Journal of Data Science 0 (0), 1–19 ??? 2025

Money Laundering Detection with Multi-Aggregation Custom Edge GIN

FILIP WÓJCIK^{1,*}

¹Faculty of Business Intelligence, Komandorska 118/120 54-132 Wroclaw, Wroclaw University of Economics and Business, Poland

Abstract

Detecting illicit transactions in Anti-Money Laundering (AML) systems remains a significant challenge due to class imbalances and the complexity of financial networks. This study introduces the Multiple Aggregations for Graph Isomorphism Network with Custom Edges (MAGIC) convolution, an enhancement of the Graph Isomorphism Network (GIN) designed to improve the detection of illicit transactions in AML systems. MAGIC integrates edge convolution (GINE Conv) and multiple learnable aggregations, allowing for varied embedding sizes and increased generalization capabilities. Experiments were conducted using synthetic datasets, which simulate real-world transactions, following the experimental setup of previous studies to ensure comparability. MAGIC, when combined with XGBoost as a link predictor, outperformed existing models in 16 out of 24 metrics, with notable improvements in F1 scores and precision. In the most imbalanced dataset, MAGIC achieved an F1 score of 82.6% and a precision of 90.4% for the illicit class. While MAGIC demonstrated high precision, its recall was lower or comparable to the other models, indicating potential areas for future enhancement. Overall, MAGIC presents a robust approach to AML detection, particularly in scenarios where precision and overall quality are critical. Future research should focus on optimizing the model's recall, potentially by incorporating additional regularization techniques or advanced sampling methods. Additionally, exploring the integration of foundation models like GraphAny could further enhance the model's applicability in diverse AML environments.

Keywords deep learning; financial fraud detection; graph neural networks; graph representation learning

1 Introduction

Money laundering represents a formidable challenge to the integrity of global financial systems, enabling illicit activities by concealing the origins of illegally acquired funds. Consequently, Anti-Money Laundering (AML) strategies have emerged as critical for financial institutions and regulatory bodies. The evolution of AML solutions has increasingly embraced advanced technologies, with graph neural networks (GNNs) recognized for their potential to model the intricate relationships within financial transaction networks.

However, AML research and the development of GNN-based solutions are hindered by the scarcity of accessible and realistic datasets (Eddin et al., 2021), which are essential for rigorous model evaluation and refinement. Addressing this challenge, Silva et al. (2023) introduced and

© 2025 The Author(s). Published by the School of Statistics and the Center for Applied Statistics, Renmin University of China. Open access article under the CC BY license. Received September 30, 2024; Accepted May 23, 2025

^{*} Email: filip.wojcik@ue.wroc.pl or ds@filip-wojcik.com.

Wójcik, F.

published a specially designed, imbalanced dataset generated with the IBM AML simulator, alongside their model results, providing a valuable benchmark for assessing the efficiency of graph neural networks in detecting illicit transactions.

This study advances the design and application of GNNs in AML tasks by proposing the Multiple Aggregations for Graph Isomorphism Network with Custom Edges (MAGIC), a novel convolutional architecture that enhances the traditional Graph Isomorphism Network (GIN). MAGIC extends GIN by integrating edge convolution (GINE Conv), expanding its functional scope. Unlike the original GIN, which lacks the capacity for multiple aggregations, MAGIC introduces multiple learnable aggregation mechanisms, allowing for diverse embedding sizes. This adaptability enhances the model's expressiveness, as demonstrated in recent studies (Corso et al., 2020; Tailor et al., 2022; Li et al., 2023). The ability to learn a broader range of feature representations positions MAGIC as effective tool for modeling the complex and imbalanced data structures prevalent in AML applications.

To further optimize performance, this study develops a hybrid link-prediction model that combines the MAGIC convolution with XGBoost, a gradient-boosted decision tree framework. This integration leverages the complementary strengths of graph neural networks and gradient boosting to enhance the detection of illicit transactions, even under the challenging conditions of highly imbalanced datasets.

A comprehensive comparative analysis is also undertaken using the dataset and protocols established by Silva et al. (2023). This analysis enables a direct evaluation of the MAGIC model against existing approaches, using identical experimental setups. Such rigorous comparison not only highlights the capabilities and limitations of MAGIC but also provides valuable insights into its performance relative to competing methodologies in AML detection tasks.

The paper is structured as follows: Section 2 provides an overview of the money laundering problem, AML systems, reviews existing graph neural network studies, and discusses available training datasets for AML. Section 3 outlines the foundations of the MAGIC model, including the GIN and GINE architectures, the formalization of the novel approach, and the experimental design. Section 4 presents the findings and their implications. The paper concludes in Section 5, which summarizes the study and proposes future research directions.

2 Related Work

2.1 Money Laundering and AML Solutions

Money laundering involves converting proceeds from criminal activities into assets that appear legitimate, with the aim of obscuring the origins of illegally acquired funds by exploiting financial systems. Modern legal frameworks, such as the European Union's 2015/849 directive (Parliament, 2015) and the American Anti-Money Laundering Act (Thornberry, 2021), similarly target efforts to conceal or misrepresent the control or ownership of assets involved in financial dealings.

As the global financial system grows more complex, effective AML solutions are becoming increasingly important. The rise of machine learning tools has driven significant advances in AML strategies, ranging from basic rule-based approaches to sophisticated methods incorporating machine learning and graph analytics (Han et al., 2020). Techniques such as Support Vector Machines, Decision Trees, and more recently, GNNs have been effectively employed to identify suspicious activities within transactional data, underscoring their broad applicability in AML tasks (Alsuwailem and Saudagar, 2020; Dumitrescu et al., 2022).

3

Graph Neural Networks, in particular, have become instrumental in modeling the heterogeneous nature of money laundering data, enabling innovative solutions for detecting illicit activities through traditional graph analyses as well as more advanced machine learning pipelines (Alarab et al., 2020; Johannessen and Jullum, 2023). Recent studies illustrate the superiority of GNN-based models over conventional machine learning approaches in identifying suspicious transactions, especially when combining techniques such as self-supervision and hybrid modeling (Silva et al., 2023).

The use of GNNs in money laundering detection spans various tasks, including node classification, link prediction, and anomaly detection within financial networks. For instance, a study by Alarab et al. (2020) applied Graph Convolutional Networks (GCNs) to classify nodes representing transactions as either licit or illicit within the Elliptic dataset. In this study, the GCN model effectively aggregated features of the neighboring nodes, utilizing both convolutional and linear layers to achieve improved classification results. The model surpassed earlier attempts by demonstrating the power of GCNs to incorporate local node information into the final classification outcome.

