ECONOMETRICS
EKONOMETRIA
Advances in Applied Data Analysis

Econometrics. Ekonometria.
Advances in Applied Data Analysis
Year 2024, Vol. 28, No. 2
ISSN 2449-9994
journals.ue.wroc.pl/eada

An Analysis of Novel Money Laundering Data Using Heterogeneous Graph Isomorphism Networks. FinCEN Files Case Study

Filip Wójcik

Wroclaw University of Economics and Business, Poland

e-mail: filip.wojcik@ue.wroc.pl

ORCID: 0000-0001-5938-7260

Quote as: Wójcik, F. (2024). An Analysis of Novel Money Laundering Data Using Heterogeneous Graph Isomorphism Networks. FinCEN Files Case Study. *Econometrics. Ekonometria. Advances in Applied Data Analysis*, *28*(2), 32-49.

DOI: 10.15611/eada.2024.2.03

JEL: G21, C44, C45

Abstract

Aim: This study aimed to develop and apply the novel HexGIN (Heterogeneous extension for Graph Isomorphism Network) model to the FinCEN Files case data and compare its performance with existing solutions, such as the SAGE-based graph neural network and Multi-Layer Perceptron (MLP), to demonstrate its potential advantages in the field of anti-money laundering systems (AML).

Methodology: The research employed the FinCEN Files case data to develop and apply the HexGIN model in a beneficiary prediction task for a suspicious transactions graph. The model's performance was compared with the existing solutions in a series of cross-validation experiments.

Results: The experimental results on the cross-validation data and test dataset indicate the potential advantages of HexGIN over the existing solutions, such as MLP and Graph SAGE. The proposed model outperformed other algorithms in terms of F1 score, precision, and ROC AUC in both training and testing phases.

Implications and recommendations: The findings demonstrate the potential of heterogeneous graph neural networks and their highly expressive architectures, such as GIN, in AML. Further research is needed, in particular to combine the proposed model with other existing algorithms and test the solution on various money-laundering datasets.

Originality/value: Unlike many AML studies that rely on synthetic or undisclosed data sources, this research was based on a publicly available, real, heterogeneous transaction dataset, being part of a larger investigation. The results indicate a promising direction for the development of modern hybrid

AML tools for analysing suspicious transactions, based on heterogeneous graph networks capable of handling various types of entities and their connections.

Keywords: money laundering, deep learning, machine learning, network analysis, graphs

1. Introduction

The practice of concealing the source of funds obtained from illegal activities, such as drug trafficking, corruption, embezzlement and gambling, by converting it into a legitimate source, is commonly referred to as money laundering (Korejo et al., 2021). This type of criminal activity has significant negative effects on the economy and society, as it damages financial sector institutions critical for economic growth, promotes crime and corruption and undermines legitimate private-sector efforts (Ofoeda et al., 2022). One notable and recent example of money laundering was the FinCEN Files investigation, which exposed the role of global banks in the industrial-scale process of money laundering (Fitzgibbon & Hallman Ben, 2020; Lopez-de-Silanes et al., 2022). The investigation was conducted by the International Consortium of Investigative Journalists (ICIJ) and BuzzFeed News and involved more than 400 journalists from 110 news agencies in 88 countries (D'avino, 2023; Fitzgibbon & Hallman, 2020; Shiel & Starkman, 2020). The leaked documents include more than 2,100 suspicious activity reports (SARs) filed and sent by global banks to the USA Treasury Department's intelligence unit, the Financial Crimes Enforcement Network (FinCEN). The involved banks moved more than $2 trillion in payments that were supposed to be suspicious, over the period of 18 years. The data utilised by journalists, analytical reports and summaries were made freely accessible online by ICIJ, thereby enabling other researchers to pursue additional clues and conduct a retrospective analysis of the material (Fitzgibbon & Hallman Ben, 2020). The FinCEN Files investigation was honoured as a finalist for the 2021 Pulitzer Prize in International Reporting (Hudson, 2020; *Megha Rajagopalan, Alison Killing and Christo Buschek of BuzzFeed News – The Pulitzer Prizes*, 2021).

This case was also acknowledged as a significant advancement in the field of investigative journalism, owing to the application of modern analytical tools to uncover connections and correlations among the leaked papers (Haenens et al., 2022). One of these technologies was Neo4j, a graph database, which made it possible to efficiently query and analyse complicated relationships (Hunger et al., 2020).

Over the years, various techniques have been developed for the automatic and intelligent detection of money laundering (Anti Money Laundering, AML) (Tundis et al., 2021). The systems encompass a range of methodologies including machine learning and deep learning (DL) (Alotibi et al., 2022).

Traditional machine learning methodologies, encompassing simpler models such as regression variants, decision trees, ensemble methods including boosting trees, support vector machines (SVMs), rule-based models, and Bayesian inference techniques (Chauhan & Singh, 2019; Kamath et al., 2018; Suyal et al., 2021), are recognised for their satisfactory performance. Nonetheless, these approaches exhibit inherent limitations, particularly their inability to directly leverage the structural relations between edges and vertices, and the lack of robustness against the local density or sparsity within transaction networks (Alarab et al., 2020; Silva et al., 2023; Weber et al., 2018). At the same time, Graph Neural Networks (GNNs) are naturally capable of processing such data, while preserving all the remaining capabilities of DL techniques (Weber et al., 2018).

The presented study makes a two-fold contribution to the field of money laundering detection using graph neural networks. Firstly, it aimed to develop and apply a novel extension of Graph Isomorphism Networks (GIN) called HexGIN (Heterogeneous extension for Graph Isomorphism Network) to the publicly available FinCEN Files case data. The proposed approach leverages the flexibility and expressiveness of GIN (Xu et al., 2018) to handle varying entity types and dimensionality. Its performance is compared with established solutions such as SAGE-based GNN (Hamilton et al., 2017) and Multi-Layer Perceptron (MLP). By demonstrating the potential advantages of HexGIN, this paper

contributes to the development of more effective AML systems. The heterogeneous application of very flexible and expressive Graph Isomorphism Networks (GIN) for such tasks has not yet been extensively explored.

Secondly, unlike previous research (Johannessen & Jullum, 2023; Song & Gu, 2023) that relied on different assumptions, such as edge-feature utilisation, and extends other GNN architectures, the presented approach focuses on developing a heterogeneous GIN that can effectively handle graph-structured data. Moreover, this study employed the publicly available, already graph-enhanced FinCEN files investigation data, which is not subject to legal restrictions or synthetic data limitations. This allows for a more realistic evaluation of the proposed solution and its comparison to existing algorithms.

The research hypothesis is that the more flexible architecture of HexGIN will outperform traditional machine learning models such as MLP, which operate on flattened tabular data, and less expressive GNN models, e.g. heterogeneous Graph SAGE.

The remainder of this paper is organized as follows:

1. Literature Review, is divided into two parts: an analysis of anti-money laundering tools and a review of the relevant graph neural networks literature, providing a comprehensive background for the study.
2. Methodology, details the preparation and structure of the FinCEN Files data, followed by an in-depth explanation of the HexGIN model mechanics, outlining the methodology employed in this research.
3. Results, presents a comparative study between HexGIN, SAGE, and MLP, applied to the FinCEN case files, highlighting the performance and potential advantages of the proposed HexGIN model.
4. Discussion and Conclusions, summarises the key findings, discusses the implications of the research, and suggests potential avenues for future work in the field of money laundering detection using graph neural networks.

