Cupertino. Identifying financial patterns of money laundering with social network analysis: A brazilian case study. *Journal of Money Laundering Control*, 25(1):118–134, 2022.

[118] Michele Starnini, Charalampos E Tsourakakis, Maryam Zamanipour, André Panisson, Walter Allasia, Marco Fornasiero, Laura Li Puma, Valeria Ricci, Silvia Ronchiadin, Angela Ugrinoska, et al. Smurf-based anti-money laundering in time-evolving transaction networks. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21*, pages 171–186. Springer, 2021.

[119] Kai Sun, Kun Meng, and Ziqiang Zheng. Game-bc: A graph attention model for exploring bitcoin crime. In *2022 6th International Symposium on Computer Science and Intelligent Control (ISCSIC)*, pages 342–346. IEEE, 2022. doi:10.1109/ISCSIC57216.2022.00077.

[120] Haseeb Tariq and Marwan Hassani. Topology-agnostic detection of temporal money laundering flows in billion-scale transactions. In Rosa Meo and Fabrizio Silvestri, editors, *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 402–419, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-74643-7. doi:https://doi.org/10.1007/978-3-031-74643-7_29.

[121] Jeyakumar Samantha Tharani, Eugene Yougarajah Andrew Charles, Zhé Hóu, Marimuthu Palaniswami, and Vallipuram Muthukkumarasamy. Graph based visualisation techniques for analysis of blockchain transactions. In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, pages 427–430. IEEE, 2021. doi:10.1109/LCN52139.2021.9524878.

[122] Elena Tiukhova, Manon Reusens, Bart Baesens, and Monique Snoeck. Benchmarking conventional outlier detection methods. In Renata Guizzardi, Jolita Ralyté, and Xavier Franch, editors, *Research Challenges in Information Science*, pages 597–613, Cham, 2022. Springer International Publishing. ISBN 978-3-031-05760-1.

[123] Elena Tiukhova, Emiliano Penaloza, María Óskarsdóttir, Bart Baesens, Monique Snoeck, and Cristián Bravo. Inflect-dgnn: Influencer prediction with dynamic graph neural networks. *IEEE Access*, 2024. doi:10.1109/ACCESS.2024.3443533.

[124] Adam B Turner, Stephen McCombie, and Allon J Uhlmann. A target-centric intelligence approach to wannacry 2.0. *Journal of Money Laundering Control*, 22(4):646–665, 2019.

[125] Adam B Turner, Stephen McCombie, and Allon J Uhlmann. Discerning payment patterns in bitcoin from ransomware attacks. *Journal of Money Laundering Control*, 23(3):545–589, 2020.

[126] United Nations Office on Drugs and Crime. Money laundering. https://www.unodc.org/unodc/en/money-laundering/overview.html. Accessed: 2023-04-07.

[127] Rafaël Van Belle, Sandra Mitrović, and Jochen De Weerdt. Representation learning in graphs for credit card fraud detection. In *Mining Data for Financial Applications: 4th ECML PKDD Workshop, MIDAS 2019, Würzburg, Germany, September 16, 2019, Revised Selected Papers 4*, pages 32–46. Springer, 2020.

[128] Rafaël Van Belle, Charles Van Damme, Hendrik Tytgat, and Jochen De Weerdt. Inductive graph representation learning for fraud detection. *Expert Systems with Applications*, 193:116463, 2022. doi:https://doi.org/10.1016/j.eswa.2021.116463.

[129] Rafaël Van Belle, Bart Baesens, and Jochen De Weerdt. Catchm: A novel network-based credit card fraud detection method using node representation learning. *Decision Support Systems*, 164:113866, 2023. ISSN 0167-9236. doi:https://doi.org/10.1016/j.dss.2022.113866. URL https://www.sciencedirect.com/science/article/pii/S0167923622001373.

[130] Toon Vanderschueren, Bart Baesens, Tim Verdonck, and Wouter Verbeke. A new perspective on classification: Optimally allocating limited resources to uncertain tasks. *Decision Support Systems*, 179:114151, 2024. ISSN 0167-9236. doi:https://doi.org/10.1016/j.dss.2023.114151. URL https://www.sciencedirect.com/science/article/pii/S0167923623002269.