Similarly, GNNs can also model more complex relationships in heterogeneous data. In a study by Johannessen and Jullum (2023), a heterogeneous extension of the Message Passing Neural Network (MPNN) was applied to a large-scale real-world dataset derived from Norway's largest bank. This dataset encompassed customer information, transaction data, and business role data, forming a highly heterogeneous graph of 5 million nodes and nearly 10 million edges. The heterogeneous MPNN (HMPNN) utilized distinct message-passing operators for each node-edge type combination, effectively capturing the diverse relationships inherent in this dataset. By modeling various types of nodes (e.g., customers, businesses) and edges (e.g., transactions, own-ership), the HMPNN could discern nuanced patterns indicative of money laundering. Notably, the model provided insights into potentially mislabeled customers who were initially identified as regular but exhibited characteristics associated with high-risk activities.

Traditional graph analytics also play an important role in AML analysis, particularly for anomaly detection within transactional networks. Weber et al. (2018) discussed how graph analytics have become a central tool for AML, especially in analyzing cash flow relationships between entities represented as nodes and edges within graphs. Graph representations may vary, from modeling individual accounts as vertices and transactions as edges, to aggregating accounts under a holding entity and studying their relationships collectively. These approaches, including graph-based analysis of cryptocurrency transactions, are increasingly used for forensic investigation, combining clustering techniques with public attribution data to uncover money laundering activities. Classical graph algorithms like cycle detection, PageRank, egonet, and label propagation have proven effective in identifying anomalies and suspicious relationships in both cryptocurrency networks and traditional financial systems. Such algorithms enhance the capacity to identify structural features of the financial network that deviate from normal behavior, assisting in the early detection of fraudulent activities and significantly reducing false positives when analyzed by expert AML professionals.

The versatility of GNNs in addressing money laundering detection becomes evident through their varied applications. These include tasks like node classification for flagging illicit transactions, link prediction for identifying suspicious financial relationships, and graph analytics for deciphering complex network structures. By capturing hidden patterns and relationships within transactional data, GNNs offer a more nuanced understanding of financial networks, thereby enhancing the detection of sophisticated money laundering schemes. When combined with traditional graph analytics, GNNs further increase the effectiveness of AML systems, enabling financial institutions to detect and respond to illicit activities with greater precision and efficiency.

2.2 Graph Neural Networks

GNNs constitute a specialized category of deep learning models designed to interpret graph data intrinsically. Unlike traditional neural network architectures, GNNs can directly incorporate relationships (edges) between nodes (vertices), exhibit resilience against irregular graph structures, and manage large-scale sparse data, while leveraging the capabilities of deep learning algorithms (Wu et al., 2021).

The genesis of GNNs can be traced back to early models like Recurrent Graph Neural Networks by Gori et al. (2005); Scarselli et al. (2009) and Echo State Networks by Gallicchio and Micheli (2010). These models laid the groundwork for identifying structural patterns within graphs and introduced mechanisms for learning that are similar to the concept of backpropagation through time used in recurrent neural networks. Spectral-based Convolutional Graph Neural Networks, presented by Bruna et al. (2014); Defferrard et al. (2016), heralded a pivotal advancement towards the contemporary message-passing paradigm (Wu et al., 2021).

A milestone in this evolution was the introduction of the GCN (Kipf and Welling, 2017). Due to its simplicity and adaptability, it established a foundation for developing the MPNN (Gilmer et al., 2017) and its subsequent variants. A comprehensive meta-analysis by Wu et al. (2021) identified over forty distinct GNN variants utilizing the MPNN paradigm. Prominent models that have significantly influenced the field include:

- 1. GraphSAGE (Hamilton et al., 2017) enhances the message-passing framework's aggregation phase by integrating the node's current features (skip-connection), followed by a pooling operation.
- 2. Graph Attention Network (GAT) (Velickovic et al., 2017) introduces attention mechanisms, similar to those utilized in natural language processing models, to message-passing.
- 3. GIN network (Xu et al., 2018) demonstrated to theoretically match the Weisfeiler-Lehman graph isomorphism test's expressiveness through the deployment of nested multi-layer perceptron (MLP) networks.

These models have traditionally focused on homogeneous graphs, which consist of a single type of node and edge. However, recent research has increasingly focused on heterogeneous graphs, which encompass multiple types of nodes and edges, reflecting a broader range of realworld scenarios and complex knowledge representations (Shi, 2022). For instance, heterogeneous graphs can effectively model academic publishing activities, denoting relationships such as author-writes-paper, reviewer-reviews-paper, and editor-publishes-paper.

A systematic survey (Yang et al., 2020) categorizes over forty heterogeneous GNN architectures into families based on their processing methodologies and transformation techniques. These encompass message-passing network variants, proximity-preserving methods, and relational learning strategies. The challenge of integrating heterogeneous relations without compromising the network's depth, stability, and generalization capacity remains one of the paramount areas of research within the field (Wu et al., 2021; Shi, 2022).

2.3 AMLSim Datasets

A major challenge in money laundering research is the limited availability of publicly accessible, real-world datasets. The confidential nature of financial data, particularly those obtained during money laundering investigations, coupled with legal and privacy constraints, significantly restricts access to authentic datasets for research purposes. This scarcity of real-world data poses obstacles to the reproducibility and transparency of studies on AML models, often forcing researchers to rely on simulated or synthetic datasets that may not fully capture the complexity of actual money laundering operations.

This study builds upon the datasets and methodological framework established by Silva et al. (2023), which utilized data generated by the financial transaction simulation tool 'AML-Sim', developed by IBM (Weber et al., 2018). AMLSim extends the functionality of the PaySim architecture and is specifically designed to simulate transactional patterns indicative of money laundering activities. This simulator is notable for its ability to fine-tune a wide range of parameters, including the duration of the simulation in time steps (e.g., days), transaction value ranges, class imbalance ratios, and other statistical variables. In their study, Silva et al. (2023) generated a dataset representing daily transactional activities over the year 2020, comprising 365 time steps. The dataset includes account identifiers, predefined suspicion indicators, details of sending and receiving accounts, and transaction types. Detailed descriptions of the dataset attributes and their interpretation are provided in the supplementary material.

To ensure comparability with the methodologies and datasets used in that study and to guarantee consistency and reproducibility of results, this work adopts the same datasets, feature attributes, and train-test splits as specified by Silva et al. (2023).