2. Literature Review

2.1. Money Laundering

Money laundering is a broad term that lacks a single international definition, as interpretations vary among jurisdictions and organizations. The concept, however, has been well-known since its colloquial mentions during the US Watergate scandal (Korejo et al., 2021). Despite the absence of a universally accepted definition, money laundering consistently exhibits several key elements across various legal acts. These shared elements are discernible in definitions provided by the following institutions: the European Union Parliament and Council Directive 2015/849, which establishes a legal framework to combat money laundering and terrorist financing (The European Parliament, 2015); the American Anti--Money Laundering Act (AMLA) of 2020, a US law that penalises money laundering activities and strengthens regulatory oversight (Thornberry, 2021); the United Nations Convention Against Corruption, an international treaty that establishes criminal offences related to corruption, including money laundering (UN General Assembly, 2003); the International Financial Action Task Force (FATF), an intergovernmental organization that issues recommendations and standards for combating money laundering and terrorist financing (FATF, 2022).

As defined by the aforementioned institutions, the key elements of money laundering can be summarised as follows:

1. Criminal origin of funds: money laundering pertains to the proceeds derived from criminal activities.
2. Conversion or transfer: the process entails converting or transferring funds to render them ostensibly legitimate.

3. Concealment or disguise: the primary objective is to obfuscate or camouflage the source, ownership, or control of the illicit funds.
4. Utilisation of financial systems: money laundering typically capitalises on financial system mechanisms to accomplish its objectives.

2.2. AML Systems

As the global financial system becomes more intricate, the need for advanced anti-money laundering (AML) measures has grown (Han et al., 2020). With the advancement of machine learning algorithms, numerous approaches have been proposed over the years, although a detailed presentation of these methods is beyond the scope of this study. A systematic literature review of over a hundred papers published by American and Chinese researchers between 1994 and 2014 outlined the taxonomy of typical AML approaches, including rule-based or risk-based policies, link analysis, outlier detection, risk classification, and graph analytics (Han et al., 2020).

Rule-based and risk-based approaches, while easy to understand and implement, lack flexibility and adaptability (Han et al., 2020). Association rule mining algorithms can help address these limitations (Drezewski et al., 2016). Risk classification and outlier detection (considered to be typical machine learning tasks) employ machine learning techniques such as Support Vector Machines, Random Forest, Decision Trees, Bayesian approaches, Gradient Boosting, and regression (Alsuwailem & Saudagar, 2020; Tundis et al., 2021). Link and graph analytics involve the analysis of data in a graph format, using classic graph topology and structure analysis (Dumitrescu et al., 2022), and hybrid graph machine learning pipelines (Eddin et al., 2021).

Graph Neural Networks (GNNs) are a class of deep learning models designed to handle graph data, offering advantages such as the utilisation of relations between vertices, robustness to irregular structures, and processing large-scale sparse structures (Wu et al., 2021). The Graph Convolution Network (GCN) architecture (Kipf & Welling, 2017) laid the foundation for the widely used nowadays, message-passing procedure (MPNN) (Gilmer et al., 2017) and its later variations. A meta-analysis of papers on GNNs (Wu et al., 2021) identified over forty such variants, with notable models including Graph Attention Network (GAT) (Velickovic et al., 2017), GraphSAGE (Hamilton et al., 2017), and Graph Isomorphism Network (GIN) (Xu et al., 2018). The latter was theoretically proven to be expressive enough to reflect the Weisfeiler-Lehman graph isomorphism test (Shervashidze et al., 2011) through the application of nested MLP networks.

Recent developments have focused on heterogeneous graphs, which consist of various node types and relations (Shi, 2022). Such networks can model numerous real-world connections and are used to represent knowledge bases (Paulheim, 2017). A systematic survey (Yang et al., 2020) presented over thirty variants of heterogeneous architectures, differing in the way of aggregation and message passing details. Incorporating heterogeneous relations while maintaining stability and generalisation capabilities is considered to be a critical research aspect in this area (Shi, 2022; Wu et al., 2021).

GNNs have been applied to money laundering detection in cryptocurrency trading, using convolutional GNN architectures (Alarab et al., 2020) or self-supervision mechanisms (Lo et al., 2023). Comparisons between classic machine learning methods, GNN variants, and hybrid approaches show that the latter performs better in terms of prediction accuracy (Silva et al., 2023). Recent studies aimed to extend sophisticated GNN architectures to handle heterogeneous data in AML detection problems (Johannessen & Jullum, 2023). However, unlike the research presented in this paper, these latest studies are either based on homogeneous data sources (Alarab et al., 2020) – without varying the types of nodes and relations, heterogeneous datasets undisclosed due to legal reasons (Johannessen & Jullum, 2023), or the homogeneous datasets generated using a simulator (Silva et al., 2023).

3. Methodology

3.1. The Data

The FinCEN case files, published by the International Consortium of Investigative Journalists (ICIJ), consist of transactions log dataset and (partially anonymised) PDF files with details. These files describe Suspicious Activity Reports (SARs) submitted by banks (Díaz-Struck & Armendariz, n.d.). During the investigation, the records were transformed into a graph format. The logical schema of the graph can be illustrated as shown in Figure 1 with the nodes of various colors representing graph vertices, while labelled edges represent various types of relations. These elements combined constitute the heterogeneous transactions graph.

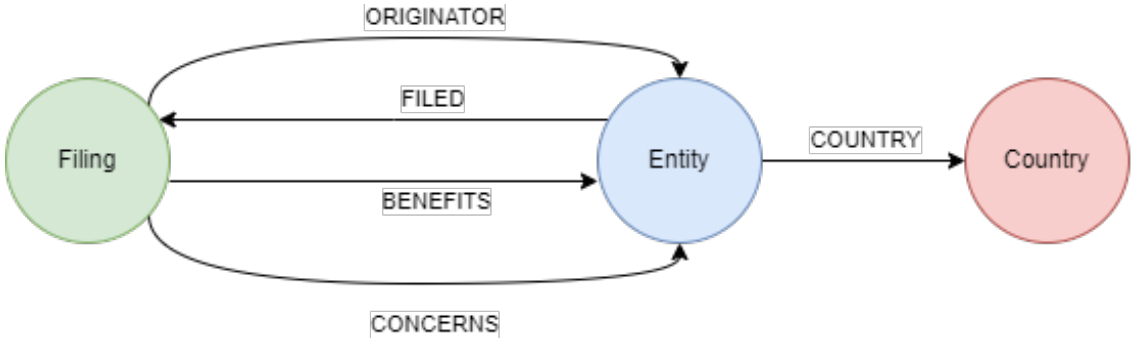

Fig. 1. Logical schema of the graph

Source: own work.

A Filing represents a money transfer initiated by an Entity, such as a bank or financial institution, mentioned by name in the files. Another Entity fills the transfer records and sends them. Multiple parties, including banks and institutions, can potentially be involved in processing the transfer, constituting the “concerns” relationship. The final Entity, that receives the transfer, is its beneficiary. Each Entity is characterised by its name, international code, and the country in which it is registered. The graph data structure described above was used directly as the input for training models.

The dataset contains suspicious activity records (SAR) that are yet to be verified by financial institutions, therefore it does not contain fraudulent/non-fraudulent labels. The research problem and task for the model was therefore formulated as the beneficiary prediction task – the goal of the model was be to learn the money flow patterns, and to correctly recognise the most probable money receiver given some part of the structure (e.g. the originator and some concerned parties).

The FinCEN case files dataset contains 2742 unique entities, 4507 unique filings and 131 countries, with 4501 filing to beneficiary relationships and 24491 filing-concerns-entity relationships. Entities, filings, and names of countries were converted into a unique ID, used by the model to learn numerical representation (embedding) – e.g. “HSBC bank” converted to id = 3.

