HOW MONEY FLOW STATISTICS CAN BE USED TO DETECT MONEY LAUNDERING ACTIVITY IN GRAPH-BASED FINANCIAL CRIME DETECTION

LUCA-SERBAN IONESCU[1] (IONESCU-8@STUDENT.TUDELFT.NL), KUBILAY ATASU[1] (SUPERVISOR), ZEKI ERKIN[1] (RESPONSIBLE PROFESSOR)
[1] EEMCS, DELFT UNIVERSITY OF TECHNOLOGY, THE NETHERLANDS

INTRODUCTION

- **Money laundering(ML)** refers to the movement of illicit funds to conceal their origin and make them appear to come from legitimate sources. [1]
- In a **graph representation** of a financial network, **vertices** represent **accounts** and **edges** represent **transactions**.
- In order to **detect** money laundering in graph representations of financial data, two approaches have been explored:
  - **Supervised**, which requires **labelled data** to be trained.
  - **Unsupervised**, which relies on **graph structure**, rather than label-based training

Unsupervised approaches include **dense subgraph detection** algorithms, which are vulnerable to **camouflage** and **graph mining** approaches, which are limited to a **fixed set** of **laundering patterns.**

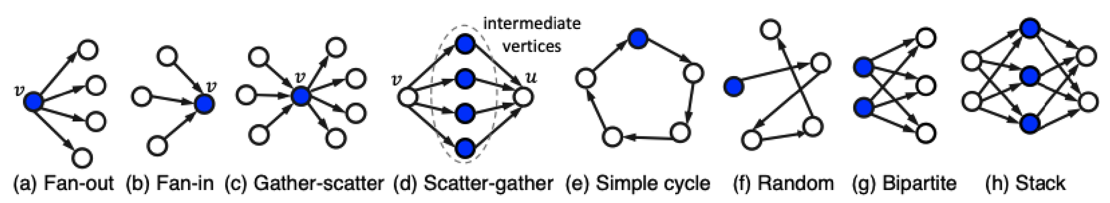

Figure 1 - Laundering patterns [1]

Supervised approaches include **Graph Neural Networks**. They require a **large number** of both positive and negative **labels**, which in general can be **costly**, and even **impossible**, to acquire. [3]

Literature proposes **flow statistics analysis,** an **unsupervised** technique which allows for detection of money laundering in **multi-step transfers** without relying on pre-defined **subgraph patterns**.

The paper aims to explore **how** the **flow statistics**-based methods can be **used** for money **laundering detection** by answering ***two research sub-questions***:

1. ***What are the existing solutions using money flow statistics?***
2. ***How would the money flow statistics methods perform on a realistic dataset of transactions?***

PRELIMINARIES

Directed multigraphs

- Directed graphs which allow **multiple edges** between the **same** two **accounts** and **cycles** within the graph.

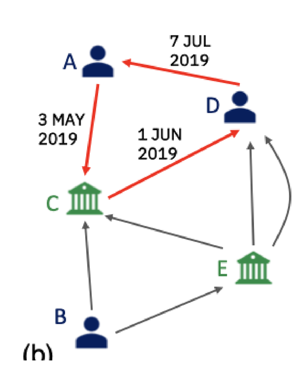

Figure 2 - Directed financial multigraph [1]

Multipartite graphs

- Directed graphs with **vertex** set **partitioned** in a number of **disjoint partitions**.
- A node of partition **i** can **only** have **edges towards** nodes of partition **i+1**.

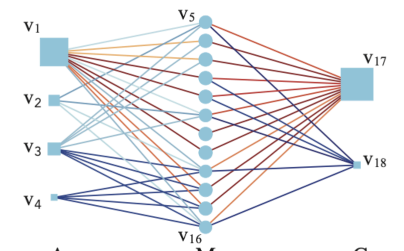

Figure 3 - Tripartite financial graph [4]

Tensors

- N-dimensional arrays which are able to express **multiple relations** of any order.
- **Graphs** can only represent **source** and **destination**, **tensors** can represent **multiple dimensions** (e.g. **time**).

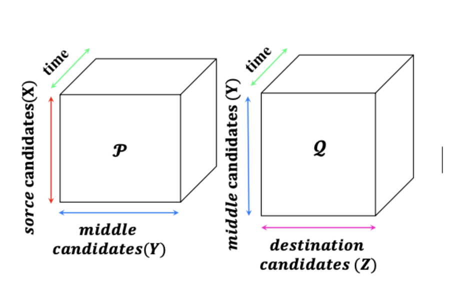

Figure 4 - Pair of coupled tensors [5]

Money Flow Statistics

- Information obtained from **analysing** the amount of **money received** (*inflow*) or **sent** (*outflow*) by an account.