The contribution of Silva et al. (2023) is particularly valuable, providing a standardized synthetic dataset and a methodological reference point for AML research. By offering simulated transactional data that approximates real-world conditions, their work facilitates model evaluation and comparison while addressing the reproducibility gap in the field. This study builds upon their contributions, leveraging these resources to further advance the development of AML models.

3 Proposed Method

This section provides an overview of the proposed method and its core components. It includes a description of the MPNN framework and the variants of the GIN model, followed by an introduction to the new MAGIC model, and concludes with an explanation of the hybrid link predictor and the experimental design.

3.1 MPNN and GIN

Many modern graph neural networks are defined in terms of MPNN equations (Gilmer et al., 2017; Hamilton, 2020). In this context, a graph is described formally as $G = \{\mathcal{V}, \mathcal{E}\}$, where \mathcal{V} is a set of vertices and \mathcal{E} is a set of edges. Each vertex $u \in \mathcal{V}$ and edge $e_{ij} \in \mathcal{E}$ are represented as vectors $h_u, h_v \in \mathbb{R}^d$ and $e_{ij} \in \mathbb{R}^e$, respectively, where d and e denote the dimensions of the vertex and edge embeddings or initial features.

The general message passing process at iteration k involves two main steps. First, the aggregation step uses the function AGGREGATE^(k) to combine information from the neighbors of node v into an intermediate message representation

$$m_{\mathcal{N}(v)}^{(k)} = \text{AGGREGATE}^{(k)} \left(\left\{ h_u^{(k)} \mid \forall u \in \mathcal{N}(v) \right\} \right), \tag{1}$$

where $\mathcal{N}(v)$ denotes the set of neighbors of node v. Next, the update step employs the function

 $UPDATE^{(k)}$ to integrate the aggregated message with the node's current representation

$$h_{v}^{(k+1)} = \text{UPDATE}^{(k)} \left(h_{v}^{(k)}, m_{\mathcal{N}(v)}^{(k)} \right).$$
(2)

This framework serves as the foundation for various GNN models.

The model proposed in this study is an extension of the GIN (Xu et al., 2018) and its edge-processing version (Liu et al., 2020; Hu et al., 2019) for homogeneous graphs. The original GIN is formalized as

$$h_{v}^{(k)} = \mathrm{MLP}\left(\left(1 + \epsilon^{(k)}\right)h_{v}^{(k-1)} + \sum_{u \in \mathcal{N}(v)}h_{u}^{(k-1)}\right),\tag{3}$$

where MLP : $\mathbb{R}^d \to \mathbb{R}^{d'}$ is a multi-layer perceptron, $\epsilon^{(k)}$ is a learnable parameter that modulates self-loops, and $h_v^{(k-1)}$ and $h_u^{(k-1)}$ are the embeddings of nodes v and u from the previous iteration.

Because of the non-linearity introduced by the MLP, GIN is theoretically proven to behave like a graph-injection function, properly identifying nodes in the graphs and passing a Weisfeiler-Lehman graph isomorphism test (Xu et al., 2018; You et al., 2020).

Transformations applied by GIN convolution can be depicted as in Figure 1, where each set of colored squares represents a vector of a given dimensionality.

Subsequent research has focused on enhancing GNNs through various techniques, including multiple aggregation functions (Corso et al., 2020; Tailor et al., 2022), or adaptive, learnable aggregations (Li et al., 2023). A significant development was the introduction of GINE convolution



Figure 1: GIN embedding visualization.

for edge-inclusive graphs (Hu et al., 2019). Unlike previously described models, GINE explicitly integrates edge features into the aggregation process, which can be formalized as

$$m_{\mathcal{N}(v)}^{(k)} = \text{EDG-AGGREGATE}\left(\{h_v^{(k-1)}, h_u^{(k-1)}, e_{uv} \mid u \in \mathcal{N}(v)\}\right),\tag{4}$$

where EDG-AGGREGATE is the aggregation function that combines edge features $e_{uv} \in \mathbb{R}^e$ with neighboring node embeddings. A simplified formulation by Brossard et al. (2020) is defined as

$$m_{\mathcal{N}(v)}^{(k)} = \sum_{u \in \mathcal{N}(v)} \sigma \left(h_u^{(k-1)} + E(e_{uv}) \right),$$

$$h_v^{(k)} = \mathrm{MLP}\left((1+\epsilon) h_v^{(k-1)} + m_{\mathcal{N}(v)}^{(k)} \right),$$
(5)

where $E : \mathbb{R}^e \to \mathbb{R}^d$ is a neural network that maps edge features to the same dimensionality as node embeddings, and σ is a non-linear activation function, such as rectified linear unit.

3.2 MAGIC Architecture

The proposed MAGIC model extends GINE by incorporating multiple aggregation functions and handling embeddings with varying dimensionalities. For a node v, its embedding from the previous iteration $h_v^{(k-1)} \in \mathbb{R}^d$, and its neighbors' embeddings $\{h_u^{(k-1)} \in \mathbb{R}^d \mid u \in \mathcal{N}(v)\}$, we define \mathcal{I} parameterized aggregation functions as

$$f_i\left(\left\{h_u^{(k-1)} \mid u \in \mathcal{N}(v)\right\}; \theta_i\right) \text{ for } i = 1, \dots, \mathcal{I},\tag{6}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}^{d_i}$ maps embeddings to a new dimensionality d_i .

This design allows for capturing various neighborhood features with distinct dimensionalities, enriching the representation capacity of the model as presented in (Corso et al., 2020; Tailor et al., 2022; Li et al., 2023). The aggregated messages are combined using concatenation (denoted by the \bigoplus), as

$$m_{\mathcal{N}(v)}^{(k)} = \bigoplus_{i=1}^{\mathcal{I}} f_i \left(\left\{ h_u^{(k-1)} + \operatorname{ANET}^{(k)}(e_{u,v}) \mid u \in \mathcal{N}(v) \right\}; \theta_i \right),$$

$$m_{\mathcal{N}(v)}^{(k)} \in \mathbb{R}^{(d_1+d_2+\dots+d_{\mathcal{I}})},$$
(7)

where $ANET^{(k)} : \mathbb{R}^e \to \mathbb{R}^d$ is a neural network used in k-th convolution step, that aligns edge features with neighbor embeddings.