To effectively capture the inherent complexity of the data, it was structured as a heterogeneous graph. This representation employs adjacency matrices to illustrate the relations in a two-dimensional form, differentiating between source and target entities for each edge type, alongside matrices that detail the attributes of these connections. Figure 2 below provides an illustration of the graph's symbolic representation, as inputted into the model:

```
HeteroData(
  # Features / attributes matrix
  # Dim: (N x D)
  entity={ x=[2823, 2] },
  filing={ x=[4507] },

  # Adjacency matrices per relation type
  # Dim: (2 x |E|)
  (entity, sends, filing)={ edge_index=[2, 4507] },
  (filing, benefits, entity)={ edge_index=[2, 1894] },
  (filing, concerns, entity)={ edge_index=[2, 24491] }
)
```

Fig. 2. Symbolic heterogeneous graph representation in the model

Source: own work.

with $X \in \mathbb{R}^{N \times d}$ being matrices for each vertex type (entity, filing) capturing their attributes (N – number of vertices of given type, d – dimensionality of attributes for that entity) and edge_index $\in \mathbb{N}^{2 \times |E|}$ is the adjacency matrix for each relation type where $|E|$ is the number of connections in that type of relation.

Following the methodology outlined in (Hamilton, 2020; Hamilton et al., 2017), the dataset was divided into disjoint subgraphs with a split ratio for training/validation at 75 percent and testing at 25 percent. The process begins with the random selection of initial nodes to form a connected subgraph followed by the random allocation of edges across different datasets, namely training, validation, and testing. The training set encompasses a subset of edges, establishing the foundational framework crucial for message passing across all phases. The validation set includes edges for use during both validation (for quality checks) and testing (message passing), in addition to training edges. These validation edges are vital for model tuning and initial quality assessments but are excluded from message passing during the training phase to ensure data integrity and prevent information leakage. The testing set contains separate edges designated for final model evaluation post-training and validation phases, ensuring a rigorous assessment of the model's performance on unseen data.

Figure 3 visualises this concept, showing nodes A, B, C, D, E, F in a connected subgraph. The solid black lines represent the training edges, the long-dashed red lines denote the validation edges, and the short-dashed blue lines indicate the testing edges.

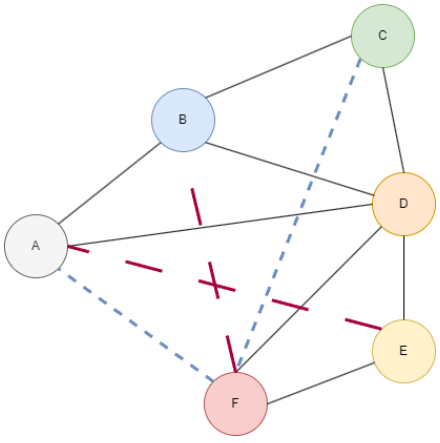

Fig. 3. Edge split example – a complete graph

Source: own work.

Figure 4 shows the evolution of the split and adding edges during subsequent phases.

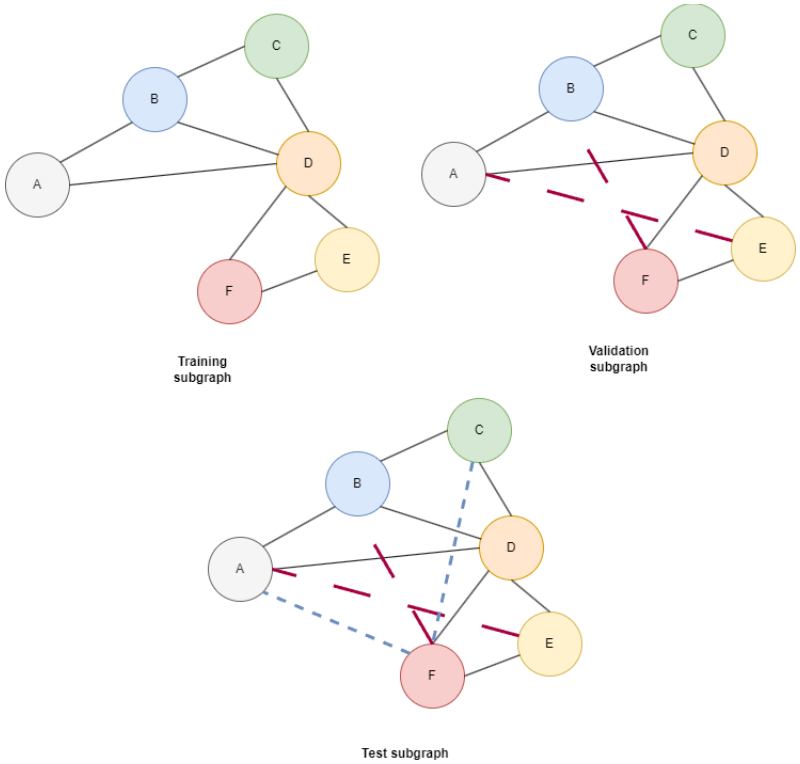

Fig. 4. Train-validation-test split

Source: own work.

To accommodate the heterogeneous nature of the FinCEN files graph, which features diverse types of nodes and edges, a specialised splitting strategy called HGTLoader (Hu et al., 2020) was employed. This technique aims to preserve, as much as possible, the original proportions of node and edge types within each subset, akin to stratified sampling.

3.2. Proposed Model

The model proposed in this study extends the expressive and flexible Graph Isomorphism Network (GIN) architecture, initially designed for homogeneous graphs. The original model was formalised as in equation (1) below (Xu et al., 2018):

$$h_v^{(k)} = MLP^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right), \tag{1}$$

where $k \in \mathbb{N}$ is the current iteration (step) of building numerical representation, MLP is any multi-layer-perceptron neural network, v is the graph vertex for which neural representation is being constructed, $\epsilon \in \mathbb{R}$ is the learnable parameter of weight assigned to node v, $\mathcal{N}(v)$ being the set of nodes in the direct neighbourhood of v (connected to it), and $h_v^{(k-1)}, h_u^{(k-1)} \in \mathbb{R}^d$, are d-dimensional embeddings (vectors of representation) for nodes v, u from previous iteration. Such a formulation assumes, that dimensionality of $h_v^{(k-1)}$ and aggregated neighbouring nodes $\sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}$ must be the same and equal to *d*, so the summation can be performed, therefore the GIN model is applicable to homogeneous graphs only. Visually, equation (1) can be depicted as in Figure 2.

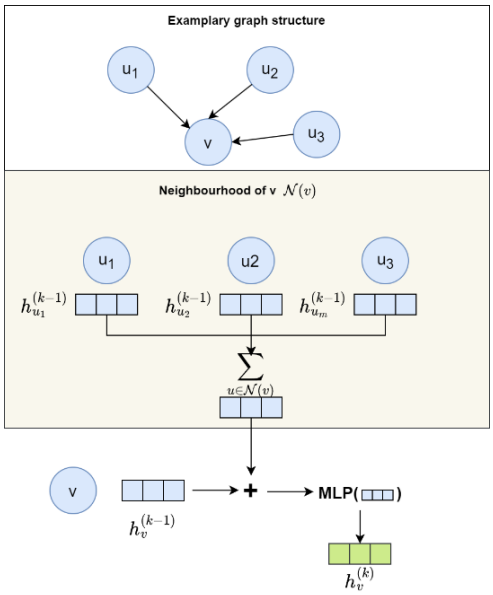

Fig. 5. The process of obtaining node representation in the GIN network

Source: own work.

Each coloured set of squares represents a vector of given dimensionality for a specific node in the network. A change of colour indicates a change of mapping – the result of a specific function application. The vectors for nodes u_1, u_2, u_3 have the same dimensionality, so the summation is possible.

The proposed HexGIN model modifies the equation for the node embedding function to incorporate two factors: heterogeneity of relations (neighbours of node v can be connected via different link types) and different dimensionality of the neighbours' embeddings. The updated formulation of equation (1) combines elements from R-GCN networks (Schlichtkrull et al., 2017), which defined the notion of the aggregation of heterogeneous relations, with the concept of mappings concatenation (Z. Hu et al., 2020).