[131] Rafael B Velasco, Igor Carpanese, Ruben Interian, Octavio CG Paulo Neto, and Celso C Ribeiro. A decision support system for fraud detection in public procurement. *International Transactions in Operational Research*, 28(1):27–47, 2021.

[132] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[133] Zhonglin Wang, Niu Guiqian, Zhou Yan, and Yaqiong Mu. Detection mechanism of money laundering based on random walk and skip-grim model. In *2022 IEEE 5th International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 444–448. IEEE, 2022. doi:10.1109/ICEICT55736.2022.9909113.

[134] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics, 2019.

[135] Ollie A Williams. Watchdogs hit money launderers with record fines, Aug 2021. URL https://www.forbes.com/sites/oliverwilliams1/2021/08/26/money-laundering-thrived-during-covid-pandemic/.

[136] Donald Winiecki, Katherine Kappelman, Bryant Hay, Mikel Joaristi, Edoardo Serra, and Francesca Spezzano. Validating bad entity ranking in the panama papers via open-source intelligence. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 752–759. IEEE, 2020. doi:10.1109/ASONAM49781.2020.9381389.

[137] Jiajing Wu, Jieli Liu, Weili Chen, Huawei Huang, Zibin Zheng, and Yan Zhang. Detecting mixing services via mining bitcoin transaction network with hybrid motifs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2237–2249, 2021.

[138] Lei Wu, Yufeng Hu, Yajin Zhou, Haoyu Wang, Xiapu Luo, Zhi Wang, Fan Zhang, and Kui Ren. Towards understanding and demystifying bitcoin mixing services. In *Proceedings of the Web Conference 2021*, WWW '21, page 33–44, Ljubljana, Slovenia, 2021. ISBN 9781450383127. doi:10.1145/3442381.3449880. URL https://doi.org/10.1145/3442381.3449880.

[139] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. doi:10.1109/TNNLS.2020.2978386.

[140] Pingfan Xia, Zhiwei Ni, Hongwang Xiao, Xuhui Zhu, and Peng Peng. A novel spatiotemporal prediction approach based on graph convolution neural networks and long short-term memory for money laundering fraud. *Arabian Journal for Science and Engineering*, 47(2):1921–1937, 2022. doi:10.1007/s13369-021-06116-2. URL https://doi.org/10.1007/s13369-021-06116-2.

[141] Zhao Xiao, Yuelei Li, and Kang Zhang. Visual analysis of risks in peer-to-peer lending market. *Personal and Ubiquitous Computing*, 22:825–838, 2018.

[142] Jianying Xiong and Wen Xiao. Identification of key nodes in abnormal fund trading network based on improved pagerank algorithm. *Journal of Physics: Conference Series*, 1774(1):012001, jan 2021. doi:10.1088/1742-6596/1774/1/012001. URL https://dx.doi.org/10.1088/1742-6596/1774/1/012001.

[143] Jianying Xiong and Haifeng Zhong. Identification of money laundering accounts based on weighted capital flow network. *Journal of Physics: Conference Series*, 1629(1):012023, sep 2020. doi:10.1088/1742-6596/1629/1/012023. URL https://dx.doi.org/10.1088/1742-6596/1629/1/012023.

[144] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[145] Xie Xueshuo, Wang Jiming, Ye Junyi, Fang Yaozheng, Lu Ye, Li Tao, and Wang Guiling. Awap: Adaptive weighted attribute propagation enhanced community detection model for bitcoin de-anonymization. *Applied Soft Computing*, 109:107507, 2021. ISSN 1568-4946. doi:https://doi.org/10.1016/j.asoc.2021.107507. URL https://www.sciencedirect.com/science/article/pii/S1568494621004300.

[146] Qingqing Yang, Yuexin Xiang, Wenmao Liu, and Wei Ren. An illicit bitcoin address analysis scheme based on subgraph evolution. In *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 679–686. IEEE, 2022. doi:10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00116.

[147] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 40–48, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/yanga16.html.