Given the aforementioned elements, the new representation for node v, and the final output of the MAGIC graph convolution model, can be obtained in a similar way as in GIN or GINE, formalized as

$$h_{v}^{(k)} = \mathrm{MLP}^{(k)}\left((1+\epsilon^{(k)})h_{v}^{(k-1)}\bigoplus m_{\mathcal{N}(v)}^{(k)}\right),\tag{8}$$

where $MLP^{(k)}$ is a MLP network used for the k-th convolution step.

This approach supports the use of any learnable aggregation functions, including their generalized or parameterized forms (Tailor et al., 2022; Li et al., 2023). The skip-connection mechanism of GIN is maintained by concatenating neighborhood embeddings with the node v vector. During the backpropagation phase, the MAGIC model employs a technique known as stochastic weight averaging (Izmailov et al., 2018), which has demonstrated strong predictive performance in existing studies. Algorithm 1 summarizes the MAGIC embedding mechanism, and Figure 2 illustrates the flow of the convolution process.

Algorithm 1: MAGIC graph convolution embedding.
Result: Updated node embeddings for graph G
Input: Graph: $G = \{\mathcal{V}, \mathcal{E}\}$
Edge features: $\mathcal{E} = \{ e_{uv} \in \mathbb{R}^e \mid \forall u, v \in \mathcal{V} \}$
Initial node embeddings: $h_u^{(0)} \in \mathbb{R}^d, \forall u \in \mathcal{V}$
Number of convolution steps: K Set of aggregation functions:
$f_i : \mathbb{R}^d \to \mathbb{R}^{d_i}$, parameterized by θ_i , for $i = 1, \dots, \mathcal{I}$
Edge-alignment network for each step: $ANET^{(k)} : \mathbb{R}^e \to \mathbb{R}^d$
MLP for each step: $MLP^{(k)} : \mathbb{R} \to \mathbb{R}^{d'}$
1 for $k = 1$ to K do
2 for each node $v \in \mathcal{V}$ do
3 Compute the neighborhood aggregation $m_{\mathcal{N}(v)}^{(k)}$ by concatenating multiple
aggregations:
\mathcal{I}
$m_{\mathcal{N}(w)}^{(k)} = \bigoplus f_i \left(\{ h_w^{(k-1)} + \operatorname{ANET}^{(k)}(e_{uv}) \mid u \in \mathcal{N}(v) \}; \theta_i \right)$
$\prod_{i=1}^{N(0)} \bigcup_{i=1}^{N(0)} \bigcup_{i$
Update the embedding $h_v^{(k)}$ by applying the k-th MLP to the concatenation of
v's previous embedding and the aggregated neighborhood message:
$h^{(k)} - \operatorname{MIP}^{(k)} \left((1 + \epsilon^{(k)}) h^{(k-1)} \bigoplus m^{(k)} \right)$
$n_v = \text{MLL} \left((1 + \epsilon^{-v}) n_v - \sum m_{\mathcal{N}(v)} \right)$
4 end
b = hu
$0 \text{return} \{n_v^{-1} \mid v \ v \in V\}$

3.3 Link Predictor Module

The construction of node embeddings is the first step in the prediction process. However, the primary goal of this study is to perform binary link classification.

To achieve this, an additional model, XGBoost (XGB, Extreme Gradient Boosting) (Chen and Guestrin, 2016), is applied on top of the node embeddings. XGB, as a member of the ensemble models family, is flexible and well-suited for processing data in tabular or matrix format. Its application in graph link prediction has been demonstrated in previous studies (Behera et al., 2021).

In this experiment, after generating the node embeddings from the MAGIC model, embeddings of nodes corresponding to each edge are concatenated and paired with the corresponding edge label for training the XGB model. Given the set of all binary-labeled edges \mathcal{E} , defined as $\{(e_{uv}, l_{e_{uv}}) \mid e_{uv} \in \mathcal{E}, l_{e_{uv}} \in \{0, 1\}\}$, and the node embeddings from the k-th iteration, $h_u^{(k)}$ and $h_v^{(k)}$, the dataset for training the XGB model is constructed as

$$D = \left\{ \left(h_u^{(k)} \bigoplus h_v^{(k)}, l_{e_{uv}} \right) \mid \forall u, v \in \mathcal{V}, e_{uv} \in \mathcal{E} \right\},\tag{9}$$

where $l_{e_{uv}} \in \{0, 1\}$ is the label of edge e_{uv} , and $h_u^{(k)} \bigoplus h_v^{(k)}$ represents the concatenated embedding of nodes u and v.



Figure 2: MAGIC model embedding.

The XGB model is trained on this dataset using a standard supervised-learning scheme, where the input consists of concatenated node embeddings and the output corresponds to the binary edge labels.

The link prediction module is summarized as pseudocode in Algorithm 2, and its processing flow is illustrated in Figure 3 (edge features are omitted for clarity).

3.4 Experimental Design

The primary objective of this study is to classify graph edges, which represent financial transactions, as either illicit or legal. In the utilized graph dataset, nodes correspond to individual customers or accounts, characterized by features such as initial deposit values and behavioral categories. The edges between these nodes signify transactions and capture attributes such as

Algorithm 2: XGBoost	link prediction	module.
----------------------	-----------------	---------

Result: Trained XGBoost model for link prediction

Input: Graph $G = \{\mathcal{V}, \mathcal{E}\}$ Edge labels $\{l_{e_{uv}} \mid \forall e_{uv} \in \mathcal{E}\}$ Node embeddings $\{h_v^{(k)} \mid \forall v \in \mathcal{V}\}$ Loss function \mathcal{L}

1 for each edge $e_{uv} \in \mathcal{E}$ do

2 Concatenate MAGIC embeddings for nodes *u* and *v*:

$$h_{uv} = h_u^{(k)} \bigoplus h_v^{(k)}$$

Predict the label of the edge using the XGBoost model:

$$\hat{y}_{uv} = \text{XGB}(h_{uv})$$

Calculate the error using the loss function:

error =
$$\mathcal{L}(\hat{y}_{uv}, l_{e_{uv}})$$

Update the XGBoost model using the calculated error.

3 end

4 return Trained XGBoost model



Figure 3: Link predictor visualization.

transaction amount, timestamp, and type. These features are integrated into a graph structure, allowing the model to identify suspicious connections and flag activities indicative of money laundering.

Each dataset was divided into predefined train-test splits as established by the authors of the original study, ensuring reproducibility and comparability across trials. To optimize the MAGIC architecture, the training portion of the data was further divided into 80% for training and 20% for validation to facilitate hyperparameter tuning. Optuna was used to perform this optimization, balancing two objectives: maximizing the illicit F1 score and minimizing model complexity, defined as the product of the number of convolutional layers and their dimensions.