A heterogeneous graph is defined as $\mathcal{G} = \{\mathrm{V}, \mathrm{E}\}$, where V is a set of vertices and E is a set of edges, where additionally each vertex $v \in V$ and edge $e \in E$ are associated with specific mapping functions: $\phi(v)$: $V \rightarrow \mathcal{A}$, $\varphi(e)$: $E \rightarrow \mathcal{R}$, where $\mathcal{R}$ is a set of relations in the graph, and $\mathcal{A}$ is a set of allowed node types (Shi, 2022). For such graphs, the HexGIN model can be formalised using a message passing framework (Gilmer et al., 2017; Hamilton, 2020), with a set of three equations presented below.

Given a node *v*, $R(v) \in \mathcal{R}$ – a set of relations, connecting v to its neighbours, with $r \in R(v)$ being a concrete relation, $N_r(v)$ – a set of v's neighbours within that relation, $h_{u,r}^{(k-1)} \in \mathbb{R}^d$ – neighbor's u embedding from previous iteration, and $h_v^{(k-1)} \in \mathbb{R}^t$ previous t-dimensional embedding for node *v*, thus one can define the following equations, constituting HexGIN model: (2) defines a message (numerical vector) sent by v's neighbours, (3) aggregates incoming neighbour messages, and (4) formulates new representation of *v*.

$$m_{u,r\,\in\mathcal{N}_r(v)}^{(k)} = \Phi_r^{(k)}\left(h_{u,r}^{(k-1)}\right), \tag{2}$$

$$m_v^{(k)} = ||_{r\in R(v)}\, AGG_{u\in\mathcal{N}_r(v)}\left(m_{u,r}^{(k)}\right), \tag{3}$$

$$h_v^{(k)} = \Phi_v^{(k)}\left(\left(1+\epsilon^{(k)}\right)h_v^{(k-1)} \,||\, m_v^{(k)}\right), \tag{4}$$

$\Phi_r^{(k)}$: $\mathbb{R}^{d'} \rightarrow \mathbb{R}^d$ is any neural network capable of handling simple matrix of features for mapping nodes connected via relation *r* from their initial dimensionality to a new dimensionality d'. The result of this

operation is each node's u (connected via relation *r*) message $m_{u,r\in\mathcal{N}_r(v)}^{(k)} \in \mathbb{R}^{d'}$. $AGG_{u\in\mathcal{N}_r(v)}: \mathbb{R}^{d'} \rightarrow \mathbb{R}^{d'}$ is any function that aggregates these messages of neighbours within relation r without changing their dimensionality (e.g. summation, averaging, etc.). $||_{r\in R(v)}$ is the vector concatenation operator, that stacks those aggregated embeddings together, resulting in a composite message vector $m_v^{(k)} \in \mathbb{R}^f$ with new dimensionality *f*, being a result of concatenation. $\Phi_v^{(k)}: \mathbb{R}^{(f+t)} \rightarrow \mathbb{R}^{t'}$ is a node-update neural network that maps nodes from (3), concatenated with previous node *v* embedding, to a new dimensionality t'. Graphically, the HexGIN operation can be depicted as in Figure 3.

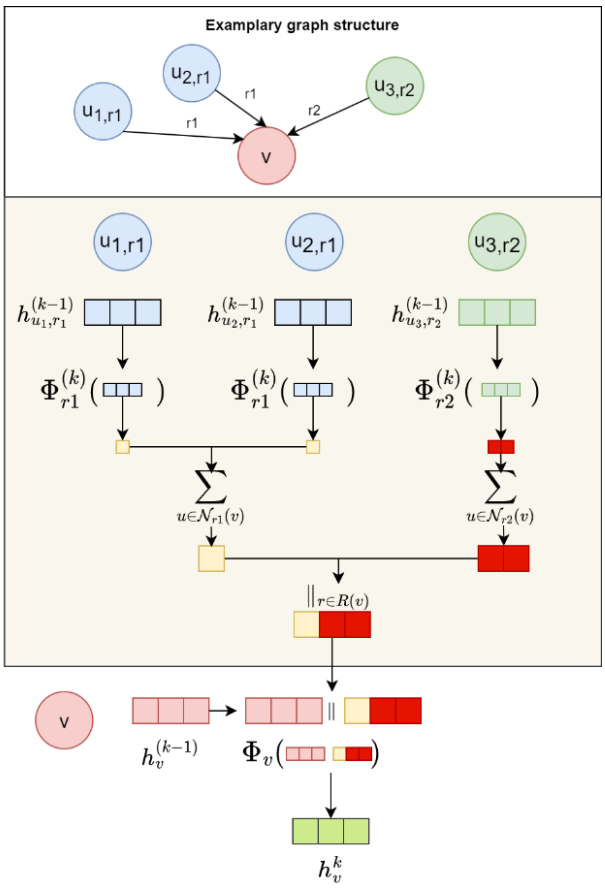

Fig. 6. HexGIN message formulation

Source: own work.

The various node types are indicated by various colours. Coloured squares depict vectors of features of each node type. The blue nodes (relation 1), are mapped by Φ_{r1} into new dimensionality (yellow) and all such remapped nodes are then aggregated using summation. The same applies to the green node (relation 2) transformed by Φ_{r2}.

It can be shown that in a certain setup, HexGIN can be formally equivalent to the original GIN network, sharing its properties. Due to space limitations, this relation is presented in the additional materials.

3.3. Experimental Design

The primary goal of the study was to identify the beneficiaries of each transaction within the FinCEN files case data. To achieve this objective, the experiment was formulated as a link-prediction task (Hamilton, 2020), where the model should predict the existence of the "benefits" relationship between a filing and an entity, given the remaining structure of the graph: the initiator and filer entities, as well as the concerned parties.

The following models were selected for the study: Graph SAGE (Hamilton et al., 2017) graph neural network model, transformed to handle heterogeneous data; a Multi-Layer Perceptron (MLP) neural network featuring a feedforward architecture, beginning with embedding lookup layers for categorical data and followed by simple densely connected layers, as benchmark (Goodfellow et al., 2016); and proposed novel HexGIN. All the models had the same structure for the initial layers: embeddings and preprocessing, differing only in the way of handling message passing in the graphs (in the case of Graph SAGE and HexGIN) or vectors (for MLP). The other architecture hyperparameters were tuned automatically during the inner loop of cross validation, with the goal to maximise the achieved scores. The models in the study made predictions on native graph data, while the MLP operated on 'flattened' information, as it is unable to handle graph structures. For the MLP, each set of elements (e.g. $\{(\mathrm{v}_1, \text{filed}, \text{filing}_1), (\text{filing}_1, \text{concerns}, \mathrm{u}_1)\}$) were concatenated together to form a single vector. Due to space limitations, the models' architecture and implementation details are presented in the appendix.

To create a binary prediction experiment, negative samples (Levy & Goldberg, 2014; Mikolov et al., 2013) were added to the dataset. The number of these samples was equal to positive ones. In this context, the negative samples help with the model learning as to which connections exist and which not. On each dataset (train/validation/test) separately, they are obtained by adding malformed versions of existing connections, constructed by creating non-existing ones. For example, if on training dataset there exists a set of connections $\{(\mathrm{v}_1, \text{filed}, \text{filing}_1), (\text{filing}_1, \text{concerns}, \mathrm{u}_1)\}$, then a negative (non-existing or fictitious) sample could be $\{(\mathrm{v}_1, \text{filed}, \text{filing}_2), (\text{filing}_2, \text{concerns}, \mathrm{u}_1)\}$. Using this technique, the model can better learn to distinguish between positive and negative instances, ultimately enhancing its ability to make accurate predictions as shown in (Herghelegiu & Lu, 2021; Liu et al., 2022; Yang et al., 2023).

