DELATOR: Money Laundering Detection via Multi-Task Learning on Large Transaction Graphs

IEEE BigData 2022

Henrique S. Assumpção[1], Fabrício Souza[1], Leandro Lacerda Campos[1,2], Vinícius T. de Castro Pires[1,2], Paulo M. Laurentys de Almeida[2], Fabricio Murai[1,3]

[1]Computer Science Department - Universidade Federal de Minas Gerais
[2]InterMind – Inter S.A
[3]Worcester Polytechnic Institute

November 8, 2024

What is money laundering?

What is money laundering?

Money laundering is characterized by three main stages:

1. **Placement**: Insert the money into the network and remove obvious traces of illegality.
2. **Layering**: Multiple transactions that aim to conceal the original source of the assets.
3. **Integration**: Integrate the gains into the economy.

We are interested in the **Layering** stage.

How banks implement money laundering detection?

- Money laundering detection varies according to each country, however many still depend mostly on manual methods.
- We will study the procedure that takes place in Brazil.
- Brazilian banks follow the guidelines from BACEN (Brazilian Central Bank).
  - Banks utilize a **rule-based** system.
  - Example: "If a client makes **ten U$800** transactions in a single day to different accounts, then that triggers an **alert**".
  - After the alert the client is considered to be **indicted**, and then can be a potential candidate for manual analysis by the anti-money laundering team (AML). If the suspicion is confirmed, client is reported to the authorities.

How banks implement money laundering detection?

- Money laundering detection varies according to each country, however many still depend mostly on manual methods.
- We will study the procedure that takes place in Brazil.
- Brazilian banks follow the guidelines from BACEN (Brazilian Central Bank).
  - Banks utilize a **rule-based** system.
  - Example: "If a client makes **ten U$800** transactions in a single day to different accounts, then that triggers an **alert**".
  - After the alert the client is considered to be **indicted**, and then can be a potential candidate for manual analysis by the anti-money laundering team (AML). If the suspicion is confirmed, client is reported to the authorities.

Problems with this system:

1. High volume of indicted clients.
2. Rules are manually defined by the AML team, and thus can often be arbitrary or not that the full context provided by the data into account.

What problem are we trying to solve?

- Goal: develop an **automated** and **efficient** method for detecting individuals potentially involved in **money laundering**.
- System must be capable of processing the data of all clients effectively.
  - Measured via quantitative (performance metrics) and qualitative (real experiments) evaluation procedures.
  - Model needs to account for the relations between clients on the network.

What problem are we trying to solve?

- Goal: develop an **automated** and **efficient** method for detecting individuals potentially involved in **money laundering**.
- System must be capable of processing the data of all clients effectively.
  - Measured via quantitative (performance metrics) and qualitative (real experiments) evaluation procedures.
  - Model needs to account for the relations between clients on the network.

Since this is a high stakes problem, we **don't** want the system to report clients in an entirely automated manner, i.e., we want the system to be a CAAT (*Computer-assisted audit technology*).

Data: Introduction

The project was developed in a joint initiative with **InterMind**, Inter's Artificial Intelligence Laboratory. The available data is comprised of a real transaction database provided by the bank, containing more than **100 million transactions**, **20 million clients**, and hundreds of transaction types, spanning across an entire month.

- Data can be naturally modeled as a **graph**.
- Nodes = Inter clients, financial institutions, clients from other banks, etc.
- Edges = transactions. Each transaction has the following characteristics: (i) it is directed (**directed edge**), (ii) it has a value (**weighted edge**), (iii) it has a type (**heterogeneous graph**), and (iv) it occurs at a given moment in time (**dynamic graph**).

Thus, a natural way to model the network is as a **dynamic, directed, weighted heterogeneous graph**.

Data: Modeling

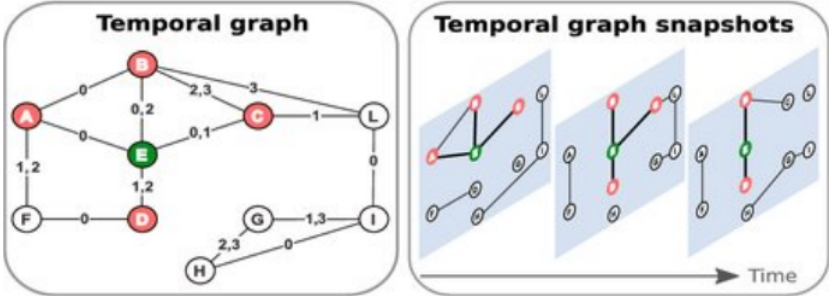
However, models for **dynamic heterogeneous graphs** have a few complications:

1. Significantly costlier to train in comparison to models for homogeneous graphs.
2. Such models are still incipient.

What if we simplify the problem?