[148] Yahan Yu, Yixuan Xu, Jian Wang, Zhenxing Li, and Bin Cao. Anti-money laundering risk identification of financial institutions based on aspect-level graph neural networks. In *2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, pages 542–546. IEEE, 2022. doi:10.1109/QRS-C57518.2022.00086.

[149] Shilei Zhang, Toyotaro Suzumura, and Li Zhang. Dyngraphtrans: Dynamic graph embedding via modified universal transformer networks for financial transaction data. In *2021 IEEE International Conference on Smart Data Services (SMDS)*, pages 184–191. IEEE, 2021. doi:10.1109/SMDS53860.2021.00032.

[150] Maria Zhdanova, Jürgen Repp, Roland Rieke, Chrystel Gaber, and Baptiste Hemery. No smurfs: Revealing fraud chains in mobile money transfers. In *2014 Ninth International Conference on Availability, Reliability and Security*, pages 11–20. IEEE, 2014. doi:10.1109/ARES.2014.10.

[151] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. A local algorithm for structure-preserving graph cut. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 655–664, Halifax, NS, Canada, 2017. ISBN 9781450348874. doi:10.1145/3097983.3098015. URL https://doi.org/10.1145/3097983.3098015.

[152] Jiajun Zhou, Chenkai Hu, Jianlei Chi, Jiajing Wu, Meng Shen, and Qi Xuan. Behavior-aware account de-anonymization on ethereum interaction graph. *IEEE Transactions on Information Forensics and Security*, 17: 3433–3448, 2022. doi:10.1109/ASONAM.2012.156.

[153] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi:https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.

[154] Yadong Zhou, Ximi Wang, Junjie Zhang, Peng Zhang, Lili Liu, Huan Jin, and Hongbo Jin. Analyzing and detecting money-laundering accounts in online social networks. *IEEE Network*, 32(3):115–121, 2017. doi:10.1109/MNET.2017.1700213.

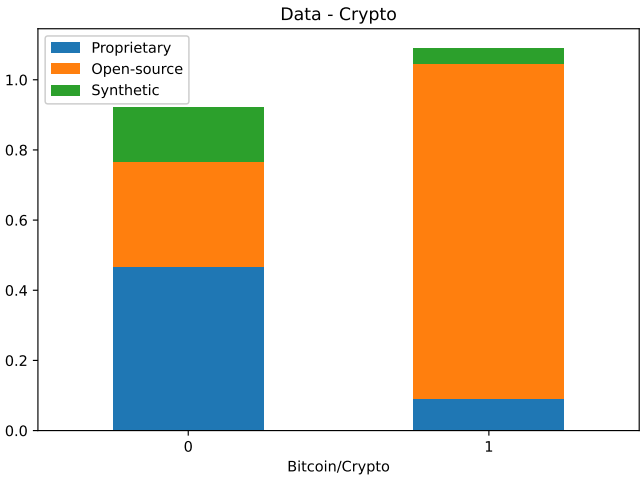
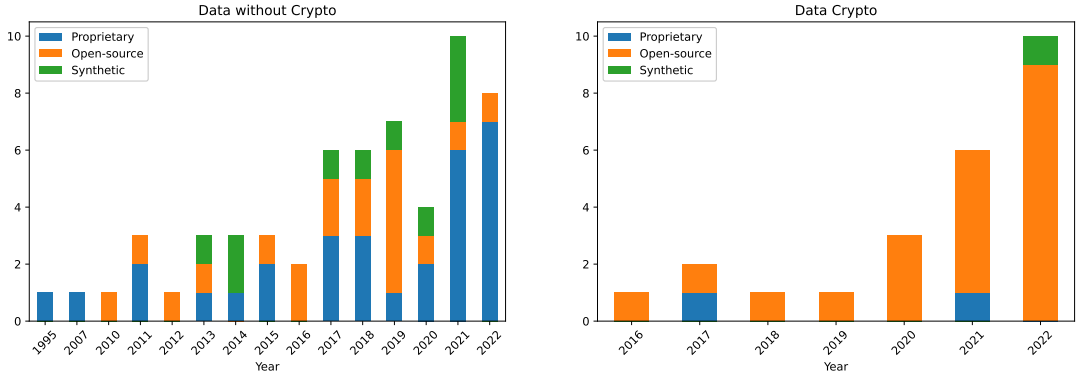


Figure A1: The data for papers dealing with crypto currencies (1) and those that do not (0).