In each cross-validation fold, the task of models was to perform embedding of vertices (turn their unique identifiers/names to numerical vectors), indicated as colored, square bars in Figures 2 and 3, considering the whole graph structure (using equations (2), (3), (4)) and then carry out a binary classification of the edges between filings and beneficiaries. Predictions should minimise binary-cross entropy loss function by setting the probability of positive (existing) connection to a high value and reducing the probability of non-existing (fictitious) connections. The formal definition of loss function J is given by equation (5) below – a general binary cross entropy function used in machine learning systems (Ho & Wookey, 2020):

$$J = -\frac{1}{M}\sum_{m=1}^{M}\left[y_m \, log\big(y_\theta(x_m)\big) + (1 - y_m) \, log\big(1 - y_\theta(x_m)\big)\right], \tag{5}$$

where *M* is the number of all relation predictions, y_m is the expected relation label (existing = 1 /fictitious = 0), $y_\theta(\cdot)$ is the HexGIN model prediction given parameters θ and x_m is a graph data input (initial vertex IDs for filing, entity and country, as well as the graph neighbourhood information).

All the models start with random numbers (embeddings) for each unique ID for entities, filings, and relations. Graph SAGE and HexGIN, iteratively, during message passing (equations (2), (3) (4) for HexGIN) update these numerical representations so that they can be used during link prediction. Given two vertices v, u (for example filing 1 and entity 1) and relation r (for example benefits), the models perform a non-linear mapping to 0/1 value indicating whether there should be a link or not, attempting to minimise loss function (5).

Each model was tested using 10 times repeated 10-fold cross-validation (resulting in 100 attempts per model), with random restarts each time and random training graph division into train and validation part, for the better estimation of average predictive quality on unseen data (Bates et al., 2023; Bouthillier et al., 2021; Kohavi, 1995; Raschka, 2018). Such a method is advised for deep learning models, as it allows assessing the two sources of variance and instability of predictions – the data (random division of dataset multiple times) and model (random weights initialisation) (Bouthillier et al.,

2021; Raschka, 2018). On each fold, 30% of the edges were used for supervision in a transduction split (Hamilton et al., 2017).

4. Results

Statistical comparisons of cross-validation results are subject to controversies in statistical and data science literature. Some authors indicated that most tests assume independence of samples (scores), which is clearly violated during such experiments (Bates et al., 2023; Vanwinckelen & Blockeel, 2012; Wong & Yeh, 2020). Other authors suggested that independence assumption violation can be ignored, when proper p-value corrections or if different additional techniques are applied (Demšar, 2006, 2008). However, graph cross-validation requires careful sample selection that does not break the structure of the graph (e.g. introducing artificial, disconnected subgraphs) or leak connections between sets (Leskovec & Faloutsos, 2006; Leskovec & Sosič, 2016), which supports even more strongly skepticism against the statistical tests.

In light of these considerations, the presented study employed a percentile-based bootstrap method for establishing confidence intervals during model comparison (Raschka, 2018). For each model and the corresponding training metric, which includes 100 scores from repeated 10 x 10 cross-validation with varying random seeds per model, percentile-based 95% bootstrap confidence intervals were generated through resampling 1,000 times (Bouthillier et al., 2021; Raschka, 2018). This process yielded a vector of resampled metrics for each model, represented as $\mathrm{Metric}_{\mathrm{model}} \in \mathbb{R}^{1000}$. With significance level $\alpha = 0.05$, the 2.5th percentile $\alpha_{low} = \alpha/2$) and 97.5th ($\alpha_{high} = 1 - \alpha/2$) percentile values within this vector were then determined as the limits of the confidence intervals, denoted as $\{\alpha_{low} \times \mathrm{Metric}_{\mathrm{model}},\ \alpha_{high} \times \mathrm{Metric}_{\mathrm{model}}\}$.This approach, which does not require parametric assumptions, is favoured for its computational efficacy and reliance on model scores derived from varied data samples (Koźniewski et al., 2016; Noma et al., 2021; Raschka, 2018). Table 1 and Figure 4 summarise the mean training metrics scored by each model, their standard deviation (in brackets with +/– sign) and bootstrap 95% confidence intervals (in square brackets – low and high values).

Table 1. Training scores of each model

Model\Metric	F1 score	ROC AUC	Precision	Recall
HexGIN	**0.864** (± 0.04) **[0.855; 0.871]**	**0.921** (± 0.02) **[0.915; 0.926]**	**0.819** (± 0.02) **[0.813; 0.824**]	0.924 (± 0.09) [0.909; 0.939]
MLP	0.828 (± 0.04) [0.817; 0.834]	0.896 (± 0.03) [0.887; 0.901]	0.721 (± 0.05) [0.723; 0.733]	**0.966** (± 0.09) **[0.941; 0.979]**
Graph SAGE	0.849 (± 0.03) [0.84; 0.854]	0.864 (± 0.02) [0.864; 0.874]	0.774 (± 0.02) [0.769; 0.779]	0.946 (± 0.08) [0.928; 0.959]

Source: own work.

The bold font indicates the highest result for a given metric.

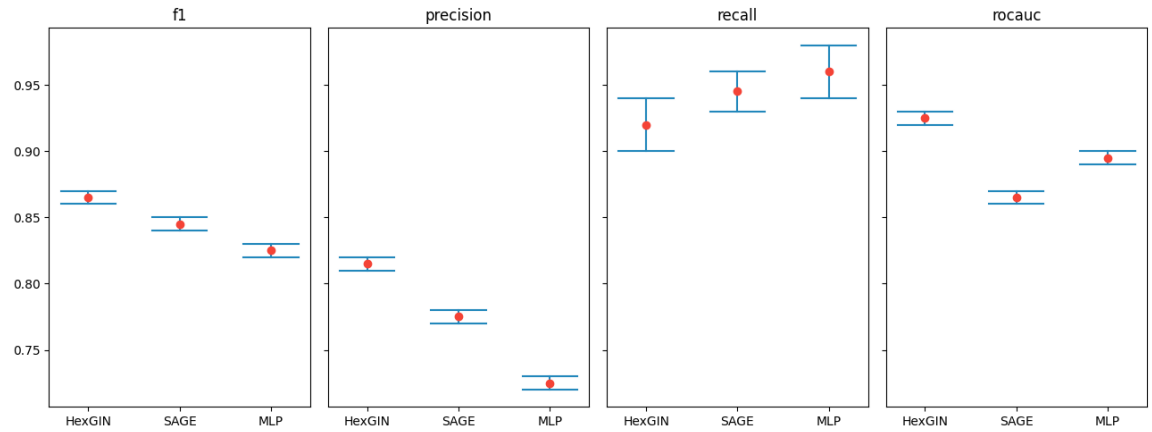

Fig. 7. Confidence intervals for bootstrapped metrics

Source: own work.

Finally, the test scores were obtained by training each model on the whole available training and validation data and predicting beneficiary connections on the test set. The results are presented in the table below.

Table 2. Test scores of models

Model	F1 score	ROC AUC	Precision	Recall
HexGIN	**0.756**	**0.813**	**0.766**	**0.758**
MLP	0.637	0.695	0.659	0.497
Graph SAGE	0.726	0.783	0.741	0.73

Source: own work.

The bold font indicates the highest result for the given metric.

5. Discussion and Conclusions

The experimental results show the superior performance of the proposed HexGIN model compared to the existing solutions, such as MLP and heterogeneous Graph SAGE. During the training phase, HexGIN outperformed (with non-overlapping confidence interval values) the other models in terms of F1 score, precision, and ROC AUC. On the final test set, the proposed solution surpassed other algorithms across all metrics. It should be noted that, despite the use of overfitting-prevention measures such as neurons' dropout, regularisation and early stopping, there was a difference between the overall test results and the training. This can be attributed to the fact that the testing subgraph, separated from training/validation (Hamilton et al., 2017) was smaller and may not exhibit all the same structural properties due to some nodes' extraction. This issue is usually mitigated with the help of predefined public train-test splits in various datasets (Hu et al., 2020), but this was not possible in the study, as the FinCEN files lacked predefined label assignments. Variance of all the studied models was relatively low and comparable, indicating stable and consistent predictions from run to run.