The search space included the number and size of convolutional layers, neighborhood aggregation methods, and embedding reduction modes, while certain parameters—such as the optimizer, learning rate, batch size, and number of epochs—were fixed across all datasets. This approach ensured consistency and comparability across trials while focusing the analysis on evaluating the key architectural innovations introduced by the MAGIC model. By fixing these parameters to reasonable defaults, the study aimed to isolate the effects of the network components being optimized.

Machine learning models inherently exhibit prediction variability across training trials. This variability is further exacerbated in graph-based supervised learning, where induced subgraphs for validation may lead to information leakage, complicating the reliability of quality assessment methods (Leskovec and Faloutsos, 2006; Leskovec and Sosič, 2016). Following the prior research, the MAGIC model was trained 10 times for each dataset, using predefined seeds for random restarts, and evaluated at the end on the test subset (Raschka, 2018). Metrics were recorded along with their standard deviations. To quantify uncertainty, 95% bootstrap confidence intervals were constructed using the percentile approach (Bouthillier et al., 2021; Raschka, 2018) and the t-distribution, mirroring the methodology of Silva et al. (2023). Summaries of metrics, rather than point estimates, are presented in the results section to reflect performance variability.

Given the strong class imbalance, a range of evaluation metrics was employed, including precision, recall, and F1-score (Sammut and Webb, 2011). Metrics were calculated for the illicit class separately, given its critical importance in real-world AML applications, as well as for the macro average, representing the unweighted mean of scores across all classes (Opitz and Burst, 2019).

The tuning process revealed a consistent trade-off between model complexity and illicit F1 scores. Simpler models, characterized by fewer layers and smaller convolutional dimensions, matched or outperformed more complex variants every time. Figure 4 illustrates the relationship between model complexity and illicit F1 scores across all datasets. As a result, simpler architectures were selected whenever performance differences were negligible.

While a traditional ablation study was not conducted, the contribution of each hyperparameter to model performance was systematically evaluated using the PED-ANOVA Importance. This method serves as a possible alternative to ablation studies, particularly in cases where the number of tuned parameters is large and their interactions form subspaces (e.g., the interplay between the number of layers, layer size, and aggregation methods per layer). By leveraging ANOVA-based calculations, PED-ANOVA quantifies how strongly each hyperparameter contributes to achieving performance above a baseline, which is determined by fitting Parzen estimators to the results of completed trials (Watanabe et al., 2023).

The feature importance analysis revealed that no single hyperparameter dominated performance across all datasets. Instead, the relative importance varied depending on dataset properties, particularly the class imbalance ratio and dataset size. For less-imbalanced cases (AMLSim

Wójcik, F.



Figure 4: Trade-off between model complexity (number of layers \times convolution size) and illicit F1 scores for each dataset.

1/3 and 1/5), the number and size of convolutional layers, as well as aggregation functions, were relatively more influential, though their importance scores were comparable. In contrast, in highly imbalanced scenarios (AMLSim 1/10 and 1/20), embedding reduction modes became more critical, alongside aggregation functions.

This variability highlights the importance of adaptive hyperparameter tuning tailored to dataset-specific properties. The flexibility of the MAGIC architecture further supports effective performance under a wide range of configurations, including the use of multiple, learnable aggregations. Additional figures, details of the tuning process and feature importances, are provided in the supplementary materials.

Table 1 summarizes the MAGIC hyperparameters for each dataset. For XGBoost, the default setting of 250 estimators with a learning rate of 0.05 and positive class weight scaling of 1:1 was used for all datasets except AMLSim 1/3 and AMLSim 1/5, where the number of estimators was increased to 670. For AMLSim 1/20, varied class weights were employed to address the strongest class imbalance.

The performance of the MAGIC model was compared to previously established approaches from the study by Silva et al. (2023). Specifically, evaluations were conducted against GCN (320 total parameters), NENN (560 total parameters), and SkipGCN (176 total parameters). NENN, proposed by Yang and Li (2020), incorporates both node and edge features to enrich graph relationships, while SkipGCN, introduced by Weber et al. (2019), employs skip connections to capture long-range dependencies, which are especially relevant in financial transaction networks.

Dataset	MAGIC Embedding				
	Conv. Dim	No. Conv. Layers	Aggr.	Embed. Reduction	Total params.
AMLSim 1/3	8	2	Add, Min, Max	Averaging	480
AMLSim $1/5$	16	2	Add, Mean	Concatenation	1288
AMLSim $1/10$	16	2	Add, Min, Max	Concatenation	1640
AMLSim $1/20$	32, 8	2	Add, Mean	Concatenation	1728

Table 1: Hyperparameters for the MAGIC convolution.

Although MAGIC's parameter count is higher, ranging from 480 to 1728 depending on the dataset, this increase arises from its adaptable design. MAGIC is crafted to function as a single, versatile architecture that performs well across datasets with diverse characteristics, such as varying class imbalances. Its configurability, achieved through features like multiple aggregation strategies and adjustable embedding reduction methods, allows fine-tuning for a broad range of AML scenarios. This adaptability offers advantages over simpler, more specialized models.

The additional complexity in MAGIC is balanced to enhance performance without sacrificing computational efficiency. For instance, while its parameter count exceeds those of GCN, NENN, and SkipGCN, it remains modest compared to modern graph models like graph transformers (Ying et al., 2021), which often involve tens of thousands of parameters.

4 Empirical Results

This section presents the overview of datasets, used in each experimental trial and the summary of model training results.

4.1 Dataset Characteristics

Table 2 summarizes the statistics for each dataset, including the number of nodes in each graph, the number of edges, the number of connected components, and the proportion of classes (illicit vs. legal). Each node is characterized by the vector of 6 features, each edge—by the vector of 8 features.

The label distribution indicates a significant class imbalance, particularly in the AMLSim 1/10 and AMLSim 1/20 datasets, where the proportion of illicit transactions is relatively low. The large number of connected components suggests that these graphs are composed of many

Table 2: Dataset overview.					
Dataset	# nodes # edges # Connected comp.		Cla	sses	
				Legal	Illicit
AMLSim 1/3	6331	3213	3118	66%	33%
AMLSim $1/5$	10507	5355	5154	80%	20%
AMLSim 1/10	20751	10710	10058	90%	10%
AMLSim $1/20$	40264	21420	18914	95%	5%

independent subgraphs, which may present challenges in terms of model performance and generalization.