A Analysis of Categories between Crypto and Non-Crypto Papers

This part of the appendix gives the plots in which we compare the difference in nature of the crypto-literature compared non-crypto-related research.



(a) The evolution of the data over the years for papers not covering crypto currencies.

(b) The evolution of the data over the years for papers covering crypto currencies.

Figure A2: Evolution of the data.

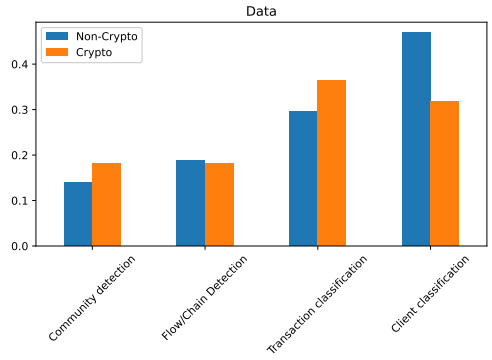


Figure A3: The distribution of the objective of the papers.

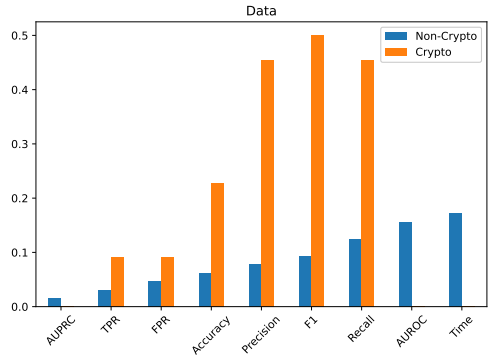


Figure A4: The distribution of the evaluation metrics used.

B The Top-Cited Papers

In this part of the appendix, we give the figures that summarise the top-cited papers.

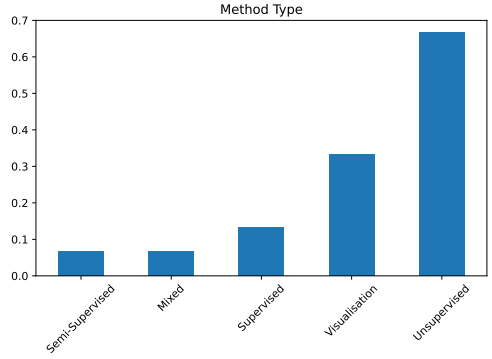


Figure B1: The methods for the top-cited papers.

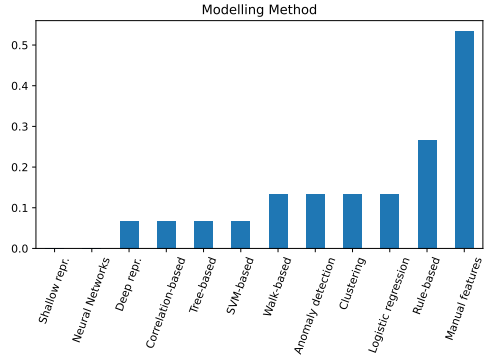


Figure B2: The modelling methods for the top-cited papers.

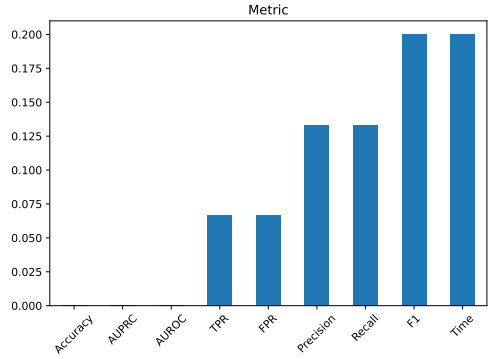


Figure B3: The performance metrics for the top-cited papers.