A noteworthy aspect of the new architecture is its ability to utilise the native heterogeneous graph representation and structure of the data while keeping the original GIN formulation, whereas MLP relied on a flattened vectors' representation, converting the graph into a tabular format. This highlights the potential benefits of using graph-based models for tasks involving intricate relations and heterogeneous data structures. This tendency was visible in almost all the metrics, where GraphSAGE, also representing a graph neural networks family of models, outperformed the MLP model.

In relation to the research goals, the results support the hypothesis that the more flexible architecture outperforms the MLP and less expressive GNN architecture. By extending the Graph Isomorphism Network architecture and incorporating heterogeneous relations handling, HexGIN offers a flexible and expressive solution for analysing graph-enhanced data.

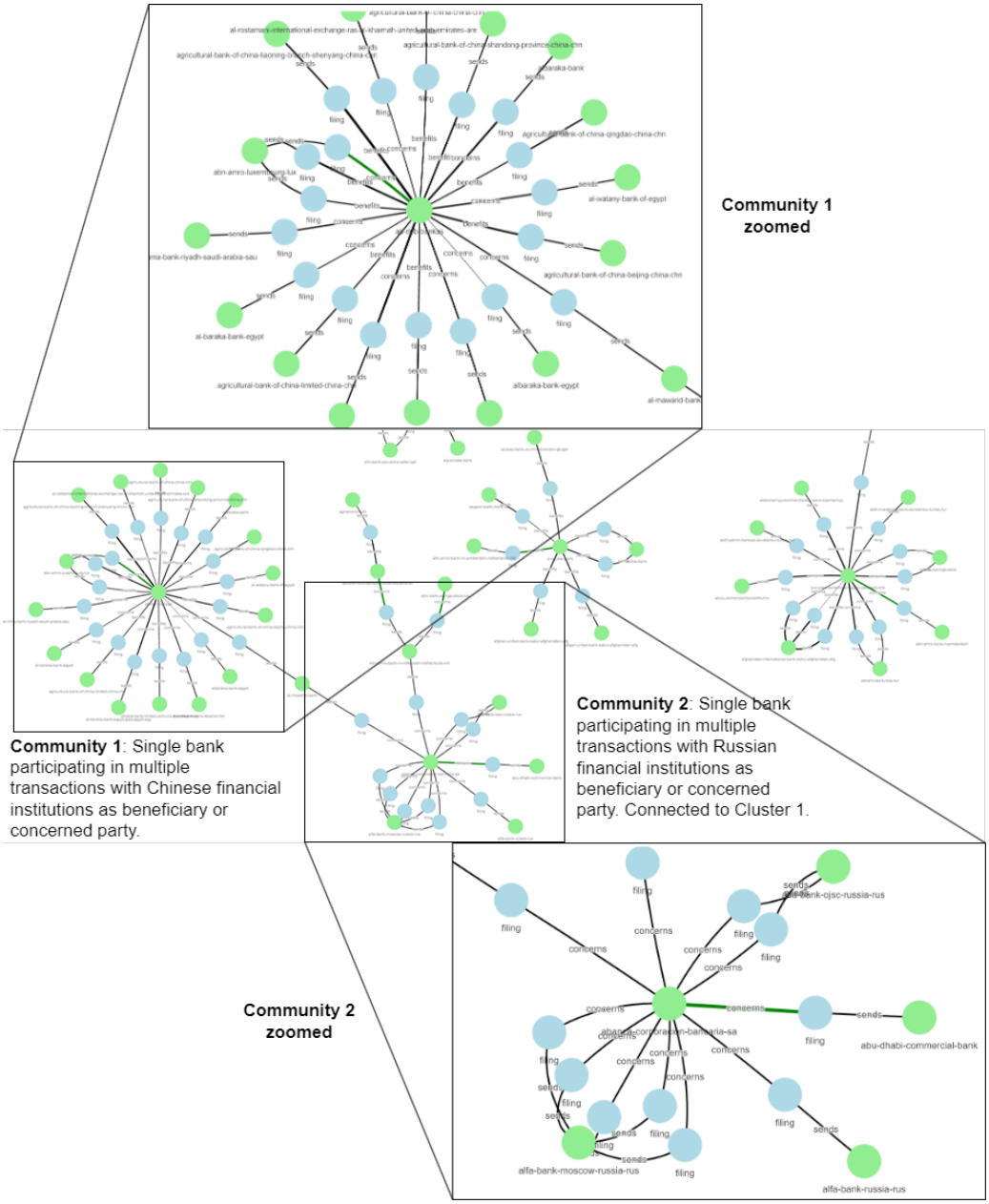

Fig. 8. Model predictions analysis

Source: own work.

Apart from the raw training/validation/testing scores analysis, the HexGIN results can be interpreted in terms of the graph structure importance and patterns. The technique referred to as "gradient-attribution" introduced by Meta's CAPTUM (Kokhlikyan et al., 2020) was used to explain model test predictions. The goal was to search for graph edges (transactions) that have the strongest weight of the evidence, helpful in making the correct predictions. CAPTUM iteratively changes the graph structure by removing certain edges and evaluating the model loss and gradients (Hao, 2023). Gradient change indicates the transaction's significance for correct prediction. Figure 8, based on part of the test data (only a subset for presentation clarity), presents the simplified results of that process.

1. Green nodes represent banks.
2. Blue nodes represent filings (money transfers).
3. Edge thickness corresponds to the gradient importance.
4. Edge colour represents the prediction results: black – not a subject of prediction only message passing, green – correct classification, red – incorrect classification.

The central part of the figure (zoomed out) presents the overview of the subgraph, with nodes forming clear sub-clusters connected via thick (therefore significant for prediction) money transfers. Two regions were enlarged. On the left-hand side: a large cluster centred around the Lithuanian bank, participating as a beneficiary or concerned party in multiple transactions with Chinese financial institutions. A single, thick transfer in the middle connects it with the Spanish bank, participating in multiple transactions with Russian institutions. Due to their data volume and complexity, such visualisations should be investigated by experts using interactive tools, published together with this work. A close inspection of the formed clusters and their weight of evidence can provide very useful information for investigators, especially if combined with another set of algorithms such as weighted community detection (Liu et al., 2020).

This study is part of broader research on the use of heterogeneous graph neural networks as tools supporting decision-making. Incorporating heterogeneous relations into GNNs while deepening the network without sacrificing stability and generalisation capabilities is considered one of the most critical research aspects in this area (Shi, 2022; Wu et al., 2021). The results of this study, as well as others focused on building AML tools with graph neural networks, confirm that such a combination could lead to the development of a new class of accurate, modern systems.

The experiment was limited to a single, publicly available dataset on money laundering. Due to the sensitivity of personal data, as well as the confidentiality of legal proceedings, examples of datasets on this topic are limited, and published research papers often cannot disclose source data, either. Many papers rely on synthetic data or homogeneous datasets, in which all the nodes and relations are the same, therefore limiting the possibility to directly compare them. Other heterogeneous datasets (e.g. molecules, citation networks or social media graphs) do not necessarily exhibit similar characteristics as money laundering networks. It would be beneficial to repeat this study on multiple realistic AML databases with various model configurations, to check the performance. Additionally, the inclusion and combination of ideas from other implementations, such as edge-attributes (Johannessen & Jullum, 2023) or variants of GNN transformers (Song & Gu, 2023) could lead to better prediction scores on such datasets.