4.2 Results

The results for each model can be found in Table 3. The scores for models GCN, GCN + XGBoost, NENN, NENN + XGBoost, SkipGCN, and SkipGCN + XGBoost (marked with an asterisk) were obtained from the original study by Silva et al. (2023). The best-performing model for each dataset and metric is highlighted in bold font.

Models	F 1	Precision	Recall	
AMLSim 1/3 – Illicit				
MAGIC+XGB	0.956 ± 0.018	0.964 ± 0.012	0.967 ± 0.036	
GCN*	0.947	0.913 ± 0.002	0.984	
SkipGCN*	0.948 ± 0.002	0.913 ± 0.003	0.984	
GCN+XGB*	0.949 ± 0.011	0.928 ± 0.010	0.971 ± 0.020	
NENN+XGB*	0.946 ± 0.017	0.918 ± 0.024	0.976 ± 0.021	
NENN*	0.933 ± 0.021	0.894 ± 0.023	0.975 ± 0.030	
SkipGCN+XGB*	0.948 ± 0.010	0.927 ± 0.011	0.970 ± 0.019	
	AMLSim 1/	3 – Macro		
MAGIC+XGB	$\boldsymbol{0.966\pm0.014}$	0.964 ± 0.012	0.968 ± 0.017	
GCN*	0.962	0.953	0.973	
SkipGCN*	0.962	0.953 ± 0.002	0.973	
GCN+XGB*	0.964 ± 0.008	0.958 ± 0.006	0.970 ± 0.010	
NENN+XGB*	0.961 ± 0.013	0.954 ± 0.014	0.970 ± 0.012	
NENN*	0.951 ± 0.015	0.941 ± 0.015	0.963 ± 0.017	
SkipGCN+XGB*	0.963 ± 0.007	0.957 ± 0.006	0.969 ± 0.009	
	$\mathbf{AMLSim} \ 1_{\mathbf{AMLSim}}$	$5-{ m Illicit}$		
MAGIC+XGB	0.910 ± 0.016	0.937 ± 0.009	0.927 ± 0.035	
GCN*	0.901 ± 0.011	0.840 ± 0.021	0.973	
SkipGCN*	0.902 ± 0.012	0.840 ± 0.022	0.974 ± 0.003	
GCN+XGB*	0.905 ± 0.019	0.878 ± 0.024	0.933 ± 0.029	
NENN+XGB*	0.895 ± 0.025	0.873 ± 0.043	0.919 ± 0.028	
NENN*	0.872 ± 0.023	0.823 ± 0.031	0.927 ± 0.032	
SkipGCN+XGB*	0.904 ± 0.017	0.876 ± 0.025	0.934 ± 0.029	
${\bf AMLSim} 1/5-{\rm Macro}$				
MAGIC+XGB	$\boldsymbol{0.943} \pm \boldsymbol{0.010}$	0.937 ± 0.009	0.949 ± 0.016	
GCN*	0.939 ± 0.007	0.917 ± 0.010	0.966 ± 0.003	
SkipGCN*	0.939 ± 0.008	0.917 ± 0.011	0.966 ± 0.003	
GCN+XGB*	0.941 ± 0.012	0.931 ± 0.013	0.952 ± 0.015	
NENN+XGB*	0.936 ± 0.016	0.928 ± 0.022	0.945 ± 0.014	
NENN*	0.921 ± 0.014	0.903 ± 0.016	0.941 ± 0.016	
SkipGCN+XGB*	0.941 ± 0.011	0.931 ± 0.012	0.952 ± 0.014	

Table 3: Model scores across datasets.

Models	F 1	Precision	Recall
	AMLSim 1/	10 – Illicit	
MAGIC+XGB	0.856 ± 0.033	0.941 ± 0.02	0.814 ± 0.042
GCN*	0.810 ± 0.035	0.724 ± 0.058	0.920 ± 0.009
SkipGCN*	0.807 ± 0.033	0.719 ± 0.056	0.920 ± 0.010
GCN+XGB*	0.814 ± 0.032	0.786 ± 0.052	0.846 ± 0.040
NENN+XGB*	0.837 ± 0.027	0.828 ± 0.035	0.847 ± 0.045
NENN*	0.725 ± 0.041	0.592 ± 0.051	0.936 ± 0.032
SkipGCN+XGB*	0.812 ± 0.030	0.779 ± 0.050	0.847 ± 0.032
	AMLSim 1/1	10 – Macro	
MAGIC+XGB	0.920 ± 0.018	0.941 ± 0.02	0.902 ± 0.022
GCN*	0.894 ± 0.020	0.858 ± 0.029	0.942 ± 0.005
SkipGCN*	0.892 ± 0.019	0.855 ± 0.028	0.942 ± 0.005
GCN+XGB*	0.898 ± 0.018	0.885 ± 0.026	0.911 ± 0.020
NENN+XGB*	0.910 ± 0.015	0.906 ± 0.018	0.914 ± 0.022
NENN*	0.844 ± 0.024	0.793 ± 0.026	0.936 ± 0.017
SkipGCN+XGB*	0.896 ± 0.017	0.882 ± 0.025	0.912 ± 0.015
	AMLSim 1/	20 - Illicit	
MAGIC+XGB	0.826 ± 0.041	0.904 ± 0.027	0.834 ± 0.047
GCN*	0.723 ± 0.036	0.673 ± 0.062	0.782 ± 0.020
SkipGCN*	0.723 ± 0.034	0.672 ± 0.057	0.782 ± 0.016
GCN+XGB*	0.743 ± 0.064	0.813 ± 0.083	0.685 ± 0.061
NENN+XGB*	0.745 ± 0.042	0.777 ± 0.063	0.717 ± 0.059
NENN*	0.465 ± 0.063	0.311 ± 0.056	0.926 ± 0.067
SkipGCN+XGB*	0.741 ± 0.042	0.814 ± 0.069	0.681 ± 0.041
	AMLSim 1/2	20 – Macro	
MAGIC+XGB	0.908 ± 0.022	0.904 ± 0.027	0.912 ± 0.024
GCN*	0.854 ± 0.019	0.831 ± 0.031	0.882 ± 0.010
SkipGCN*	0.854 ± 0.018	0.831 ± 0.029	0.882 ± 0.008
GCN+XGB*	0.866 ± 0.033	0.899 ± 0.043	0.839 ± 0.032
NENN+XGB*	0.867 ± 0.022	0.882 ± 0.032	0.853 ± 0.029
NENN*	0.706 ± 0.039	0.654 ± 0.028	0.914 ± 0.031
SkipGCN+XGB*	0.865 ± 0.022	0.900 ± 0.035	0.837 ± 0.021

Table 3: (Continued).