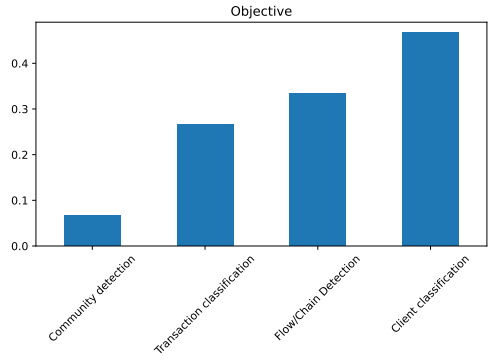


Figure B4: The objective of the top-cited papers.

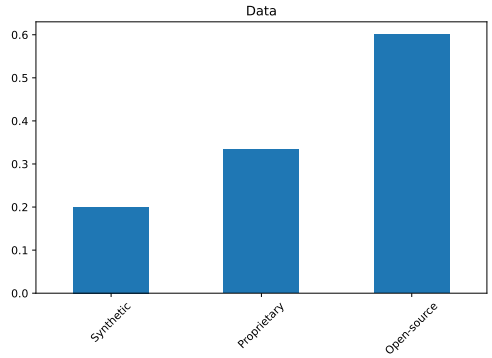


Figure B5: The nature of the data of the top-cited papers.

Table C1: Threshold-dependent metrics: Precision, recall and F1-score values for the top 0.1% scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	0.5396 ± 0.3093	0.0204 ± 0.0302	0.0383 ± 0.0542
Egonet features	0.6833 ± 0.3187	0.0129 ± 0.0060	0.0253 ± 0.0118
DeepWalk	0.5333 ± 0.3148	0.0101 ± 0.0059	0.0198 ± 0.0117
Node2vec	0.5167 ± 0.3352	0.0097 ± 0.0063	0.0191 ± 0.0124
GCN	$\mathbf{1.0000 \pm 0.0000}$	0.0189 ± 0.0007	0.0371 ± 0.0014
GraphSAGE	0.9624 ± 0.0615	$\mathbf{0.0300 \pm 0.0071}$	$\mathbf{0.0580 \pm 0.0133}$
GAT	$\mathbf{1.0000 \pm 0.0000}$	0.0187 ± 0.0005	0.0367 ± 0.0010
GIN	0.8500 ± 0.2540	0.0191 ± 0.0093	0.0372 ± 0.0179

Table C2: Threshold-dependent metrics: Precision, recall and F1-score values for the top 0.1% scores over the different methods for the IBM-AML data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	0.0050 ± 0.0087	0.0007 ± 0.0011	0.0012 ± 0.0019
Egonet features	0.0079 ± 0.0085	0.0011 ± 0.0012	0.0019 ± 0.0020
DeepWalk	0.0076 ± 0.0085	0.0011 ± 0.0013	0.0019 ± 0.0022
Node2vec	0.0035 ± 0.0058	0.0006 ± 0.0009	0.0010 ± 0.0016
GCN	$\mathbf{0.0103 \pm 0.0047}$	$\mathbf{0.8815 \pm 0.3117}$	$\mathbf{0.0203 \pm 0.0092}$
GraphSAGE	0.0057 ± 0.0062	0.4919 ± 0.5188	0.0112 ± 0.0123
GAT	0.0054 ± 0.0087	0.1798 ± 0.3688	0.0095 ± 0.0156
GIN	0.0044 ± 0.0080	0.0102 ± 0.0131	0.0016 ± 0.0015

C Results Threshold-Dependent Metrics

In this part of the appendix, we give the precision, recall and F1-scores for the models using different thresholds. The thresholds are set such that the observations with the top 0.1%, 10% and $p\%$ of scores are classified as money laundering, where $p\%$ is equal to the relative occurrence of the labels.