References

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020). *Competence of Graph Convolutional Networks for Anti-Money Laundering in Bitcoin Blockchain* (Proceedings of the 2020 5th International Conference on Machine Learning Technologies, 23-27). https://doi.org/10.1145/3409073.3409080

Alotibi, J., Almutanni, B., Alsubait, T., Alhakami, H., & Baz, A. (2022). Money Laundering Detection using Machine Learning and Deep Learning. *International Journal of Advanced Computer Science and Applications*, *13*(10), 732-738. https://doi.org/10.14569/IJACSA.2022.0131087

Alsuwailem, A. A. S., & Saudagar, A. K. J. (2020). Anti-money Laundering Systems: A Systematic Literature Review. *Journal of Money Laundering Control*, *23*(4), 833-848. https://doi.org/10.1108/JMLC-02-2020-0018

Bates, S., Hastie, T., & Tibshirani, R. (2023). Cross-Validation: What Does It Estimate and How Well Does It Do It? *Journal of the American Statistical Association*, 1-12. https://doi.org/10.1080/01621459.2023.2197686

Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi Sepahvand, N., Raff, E., Madan, K., Voleti, V., Kahou, S. E., Michalski, V., Arbel, T., Pal, C., Varoquaux, G., & Vincent, P. (2021). Accounting for Variance in Machine Learning Benchmarks. In A. Smola, A. Dimakis, I. Stoica (Eds.), *Proceedings of Machine Learning and Systems* (vol. 3, pp. 747-769). mlsys.org

Chauhan, N. K., & Singh, K. (2019). *A Review on Conventional Machine Learning vs Deep Learning* (2018 International Conference on Computing, Power and Communication Technologies, GUCON 2018). https://doi.org/10.1109/GUCON.2018.8675097

D'avino, C. (2023). Money Laundering and AML Regulatory and Judicial System Regimes: Investigation of FinCEN Files. *European Journal of Law and Economics*, *55*(2), 195-223.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*(Jan), 1-30.

Demšar, J. (2008). On the Appropriateness of Statistical Tests in Machine Learning. *Workshop on Evaluation Methods for Machine Learning in Conjunction with ICML*, 65.

Díaz-Struck, E., & Armendariz, A. (n.d.). *Download FinCEN Files Transaction Data – ICIJ*. ICJI.Org. Retrieved August 7, 2023 from https://www.icij.org/investigations/fincen-files/download-fincen-files-transaction-data/

Drezewski, R., Dziuban, G., Hernik, L., & Paczek, M. (2016). Comparison of Data Mining Techniques for Money Laundering Detection System. *Proceedings – 2015 International Conference on Science in Information Technology: Big Data Spectrum for Future Information Economy, ICSITech 2015*, 5-10. https://doi.org/10.1109/ICSITECH.2015.7407767

Dumitrescu, B., Baltoiu, A., & Budulan, S. (2022). Anomaly Detection in Graphs of Bank Transactions for Anti Money Laundering Applications. *IEEE Access*, *10*, 47699-47714. https://doi.org/10.1109/ACCESS.2022.3170467

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

FATF. (2022). The FATF Recommendations: International Standards on Combating Money Laundering and the Financing of Terrorism & Proliferation. *FATF*, (March). https://www.fatf-gafi.org/en/publications/Fatfrecommendations/Fatf-recommendations.html

Fitzgibbon, W., & Hallman, B. (2020, September 20). *What Is the FinCEN Files Investigation? ICIJ*. ICJI.Org. https://www.icij.org/investigations/fincen-files/what-is-the-fincen-files-investigation/

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. In D. Precup, Y. Whye Teh (Eds.), *34th International Conference on Machine Learning, ICML 2017* (vol. 3, pp. 1263-1272).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Haenens, L., Lo, W. H., & Moore, M. (2022). Innovation in Journalism: How Technology Affects the News Media, Publication Formats, and the Journalist Profession. In J. Trappel, T. Tomaz (Eds.), *Success and Failure in News Media Performance: Comparative Analysis in The Media for Democracy Monitor* (pp. 337-354). Nordicom.

Hamilton, W. L. (2020). Graph Representation Learning Hamilton. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *14*(3), 1-159. https://doi.org/10.2200/S01045ED1V01Y202009AIM046

Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017* (pp. 1024-1034).

Han, J., Huang, Y., Liu, S., & Towey, K. (2020). Artificial Intelligence for Anti-Money Laundering: A Review and Extension. *Digital Finance*, *2*(3-4), 211-239. https://doi.org/10.1007/S42521-020-00023-1

Hao, J. (2023). *Towards Reliable Medical Image Analysis Based on Deep Learning with XAI*. https://doi.org/10.1117/12.2680931

Herghelegiu, A. I., & Lu, H. (2021). *Improving Negative Sampling in Graph Neural Networks for Predicting Drug-Drug Interactions* (Proceedings – 2021 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2021). https://doi.org/10.1109/BIBM52615.2021.9669483

Ho, Y., & Wookey, S. (2020). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2019.2962617

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., & Leskovec, J. (2020). Open Graph Benchmark: Datasets for Machine Learning on Graphs. *Advances in Neural Information Processing Systems*, *33*, 22118-22133.

Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). *Heterogeneous Graph Transformer* (Proceedings of the Web Conference 2020, 2704-2710).

Hudson, M. (2020). *FinCEN Files Investigation Named Pulitzer Prize Finalist – ICIJ*. ICJI.Org. https://www.icij.org/investigations/fincen-files/fincen-files-investigation-named-pulitzer-prize-finalist/

Hunger, M., Lyon, W., & Van Bruggen, R. (2020). *Analyzing the FinCEN Files with Neo4j*. Neo4j. https://neo4j.com/blog/analyzing-fincen-files-data-neo4j/

Johannessen, F., & Jullum, M. (2023). Finding Money Launderers Using Heterogeneous Graph Neural Networks. *ArXiv: 2307.13499*.

Kamath, C. N., Bukhari, S. S., & Dengel, A. (2018). *Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification* (Proceedings of the ACM Symposium on Document Engineering 2018, DocEng 2018). https://doi.org/10.1145/3209280.3209526

Kipf, T. N., & Welling, M. (2017). *Semi-supervised Classification with Graph Convolutional Networks* (5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings, pp. 1-14).

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference of Artificial Intelligence*, *14*(2), 1137-1145.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & others. (2020). Captum: A Unified and Generic Model Interpretability Library for Pytorch. *ArXiv Preprint ArXiv:2009.07896*

Korejo, M. S., Rajamanickam, R., Helmi, M., & Said, M. (2021). The Concept of Money Laundering: A Quest for Legal Definition. *Journal of Money Laundering Control*, *24*(4), 725-736. https://doi.org/10.1108/JMLC-05-2020-0045

Koźniewski, M., Cypko, M. A., & Drużdżel, M. J. (2016). How Reliable Is a Measure of Model Reliability? Bootstrap Confidence Intervals Over Validation Results. *Advances in Computer Science Research*, *13*, 27-41.

Leskovec, J., & Faloutsos, C. (2006). *Sampling from Large Graphs* (Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631-636). https://doi.org/10.1145/1150402.1150479

Leskovec, J., & Sosič, R. (2016). SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Transactions on Intelligent Systems and Technology*, *8*(1), 1-20. https://doi.org/10.1145/2898361

Levy, O., & Goldberg, Y. (2014). Dependency-Based Word Embeddings(*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers*, pp. 302-308). https://doi.org/10.3115/v1/P14-2050

Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., & Yu, P. S. (2020). *Deep Learning for Community Detection: Progress, Challenges and Opportunities* (*IJCAI International Joint Conference on Artificial Intelligence, 2021-January).* https://doi.org/10.24963/ijcai.2020/693

Liu, Y., Zhong, Z., Che, C., & Zhu, Y. (2022). Recommendations with Residual Connections and Negative Sampling Based on Knowledge Graphs. *Knowledge-Based Systems*, *258*. https://doi.org/10.1016/j.knosys.2022.110049

Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., & Portmann, M. (2023). Inspection-L: Self-Supervised GNN Node Embeddings for Money Laundering Detection in Bitcoin. *Applied Intelligence*, *53*(16), 19406-19417. https://doi.org/10.1007/s10489-023-04504-9

Lopez-de-Silanes, F., McCahery, J. A., & Pudschedl, P. C. (2022). Corporate Governance and Value Preservation: The Effect of the FinCEN Leak on Banks. *Law & Contemp. Probs.*, *85*(247).