E.g. ignore transaction types, i.e., model network as a **dynamic, directed, weighted homogeneous graph**.

To keep the graph simple, we sum the transaction amounts across different types in order to obtain a single scalar for each edge in a given snapshot.



Data: Notable traits

1. We defined snapshots based on weeks due to the cyclic behavior of the data. However, this resulted in a relatively short time series (5 weeks), not suited for sequential models.
2. The target classes are **extremely unbalanced**, i.e., the ratio between the majority class (non-suspicious) and the minority class (suspicious) is considerably small, close to $2 \cdot 10^{-5}$. This has a massive impact on modeling, and thus oversampling is required in order to achieve good results.

Related Works

Here we provide a brief overview of some famous methods available in the literature:

- **Standard Architectures**: Graph Convolutional Network (GCN), GraphSAGE, Graph Attention (GAT), etc.
- **Architectures for dynamic graphs**: EvolveGCN, TGN, EdgeCNN, etc.
- **Architectures for heterogeneous graphs**: DGL-KE (provides five different models), Semi-GNN (built for fraud detection), etc.

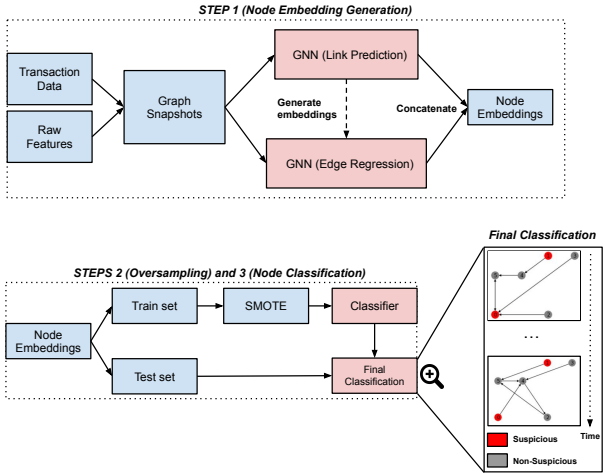
There are few general methods for processing dynamic heterogeneous graphs available in the literature, and even fewer that focus on money laundering detection.

Model: Overview

Classification task: *" Given the tabular and transactional information of a client, what is the probability of him being suspected of participating in money laundering?"*. In order to solve this task, we proposed **DELATOR**.

Model: Overview

Classification task: *" Given the tabular and transactional information of a client, what is the probability of him being suspected of participating in money laundering?"*. In order to solve this task, we proposed **DELATOR**. The method consists of three main steps:



Model: Link Prediction

We train the link prediction task according to the following loss function ($\mathcal{L}^t_{\text{lp}}$, that we wish to minimize):

$$L^t_{\text{pos}}(G^t) = \sum_{((v,u),w)\in E^t} -\log(\sigma((h^t_v)^\top h^t_u)) \tag{1}$$

$$L^t_{\text{neg}}(G^t) = \sum_{((v,u),w)\in \tilde{E}^t} -\log(1-\sigma((h^t_v)^\top h^t_u))) \tag{2}$$

$$\mathcal{L}^t_{\text{lp}} = \frac{L^t_{\text{neg}}(G^t) + L^t_{\text{pos}}(G^t)}{|E^t|+|\widetilde{E}^t|} \tag{3}$$

$h^t_v :=$ link prediction node embedding for node v at snapshot t
$\sigma :=$ sigmoid function
$L^t_{\text{pos}}(G^t) :=$ log-likelihood of the existing links in the current snapshot
$L^t_{\text{neg}}(G^t) :=$ log-likelihood of the existing links in negative samples in the current snapshot