The results across datasets highlight the sophisticated performance profile of MAGIC combined with XGBoost. The model achieved the highest scores in 16 out of 24 metrics, though it did not excel across all evaluation criteria. Its strengths were most pronounced in metrics such as F1 score and precision for both illicit and macro calculations, where it consistently outperformed other approaches. However, the recall scores were either lower or comparable to those of simpler models.

For instance, NENN, with only 560 parameters, achieved higher recall in the most imbalanced datasets, scoring 0.936 in AMLSim 1/10 and 0.926 in AMLSim 1/20, compared to 0.902 and 0.834, respectively, for MAGIC+XGBoost. This suggests that NENN is better suited for

Wójcik, F.

scenarios where sensitivity is paramount, as it captures more illicit transactions. However, this comes at the expense of precision, where the proposed method consistently outperformed all other models, ensuring fewer false positives — an important requirement in real-world AML applications to reduce unnecessary alerts and investigations.

The trade-off between precision and recall underscores the necessity of tailoring models to the specific characteristics of each dataset. Simpler models, such as GCN (320 parameters) and especially SkipGCN (176 parameters), exhibited strong recall performance, particularly in less imbalanced datasets like AMLSim 1/3. In these cases, the additional complexity of MAGIC+XG-Boost offered limited advantages. These findings highlight that simpler architectures can be sufficient for achieving high sensitivity, emphasizing that increased model complexity does not inherently ensure better performance across all evaluation metrics.

Nevertheless, the design of MAGIC+XGBoost offers advantages in practical AML scenarios. Its ability to flag transactions with high precision ensures that most flagged records are indeed illicit, minimizing the impact of false positives on investigative teams. While this conservative approach may overlook some true positives, particularly in datasets with less pronounced class imbalances, the overall balance achieved by the proposed method makes it a robust solution in contexts where precision and F1 score are prioritized.

5 Discussion

This study introduced MAGIC, a novel graph convolution model tailored for AML tasks. By combining the GIN architecture with multiple learnable aggregation methods and integrating edge features into the convolution layer, MAGIC, coupled with XGBoost, delivered strong performance across key metrics. It excelled particularly in F1 score and precision, achieving the best scores in 16 out of 24 metrics on publicly available AMLSim datasets. These results highlight its robustness in identifying illicit transactions while minimizing false positives, a critical requirement for practical AML applications.

However, the study was limited to datasets with a maximum class imbalance ratio of 1:20 (5% illicit transactions), following the methodology of Silva et al. (2023) to ensure consistency and comparability. The lack of more imbalanced publicly available datasets remains a significant challenge for AML research, as real-world scenarios often involve much lower proportions of illicit transactions. Future studies should evaluate the performance of MAGIC and similar methods under extreme imbalance conditions, such as 1:1000 or 1:10,000 ratios, to better reflect practical challenges. Simulators such as AMLSim can be employed to generate datasets with such extreme ratios, enabling reproducible benchmarking with both current and previous studies.

Unsupervised anomaly detection methods, including clustering techniques and graph-based autoencoders, offer promising alternatives for handling highly imbalanced data. These methods do not require labeled datasets and instead identify anomalies by detecting deviations from normal patterns. Future research should investigate their applicability to datasets with rich node and edge features, such as those used in this study. A systematic comparison of supervised methods, such as MAGIC and models from Silva et al. (2023), with unsupervised techniques could provide valuable insights into their relative strengths and limitations across varying imbalance scenarios.

Finally, foundation models, such as the recently proposed GraphAny (Zhao et al., 2024), present an exciting direction for future AML research. These models leverage manifold transformations of features, offering innovative ways to capture patterns in complex datasets. By training on publicly available synthetic data and transferring knowledge to private, domainspecific datasets, foundation models could address challenges related to generalization while maintaining privacy. Incorporating such approaches into future studies could provide powerful tools for tackling diverse and highly imbalanced AML scenarios, bridging the gap between research and practical applications.

Supplementary Material

The source code for this study is available on GitHub: https://github.com/maddataanalyst/Graph_MAGIC_Conv. The repository includes all the necessary components to reproduce the training results.

A supplementary PDF file attached to this publication provides detailed analyses of the train/validation/test splits, hyperparameter tuning results, and a comprehensive breakdown of the model architecture for each dataset.