Megha Rajagopalan, Alison Killing and Christo Buschek of BuzzFeed News – The Pulitzer Prizes. (2021). Pulitzer.Org. https://www.pulitzer.org/winners/megha-rajagopalan-alison-killing-and-christo-buschek-buzzfeed-news

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States* (vol. 1, pp. 3111-3119).

Noma, H., Shinozaki, T., Iba, K., Teramukai, S., & Furukawa, T. A. (2021). Confidence Intervals of Prediction Accuracy Measures for Multivariable Prediction Models Based on the Bootstrap-Based Optimism Correction Methods. *Statistics in Medicine*, *40*(26). https://doi.org/10.1002/sim.9148

Ofoeda, I., Agbloyor, E. K., & Abor, J. Y. (2022). How Do Anti-Money Laundering Systems Affect FDI Flows Across the Globe? *Cogent Economics & Finance*, *10*(1), 1-27. https://doi.org/10.1080/23322039.2022.2058735

Paulheim, H. (2017). Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semantic Web*, *8*(3), 489-508. https://doi.org/10.3233/SW-160218

Raschka, S. (2018). Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning Performance Estimation: Generalization Performance Vs . Model Selection. *ArXiv*, *January*.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2017). Modeling Relational Data with Graph Convolutional Networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10843 LNCS*, 593-607. https://doi.org/10.1007/978-3-319-93417-4_38

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. V., Mehlhorn, K., & Borgwardt, K. (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, *12*(9), 2539-2561. https://doi.org/10.5555/1953048.2078187

Shi, C. (2022). Heterogeneous Graph Neural Networks. In L. Wu, P. Cui, J. Pei, & L. Zhao (Eds.), *Graph Neural Networks: Foundations, Frontiers, and Applications* (pp. 351-369). Springer Nature. https://doi.org/10.1007/978-981-16-6054-2_16

Shiel, F., & Starkman, D. (2020, September 19). *About the FinCEN Files Investigation – ICIJ*. ICJI.Org. https://www.icij.org/investigations/fincen-files/about-the-fincen-files-investigation/

Silva, Í. D. G., Correia, L. H. A., & Maziero, E. G. (2023). Graph Neural Networks Applied to Money Laundering Detection in Intelligent Information Systems. *Graph Neural Networks Applied to Money Laundering Detection in Intelligent Information Systems*, 252-259. https://doi.org/10.1145/3592813.3592912

Song, J., & Gu, Y. (2023). HBTBD: A Heterogeneous Bitcoin Transaction Behavior Dataset for Anti-Money Laundering. *Applied Sciences*, *13*(15), 8766. https://doi.org/10.3390/APP13158766

Suyal, P., Bhatt, A. K., Pant, J., & Mohan, L. (2021). *Comparative Analysis between Traditional Learning and Deep Learning* (Proceedings – 2021 3rd International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2021). https://doi.org/10.1109/ICAC3N53548.2021.9725370

The European Parliament. (2015). Directive (EU) 2015/ 849 of the European Parliament and of the Council, 20 May 2015. *Official Journal of the European Union*

Thornberry, W. M. (Mac). (2021). National Defense Authorization Act for Fiscal Year 2021. *Public Law*, *116*(283).

Tundis, A., Nemalikanti, S., & Mühlhäuser, M. (2021). Fighting Organized Crime by Automatically Detecting Money Laundering-Related Financial Transactions. In D. Reinhardt, T. Müller (Eds.), *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021* (pp. 38:1-38:10). Association for Computing Machinery. https://doi.org/10.1145/3465481.3469196

UN General Assembly. (2003, October 31). *United Nations Convention against Corruption, 58/4*. https://www.unodc.org/unodc/en/corruption/uncac.html

Vanwinckelen, G., & Blockeel, H. (2012). On Estimating Model Accuracy with Repeated Cross-Validation. *21st Belgian-Dutch Conference on Machine Learning*, 39-44. https://lirias.kuleuven.be/handle/123456789/346385

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., & Bengio, Y. (2017). Graph Attention Networks. *ArXiv: 1710.10903*. https://doi.org/10.17863/CAM.48429

Weber, M., Chen, J., Suzumura, T., Pareja, A., Ma, T., Kanezashi, H., Kaler, T., Leiserson, C., & Schardl, T. (2018). Scalable Graph Learning for Anti-Money Laundering: A First Look. *ArXiv: 1812.00076*

Wong, T. T., & Yeh, P. Y. (2020). Reliable Accuracy Estimates from k-Fold Cross Validation. *IEEE Transactions on Knowledge and Data Engineering*, *32*(8). https://doi.org/10.1109/TKDE.2019.2912815

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How Powerful are Graph Neural Networks? *ArXiv: 1810.00826*.

Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10), 4854-4873. https://doi.org/10.1109/TKDE.2020.3045924

Yang, Z., Ding, M., Zou, X., Tang, J., Xu, B., Zhou, C., & Yang, H. (2023). Region or Global? A Principle for Negative Sampling in Graph-Based Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(6). https://doi.org/10.1109/TKDE.2022.3155155

***

Additional materials are available on the website https://journals.ue.wroc.pl/eada/supplementary-materials

Wykorzystanie heterogenicznych grafowych sieci izomorficznych w analizie danych związanych z praniem brudnych pieniędzy. Studium przypadku FinCEN

Streszczenie

Cel: Celem niniejszej analizy jest opracowanie i zastosowanie nowego modelu HexGIN (heterogeniczne rozszerzenie dla izomorfizmu sieci grafowych) do danych z dochodzenia dziennikarskiego FinCEN oraz porównanie jego jakości predykcji z istniejącymi rozwiązaniami, takimi jak sieć SAGE i wielowarstwowa sieć neuronowa (MLP).

Metodyka: W badaniach wykorzystano dane ze śledztwa FinCEN do opracowania i zastosowania modelu HexGIN w zadaniu przewidywania beneficjenta sieci powiązanych transakcji finansowych. Skuteczność modelu porównano z istniejącymi rozwiązaniami wykorzystującymi sieci neuronowe grafu w serii eksperymentów z walidacją krzyżową.

Wyniki: Eksperymentalne wyniki na danych walidacji krzyżowej i zestawie testowym potwierdzają potencjalne zalety HexGIN w porównaniu z istniejącymi rozwiązaniami, takimi jak MLP i SAGE. Proponowany model przewyższa inne algorytmy pod względem wyniku miary F1, precyzji i ROC AUC, w fazie zarówno treningowej, jak i testowej.

Implikacje i rekomendacje: Wyniki pokazują potencjał heterogenicznych grafowych sieci i ich wysoce ekspresyjnych implementacji, takich jak GIN, w analizie transakcji finansowych. Potrzebne są dalsze badania, zwłaszcza w celu połączenia proponowanego modelu z innymi istniejącymi algorytmami i przetestowania rozwiązania na różnych zestawach danych dotyczących problemu prania brudnych pieniędzy.

Oryginalność/wartość: W przeciwieństwie do wielu badań, które opierają się na syntetycznych lub nieujawnionych źródłach danych związanych z praniem brudnych pieniędzy, to studium przypadku opiera się na publicznie dostępnych, rzeczywistych, heterogenicznych danych transakcyjnych, będących częścią większego śledztwa dziennikarskiego. Wyniki wskazują obiecujący kierunek dla rozwoju nowoczesnych hybrydowych narzędzi do analizy podejrzanych transakcji, opartych na heterogenicznych sieciach grafowych.

Słowa kluczowe: pranie brudnych pieniędzy, uczenie głębokie, analiza sieci, grafy