Three properties of money laundering [5]:

- **Density:** Due to limited amount of available accounts, middle accounts have a very dense inflow and outflow.
- **Zero-Out:** Middle accounts transfer out most of their inflow.
- **Fast-In/Fast-Out:** Transfers through middle accounts happen quickly.

Network Flow Analysis

- Transfer of money from a source ***s*** to a sink ***t*** through a network obeying **capacity constraints** at edges and **conservation** at nodes.
- The **maximum flow problem** focuses on the **maximum** amount of **money flowing** from the **source** that can fully reach the **sink**, while **obeying all constraints** of the network
- Maximum flow algorithms determine the **maximum** amount of **flow**, along with all **edges** that **carry** it

BACKGROUND

FlowScope [4]

- Works on **multipartite** graphs of length k.
- Identifies suspicious nodes with **high volume** (high *inflow & outflow* → ***Density***) and **low retention** (low amount of money left in the middle account, max(*inflow*, *outflow*) - min(*inflow*, *outflow*) → ***Zero-Out***)
- Returns the **subgraph** with the **highest average anomalousness**. (highlighted in Fig. 5)

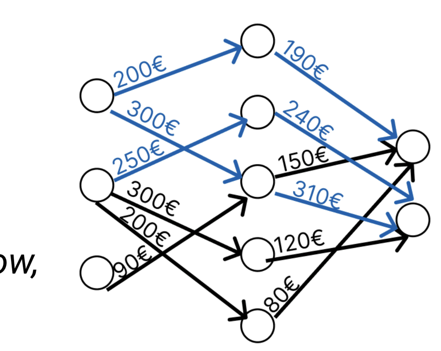

Figure 5 - Subgraph identified by FlowScope

CubeFlow [5]

- Works on pairs of coupled tensors .
- Same logic as FlowScope for considering ***Density*** and ***Zero-Out.*** Additionally considers time through **time binning** of transactions.
- Performs the extraction of **suspicious blocks** on transactions that happen in the **same time bin**, to satisfy ***Fast-In/Fast-Out.*** (highlighted in Fig. 6)

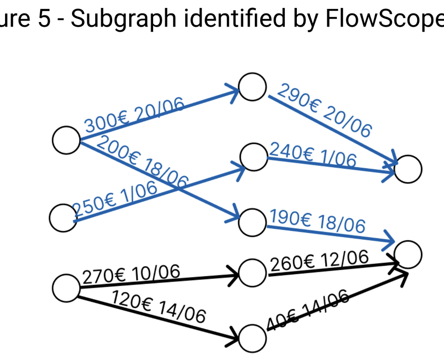

Figure 6 - Subgraph identified by CubeFlow

WeirdFlows [6]

- Used to generate a top-down **search pipeline** which highlights the **amount of flow** on a specific **path**, during a specific **interval** (shown in Fig. 7)
- Computes **all paths** between two nodes, calculates **maximum flow** as the **sum** of the **minimum edge** weights on each path and **compares** within **multiple time bins.**

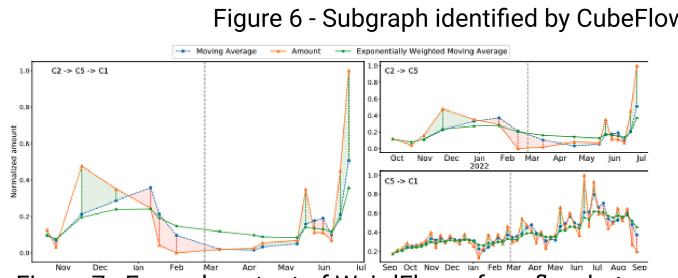

Figure 7 - Example output of WeirdFlows, for a flow between nodes C2 and C1, through a middle node C5 [6]

DenseFlow [7]

- Extracts a **dense subset** of **suspicious** nodes **S*** from a directed multigraph based on a **joint suspiciousness** composed of **topological, temporal** and **monetary** characteristics
- Joins S* with a **subset F** carrying the **maximum flow** from a **laundering source** to the dense subgraph **S***, using a **maximum flow algorithm.**