References

- Alarab I, Prakoonwit S, Nacer MI (2020). Competence of graph convolutional networks for antimoney laundering in bitcoin blockchain. In: *ICMLT 2020: 5th International Conference on Machine Learning Technologies*. Beijing, China, June, 2020, 23–27. Association for Computing Machinery.
- Alsuwailem AAS, Saudagar AKJ (2020). Anti-money laundering systems: A systematic literature review. Journal of Money Laundering Control, 23: 833–848. https://doi.org/10.1108/JMLC-02-2020-0018
- Behera DK, Das M, Swetanisha S, Nayak J, Vimal S, Naik B (2021). Follower link prediction using the xgboost classification model with multiple graph features. Wireless Personal Communications, 127(1): 695–714. https://doi.org/10.1007/s11277-021-08399-y
- Bouthillier X, Delaunay P, Bronzi M, Trofimov A, Nichyporuk B, Szeto J, et al. (2021). Accounting for variance in machine learning benchmarks. In: *Proceedings of Machine Learning* and Systems (A Smola, A Dimakis, I Stoica, eds.), volume 3, 747–769.
- Brossard R, Frigo O, Dehaene D (2020). Graph convolutions that can finally model local structure. *CoRR*. arXiv preprint: https://arxiv.org/abs/2011.15069.
- Bruna J, Zaremba W, Szlam A, LeCun Y (2014). Spectral networks and deep locally connected networks on graphs. In: Conf. on Learning Representations, ICLR 2014—Conference Track Proceedings (Y Bengio, Y LeCun, eds.), 1–14.
- Chen T, Guestrin C (2016). Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining— KDD '16, 785–794. ACM Press.
- Corso G, Cavalleri L, Beaini D, Liò P, Velickovic P (2020). Principal neighbourhood aggregation for graph nets. Advances in Neural Information Processing Systems, 33: 13260–13271.
- Defferrard M, Bresson X, Vandergheynst P (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in Neural Information Processing Systems, 29: 3837–3845.
- 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 AN, Bono J, Aparício D, Polido D, Ascensão JT, Bizarro P, et al. (2021). Anti-money laundering alert optimization using machine learning with graphs. *CoRR*. arXiv preprint: https://arxiv.org/abs/2112.07508.
- Gallicchio C, Micheli A (2010). Graph echo state networks. In: The 2010 International Joint Conference on Neural Networks (IJCNN), 1–10. IEEE.
- Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017). Neural message passing for quantum chemistry. In: *ICML* (D Precup, YW Teh, eds.), volume 70 of *Proceedings of Machine Learning Research*, *PMLR*, 1263–1272.
- Gori M, Monfardini G, Scarselli F (2005). A new model for learning in graph domains. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, 729–734.*
- Hamilton WL (2020). Graph representation learning Hamilton. Synthesis Lectures on Artificial Intelligence and Machine Learning, 14: 1–159. https://doi.org/10.1007/978-3-031-01588-5
- Hamilton WL, Ying Z, Leskovec J (2017). Inductive representation learning on large graphs. In: NIPS (I Guyon, U von Luxburg, S Bengio, HM Wallach, R Fergus, SVN Vishwanathan, R Garnett, eds.), 1024–1034.
- Han J, Huang Y, Liu S, Towey K (2020). Artificial intelligence for anti-money laundering: A review and extension. *Digital Finance*, 2: 211–239. https://doi.org/10.1007/s42521-020-00023-1
- Hu W, Liu B, Gomes J, Zitnik M, Liang P, Pande VS, et al. (2019). Pre-training graph neural networks. *CoRR*. arXiv preprint: https://arxiv.org/abs/1905.12265.
- Izmailov P, Podoprikhin D, Garipov T, Vetrov D, Wilson AG (2018). Averaging weights leads to wider optima and better generalization. In: UAI (A Globerson, R Silva, eds.), volume 2, 876–885. AUAI Press.
- Johannessen F, Jullum M (2023). Finding money launderers using heterogeneous graph neural networks. *CoRR*. arXiv preprint: https://arxiv.org/abs/2307.13499.
- Kipf TN, Welling M (2017). Semi-supervised classification with graph convolutional networks. In: *ICLR Poster*, 1–14.
- Leskovec J, Faloutsos C (2006). Sampling from large graphs. In: *KDD* (T Eliassi-Rad, LH Ungar, M Craven, D Gunopulos, eds.), volume 2006, 631–636. Association for Computing Machinery.
- Leskovec J, Sosič R (2016). Snap: A general-purpose network analysis and graph-mining library. ACM Transactions on Intelligent Systems and Technology, 8: 1–20.
- Li G, Xiong C, Qian G, Thabet A, Ghanem B (2023). Deepergen: Training deeper gens with generalized aggregation functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45: 13024–13034.
- Liu F, Xue S, Wu J, Zhou C, Hu W, Paris C, et al. (2020). Deep learning for community detection: Progress, challenges and opportunities. In: *IJCAI*, 4981–4987. ijcai.org.
- Opitz J, Burst S (2019). Macro F1 and macro F1. CoRR. arXiv preprint: https://arxiv.org/abs/1911.03347.
- Parliament TE (2015). Directive (eu) 2015/849 of the european parliamend and of the council of 20 May 2015. Official Journal of the European Union.
- Raschka S (2018). Model evaluation, model selection, and algorithm selection in machine learning performance estimation: Generalization performance vs. model selection. arXiv preprint.
- Sammut C, Webb GI (2011). Encyclopedia of Machine Learning. Springer Science & Business Media.
- Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009). Computational capabilities of graph neural networks. *IEEE Transactions on Neural Networks*, 20: 81–102.

https://doi.org/10.1109/TNN.2008.2005141

- Shi C (2022). *Heterogeneous Graph Neural Networks*. 351–369. Springer Nature Singapore, Singapore.
- Silva IDG, Correia LHA, Maziero EG (2023). Graph neural networks applied to money laundering detection in intelligent information systems. In: *Proceedings of the XIX Brazilian* Symposium on Information Systems (MXC da Cunha, MF de Souza Júnior, JC Marques, TM de Classe, RD Araújo, eds.), 252–259.
- Tailor SA, Opolka FL, Liò P, Lane ND (2022). Do we need anisotropic graph neural networks? arXiv preprint: https://arxiv.org/abs/2104.01481.
- Thornberry WMM (2021). National defense authorization act for fiscal year 2021. *Public Law*, 116: 283.
- Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017). Graph attention networks. CoRR. arXiv preprint: https://arxiv.org/abs/1710.10903.
- Watanabe S, Bansal A, Hutter F (2023). Ped-anova: Efficiently quantifying hyperparameter importance in arbitrary subspaces. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23 (E Elkind, ed.), 4389–4396.
- Weber M, Chen J, Suzumura T, Pareja A, Ma T, Kanezashi H, et al. (2018). Scalable graph learning for anti-money laundering: A first look. *CoRR*. arXiv preprint: https://arxiv.org/abs/1812.00076.
- Weber M, Domeniconi G, Chen J, Weidele DKI, Bellei C, Robinson T, et al. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *CoRR*. arXiv preprint: https://arxiv.org/abs/1908.02591.
- Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32: 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? *CoRR*. arXiv preprint: https://arxiv.org/abs/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: 4854–4873. https://doi.org/10.1109/TKDE.2020.3045924
- Yang Y, Li D (2020). NENN: Incorporate node and edge features in graph neural networks. In: Proceedings of the 12th Asian Conference on Machine Learning, ACML 2020, 18–20 November 2020, Bangkok, Thailand (SJ Pan, M Sugiyama, eds.), volume 129 of Proceedings of Machine Learning Research, PMLR, 593–608.
- Ying C, Cai T, Luo S, Zheng S, Ke G, He D, et al. (2021). Do transformers really perform badly for graph representation? Advances in Neural Information Processing Systems, 34: 28877–28888.
- You J, Ying R, Leskovec J (2020). Design space for graph neural networks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20 (H Larochelle, M Ranzato, R Hadsell, M-F Balcan, H-T Lin, eds.), 17009–17021. Curran Associates Inc., Red Hook, NY, USA.
- Zhao J, Mostafa H, Galkin M, Bronstein MM, Zhu Z, Tang J (2024). Graphany: A foundation model for node classification on any graph. arXiv preprint: https://arxiv.org/abs/2405.20445.