Model: Edge Regression

We train the edge regression task according to the smooth L1 Loss ($\mathcal{L}_{\text{er}}^t$, that we wish to minimize):

$$l_{\text{er}}^t(w, \hat{w}) = \begin{cases} \frac{0.5 \cdot (\hat{w} - w)^2}{\gamma}, & \text{if } |\hat{w} - w| < \gamma \\ |\hat{w} - w| - 0.5 \cdot \gamma, & \text{otherwise} \end{cases}$$
$$\mathcal{L}_{\text{er}}^t = \frac{1}{|E^t|} \cdot \sum_{e=((v,u),w) \in E^t} l_{\text{er}}^t(w, \hat{w}(e)) \tag{4}$$

Where the target w is the edge value, and $\hat{w}$ is the predicted edge value for $e = vu$ obtained according to:

$$\hat{w} = \text{MLP}(h_v^t \parallel z_v^t \parallel h_u^t \parallel z_u^t) \tag{5}$$

$h_v^t :=$ link prediction embedding
$z_v^t :=$ edge regression embedding

Model: Oversampling

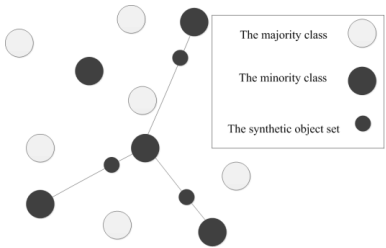
A single time-aware embedding $\eta(v)$ for a node v is obtained as follows:

$$\eta(v) = \|_{t=1}^{T}(h_v^t \parallel z_v^t). \tag{6}$$

SMOTE then oversample's the data as follows:

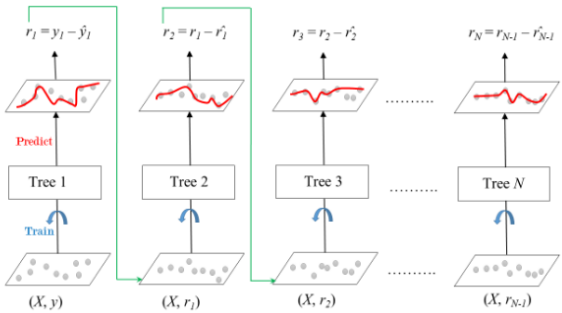
$$\eta(v') = (1 - \lambda) \cdot \eta(v) + \lambda \cdot \eta(\text{nn}(v)) \tag{7}$$

$\text{nn}(v) :=$ nearest neighbor of v from the same class
$\lambda \sim U(0, 1) :=$ uniform random variable



Model: Node classification

The final classification step utilizes **LightGBM**. We train the boosting model on the oversampled embedding set according to the available labels in order to obtain the respective probabilities of a given client being suspicious.



Proposed experiments

In order to evaluate DELATOR's performance, we consider:

1. **Offline evaluation**: quantitative experiments based on classic evaluation metrics.
2. **Online evaluation**: real world experiment conducted with the AML team.

Proposed experiments

In order to evaluate DELATOR's performance, we consider:

1. **Offline evaluation**: quantitative experiments based on classic evaluation metrics.
2. **Online evaluation**: real world experiment conducted with the AML team.

Evaluation metrics:

- **F1-fraud:** geometric mean between precision and recall w.r.t. "suspicious" label.
- **F1-fraud (max):** maximum F1-Fraud value when considering all possible thresholds.
- **AUC:** area under the ROC curve (true positive rate vs. false positive rate).
- **AUPR:** area under the precision-recall curve. Is often more informative then AUC in unbalanced scenarios.

Offline evaluation: Baselines

GNN baseline methods for extracting node embeddings from graphs:

- **DGL-KE**: A state-of-the-art framework developed by *Amazon AWS* for learning representations on knowledge graphs, i.e., heterogeneous graphs with multiple edge and node types.
- **GraphSAGE** (**SAGE**): A state-of-the-art GNN architecture that generalizes the original model for learning on graphs, by allowing for multiple different aggregator functions on the message passing step.
- **Graph Attention** (**GAT**): A state-of-the-art GNN architecture that introduces an attention mechanism to the message passing algorithm.
- **Graph Convolution** (**GCN**): The first proposed GNN architecture based on message passing on graphs.

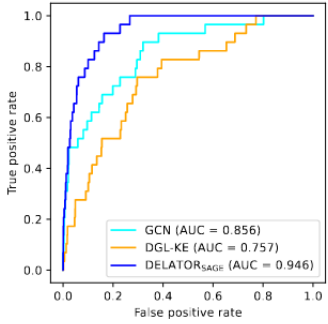
As a baseline that does not utilize the network's transaction information in its modeling, we consider **CatBoost**, a state-of-the-art gradient boosting algorithm for supervised learning, that allows for better performance on highly categorical feature spaces.