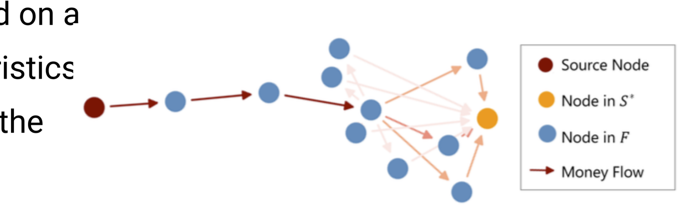

Figure 8 - Subgraph identified by DenseFlow. [7]

SMoTeF [3]

- Extracts **smurf patterns** from a directed multigraph of transactions.
- **Filters** out patterns which are **temporally infeasible** (Fig. 9c)
- Uses the **maximum flow algorithm** to compute the maximum flow of patterns. **Filters** out patterns with **very low flow** (Fig. 9b)
- Outputs a set of smurf patterns with a **high temporal flow** (Fig. 9a)

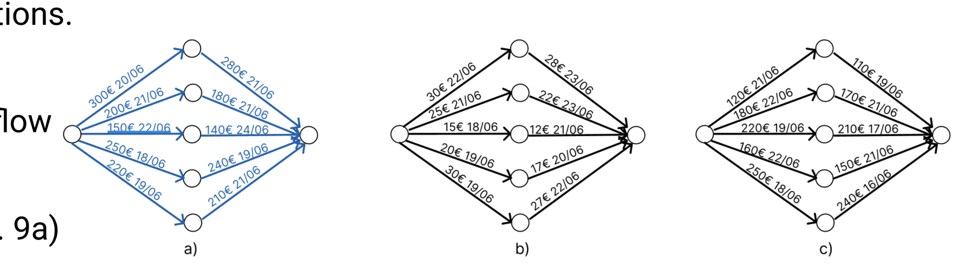

Figure 9 - Smurf patterns filtered out by SMoTeF (b, c)

METHODOLOGY

Evaluation of the algorithms has been performed on **real datasets** with **synthetically injected** patterns, designed to fit the suspiciousness metric of each algorithm. These patterns are **highly unrealistic**.

Therefore, the analysis proves correctness, but not efficiency in a realistic setting.

The experiment aims to test the algorithms on a realistic dataset of transactions in order to observe and understand their limitations.

Dataset

- Dataset chosen is AMLWorld [1] which provides a **synthetic dataset** resembling a virtual world with **multiple interacting entities**, some of which could be money launderers
- **Interactions** between entities in AMLWorld are **complex**, providing **realism** to the dataset.

Evaluation Metric

- To measure the **accuracy** of the algorithms, a **minority class F1 score** is used with a minority in laundering transactions.

Pre-processing

FlowScope

- ***Transaction normalization:*** Convert all payments to EUR)
- ***Parallel edge aggregation***: Sum up amounts of tranactions between the same accounts
- ***Conversion to multipartite format***: Convert directed multigraph (AMLWorld) to a multipartite format

CubeFlow

- ***Time binning:*** Assign time bins to transactions within a predefined time interval (e.g. 24h)
- ***Conversion to coupled tensor format:*** Two approaches: ***1.*** Same as FlowScope for tripartite graph and ***2.*** separate sources (outflow >> inflow) from destinations (inflow >> outflow) and select middle accounts connecting them.

DenseFlow

- ***Time binning (24h)*** and ***Laundering pattern source extraction.***

SMoTeF

- ***Short time binning (1min)***

RESULTS & DISCUSSION

FlowScope

- **F1 = 19.88%** for **full dataset**. **Lowering** the number of **legitimate** transactions **raises** up to **F1** = 45.60% for 10% of legit transactions. This reveals two limitations:

1. **Real laundering chains** are small and dispersed, meaning that they are **overshadowed** by **very large transfers**.
2. The **large** amount of **noise** in the dataset **affects accuracy**, observed in the **increased** F1 with **decreased noise**.

CubeFlow

- **F1 = 0.00%** for both pre-processing strategies. Analysis reveals two limitations:

1. CubeFlow can only analyse **one pair** of coupled tensors, meaning that it will **miss** all patterns **longer than 3**.
2. CubeFlow **expects** transactions to happen in the **same time bin**, however, transactions in **AMLWorld** laundering patterns are **loosely** spread across the total **timeframe**.