Table C3: Threshold-dependent metrics: Precision, recall and F1-score values for the top 10% scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	0.3526 ± 0.0163	0.6204 ± 0.0287	0.4497 ± 0.0208
Egonet features	0.3559 ± 0.0101	0.6263 ± 0.0179	0.4539 ± 0.0129
DeepWalk	0.3517 ± 0.0050	0.6189 ± 0.0089	0.4485 ± 0.0064
Node2vec	0.3536 ± 0.0040	0.6222 ± 0.0071	0.4509 ± 0.0051
GCN	0.3471 ± 0.0272	0.5988 ± 0.0358	0.4393 ± 0.0306
GraphSAGE	$\mathbf{0.3675 \pm 0.0118}$	$\mathbf{0.6395 \pm 0.0207}$	$\mathbf{0.4666 \pm 0.0124}$
GAT	0.3661 ± 0.0247	0.6388 ± 0.0288	0.4653 ± 0.0270
GIN	0.3307 ± 0.0228	0.5889 ± 0.0407	0.4234 ± 0.0274

Table C4: Threshold-dependent metrics: Precision, recall and F1-score values for the top 10% scores over the different methods for the IBM-AML data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	$\mathbf{0.0148 \pm 0.0006}$	0.1635 ± 0.0063	$\mathbf{0.0272 \pm 0.0011}$
Egonet features	$\mathbf{0.0148 \pm 0.0002}$	0.1626 ± 0.0026	0.0271 ± 0.0004
DeepWalk	0.0145 ± 0.0003	0.1602 ± 0.0033	0.0267 ± 0.0005
Node2vec	$\mathbf{0.0148 \pm 0.0006}$	0.1630 ± 0.0065	0.0271 ± 0.0011
GCN	0.0109 ± 0.0030	0.9814 ± 0.0374	0.0214 ± 0.0058
GraphSAGE	0.0121 ± 0.0026	0.9761 ± 0.0257	0.0238 ± 0.0051
GAT	0.0054 ± 0.0074	$\mathbf{0.1993 \pm 0.3722}$	0.0104 ± 0.0144
GIN	0.0081 ± 0.0077	0.0947 ± 0.0793	0.0149 ± 0.0141

Table C5: Threshold-dependent metrics: Precision, recall and F1-score values for the top $p = 2\%$ scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	0.3244 ± 0.0143	0.6303 ± 0.0277	0.4283 ± 0.0188
Egonet features	0.3271 ± 0.0098	0.6357 ± 0.0190	0.4319 ± 0.0129
DeepWalk	0.3220 ± 0.0052	0.6258 ± 0.0102	0.4252 ± 0.0069
Node2vec	0.3241 ± 0.0046	0.6299 ± 0.0089	0.4280 ± 0.0060
GCN	0.3209 ± 0.0284	0.6230 ± 0.0371	0.4234 ± 0.0325
GraphSAGE	0.3343 ± 0.0130	0.6558 ± 0.0166	0.4427 ± 0.0121
GAT	$\mathbf{0.3419 \pm 0.0084}$	$\mathbf{0.6613 \pm 0.0212}$	$\mathbf{0.4507 \pm 0.0109}$
GIN	0.3034 ± 0.0169	0.5949 ± 0.0285	0.4018 ± 0.0205

Table C6: Threshold-dependent metrics: Precision, recall and F1-score values for the top $p = 0.11\%$ scores over the different methods for the Elliptic data set, based on the test set. The standard deviation is also reported.

Methods	Precision	Recall	F1-score
Intrinsic features	0.0056 ± 0.0090	0.0008 ± 0.0012	0.0014 ± 0.0021
Egonet features	0.0085 ± 0.0077	0.0012 ± 0.0011	0.0021 ± 0.0019
DeepWalk	0.0080 ± 0.0076	0.0013 ± 0.0013	0.0023 ± 0.0021
Node2vec	0.0047 ± 0.0063	0.0008 ± 0.0010	0.0013 ± 0.0018
GCN	$\mathbf{0.0103 \pm 0.0046}$	$\mathbf{0.8811 \pm 0.3119}$	$\mathbf{0.0203 \pm 0.0089}$
GraphSAGE	0.0055 ± 0.0061	0.4907 ± 0.5175	0.0108 ± 0.0121
GAT	0.0047 ± 0.0078	0.1763 ± 0.3656	0.0085 ± 0.0146
GIN	0.0058 ± 0.0084	0.0091 ± 0.0116	0.0019 ± 0.0016