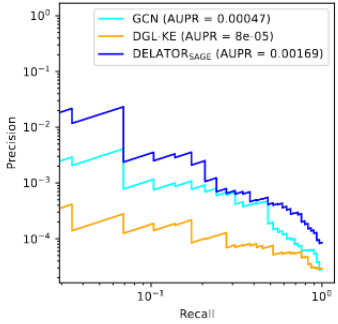
Offline evaluation: Results

Method	Evaluation Metrics			
	AUC-ROC ↑	*AUPR* ↑	*F1-Fraud* ↑	*Max. F1-Fraud* ↑
CatBoost	0.653 ± 0.075	0.007 ± 0.015	0 ± 0	0.009 ± 0.003
DGL-KE	0.735 ± 0.028	0 ± 0	0 ± 0	0.001 ± 0
SAGE	0.846 ± 0.039	0.001 ± 0	0.002 ± 0	0.019 ± 0.017
GAT	0.743 ± 0.043	0.001 ± 0.001	0.001 ± 0.001	0.025 ± 0.019
GCN	0.876 ± 0.018	0.002 ± 0.002	0.002 ± 0	0.032 ± 0.025
$\text{DELATOR}_{\text{SAGE}}$ (Ours)	$\mathbf{0.905} \pm 0.027$	0.001 ± 0	$\mathbf{0.005} \pm 0.001$	0.033 ± 0.010
$\text{DELATOR}_{\text{GCN}}$ (Ours)	0.889 ± 0.014	$\mathbf{0.011} \pm 0.019$	0.003 ± 0.001	$\mathbf{0.049} \pm 0.043$
$\text{DELATOR}_{\text{GAT}}$ (Ours)	0.882 ± 0.023	0.001 ± 0	0.003 ± 0.001	0.018 ± 0.015

Offline evaluation: Results

Method	Evaluation Metrics			
	AUC-ROC ↑	*AUPR* ↑	*F1-Fraud* ↑	*Max. F1-Fraud* ↑
CatBoost	0.653 ± 0.075	0.007 ± 0.015	0 ± 0	0.009 ± 0.003
DGL-KE	0.735 ± 0.028	0 ± 0	0 ± 0	0.001 ± 0
SAGE	0.846 ± 0.039	0.001 ± 0	0.002 ± 0	0.019 ± 0.017
GAT	0.743 ± 0.043	0.001 ± 0.001	0.001 ± 0.001	0.025 ± 0.019
GCN	0.876 ± 0.018	0.002 ± 0.002	0.002 ± 0	0.032 ± 0.025
$\text{DELATOR}_{\text{SAGE}}$ (Ours)	$\mathbf{0.905} \pm 0.027$	0.001 ± 0	$\mathbf{0.005} \pm 0.001$	0.033 ± 0.010
$\text{DELATOR}_{\text{GCN}}$ (Ours)	0.889 ± 0.014	$\mathbf{0.011} \pm 0.019$	0.003 ± 0.001	$\mathbf{0.049} \pm 0.043$
$\text{DELATOR}_{\text{GAT}}$ (Ours)	0.882 ± 0.023	0.001 ± 0	0.003 ± 0.001	0.018 ± 0.015





Online evaluation

We decided to test DELATOR's predictions on a real-world experiment conducted with the help of Inter's AML team.

1. We select the top 50 most suspicious clients according to the model **that did not trigger any of the AML rules**, i.e., that never would have been taken to manual analysis.
2. After analyzing such clients, the AML team found **7** new cases of clients suspected of money laundering, that were then reported to the authorities.

Conclusion

From the presented results, we arrived at the following conclusions:

- DELATOR provides a simple & efficient approach for detecting money laundering, outperforming all the considered baselines. Moreover, the model allowed for the **detection of individuals that would have never been analyzed under the current AML system.**
- Despite the problem's underlying complexity, it is possible to improbe on the existing methods by leveraging different aspects of the data, while following a simpler modeling strategy. The framework can also be **easily reproduced by other banks and financial institutions** interested in a data-driven approach for money laundering detection.

Thank you!

email: henriquesoares@dcc.ufmg.br