DenseFlow

- **F1 = 0.73%**. Additionally, set **F** is **empty** for most laundering patterns in the dataset. Two limitations are identified:

1. Laundering **patterns** are **isolated**, and thus most do **not** have a **flow path** towards a **dense subgraph**.
2. Laundering **patterns** in AMLWorld **do not fit** the **topological** and **temporal** suspiciousness metrics of DenseFlow.

SMoTeF

- **F1 = 19.05%** with **Precision = 1** and **Recall = 0.105**.

1. **Perfect precision** shows that there are **no false positives**, thus the maximum temporal **flow pruning** is **efficient.**
2. **Low recall** is caused by the fact that the algorithm extracts **only Scatter-Gather** patterns (Fig. 1)

CONCLUSION

- This work explores money flow statistics as a solution for money laundering detection.
- It answers the **first research sub-question** by identifying **five algorithms** which use money flow statistics to detect money laundering. The **second research sub-question** is answered by the **analysis of the algorithms**, whose **limitations** open the doors for **further research** regarding:

1. ***Zero-Out*** detection formulas which put more emphasis on low residual rather than ***density***.
2. **Combining** money **flow** statistics with **graph mining** to extract **complex patterns** from multigraphs.
3. **Dynamic time binning** which adapts to laundering **transactions** being spread **loosely** in **time**.

REFERENCES

[1] ALTMAN, E., BLANUŠA, J., VON NIEDERHÄUSERN, L., EGRESSY, B., ANGHEL, A., & ATASU, K. (2023). REALISTIC SYNTHETIC FINANCIAL TRANSACTIONS FOR ANTI-MONEY LAUNDERING MODELS. PROCEEDINGS OF THE 37TH CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NEURIPS 2023), TRACK ON DATASETS AND BENCHMARKS. HTTPS://ARXIV.ORG/ABS/2306.11586
[2] KURSHAN, E., SHEN, H., & YU, H. (2021). FINANCIAL CRIME & FRAUD DETECTION USING GRAPH COMPUTING: APPLICATION CONSIDERATIONS & OUTLOOK. ARXIV PREPRINT ARXIV:2103.01854. RETRIEVED FROM HTTPS://ARXIV.ORG/ABS/2103.01854
[3] SHADROOH, S., & NØRVÅG, K. (2024). SMOTEF: SMURF MONEY LAUNDERING DETECTION USING TEMPORAL ORDER AND FLOW ANALYSIS. APPLIED INTELLIGENCE, 54, 7461–7478. HTTPS://DOI.ORG/10.1007/S10489-024-05545-4
[4] LI, X., LIU, S., LI, Z., HAN, X., SHI, C., HOOI, B., HUANG, H., & CHENG, X. (2020). FLOWSCOPE: SPOTTING MONEY LAUNDERING BASED ON GRAPHS. IN PROCEEDINGS OF THE THIRTY-FOURTH AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI-20) (PP. 4731–4738). AAAI PRESS. HTTPS://GITHUB.COM/APLACEOF/FLOWSCOPE
[5] SUN, X., ZHANG, J., ZHAO, Q., LIU, S., CHEN, J., ZHUANG, R., SHEN, H., & CHENG, X. (2021). CUBEFLOW: MONEY LAUNDERING DETECTION WITH COUPLED TENSORS. ARXIV PREPRINT ARXIV:2103.12411. HTTPS://ARXIV.ORG/ABS/2103.12411
[6] CAPOZZI, A., VILELLA, S., MONCALVO, D., FORNASIERO, M., RICCI, V., RONCHIADIN, S., & RUFFO, G. (2024). WEIRDFLOWS: ANOMALY DETECTION IN FINANCIAL TRANSACTION FLOWS. ITADATA2024: THE 3RD ITALIAN CONFERENCE ON BIG DATA AND DATA SCIENCE. ARXIV:2503.15896. HTTPS://ARXIV.ORG/ABS/2503.15896
[7] DAN LIN, JIAJING WU, YUNMEI YU, QISHUANG FU, ZIBIN ZHENG, AND CHANGLIN YANG. 2024. DENSEFLOW: SPOTTING CRYPTOCURRENCY MONEY LAUNDERING IN ETHEREUM TRANSACTION GRAPHS. IN PROCEEDINGS OF THE ACM WEB CONFERENCE 2024 (WWW '24), MAY 13–17, 2024, SINGAPORE, SINGAPORE. ACM, NEW YORK, NY, USA, 10 PAGES. HTTPS://DOI.ORG/10.1145/3589334.3645692

TU Delft Delft University of Technology