# GAMLNet: a graph based framework for the detection of money laundering

Julien Schmidt & Dimosthenis Pasadakis & Madan Sathe & Olaf Schenk

***Abstract*—The accuracy of classification algorithms in detecting fraudulent financial activity is critical in assisting human analysts in the task of preventing financial crime. We consider financial transactions in the form of a directed graph, and propose a Graph Neural Network (GNN) model for the identification of money laundering activity. Our method generates a set of structurally aware and statistically significant features for each graph node, and utilizes them as input to the GNN classifier, that comprises of the combination of the layers of two recently proposed message passing architectures. The effectiveness of our approach is demonstrated in experiments with synthetic data that simulate real-world behaviour, and are infused with seven anomalous money laundering topologies. The accuracy of our method is consistently higher than that of other GNNs and tree-based classification methods over datasets of increasing size and increasing imbalance between the fraudulent and benign classes.**

***Index Terms*—Anomaly detection, anti-money laundering, Graph Neural Networks**

## I. Introduction

Anti-money laundering (AML) is the problem of preventing the flow of illicit funds through the financial system. These illicit transactions manifest as anomalies, or rare observations, that differ from the expected transactional behaviour of the data. It is estimated that between 2-5% of the global Gross Domestic Product is laundered each year [1]. For the detection of this fraudulent behavior financial institutions traditionally use rule-based monitoring IT systems to receive alerts on suspicious transactions. Nevertheless, the evergrowing volume of monetary transactions and complexity of fraudulent behavior, makes it increasingly difficult to detect anomalies solely through these expert systems. This necessitates the development of more advanced solutions for AML, through the incorporation of artificial intelligence and the usage of graph based methods [2].

However, due to the sensitive nature of the problem, real-world data is scarcely available, and significant recent efforts exist in the direction of creating realistic artificial instances [3], [4]. The relational nature of these transactional datasets is commonly represented by a directed graph, with its nodes representing the financial entities involved, e.g., companies, individuals, and the edges describing details regarding the transactions, e.g., amount and time. This latent graphical structure offers a global perspective and reveals fraudulent behavior as anomalies in the graph, that will otherwise be unnoticed if transactions are described as independent vectors of characteristics.

Anomaly detection approaches for such graphs are commonly based on neural networks that learn representations on which anomalies are spotted. Generative adversarial networks have also been proposed that employ long short-term memory autoencoders (LSTMs) to separate fraudulent and benign nodes using their historical data [5]. In [6] the authors propose a semi-supervised learning approach based on encoder-decoder GNNs. A key challenge for all aforementioned methods is the highly imbalanced nature of the problem, with money laundering topologies typically accounting for a very small fraction of the data, thus hindering the ability of the algorithms to learn the characteristics of the anomalous nodes [2].

We propose a framework for the detection of money laundering activity that employs a tailored GNN architecture for the classification of fraudulent accounts. Initially, we generate a set of features that are both structurally aware and statistically significant, and are specifically selected in order to provide distinguishable characteristics for each account. We then consider the latest research conducted in Message Passing Neural Networks (MPNNs) [7] , and introduce the Graph Anti-Money Laundering Network (GAMLNet). The proposed architecture is designed to maximize the discovery of similar fraudulent subgraph structures, and to identify the statistical patterns present in the features of a node and those of its neighbors. Our GNN model comprises of a combination of Graph Isomorphism Network (GIN) [8] and GraphSAGE [9] convolutional layers.

In the remainder of this text we discuss the generated features, and the proposed GNN model in Section II. Then in Section III we present our experimental setting, and our numerical results in the created artificial instances. Last, in Section IV, we draw conclusions from this work.

## II. A structure-informed GNN for anomaly detection

We initially compute in Subsection II-A a set of meaningful node features from the input financial network, and extract node-specific structural data and monetary transactional statistics. The second segment of our approach is the model training and the evaluation procedure presented in Subsection II-B.

The financial networks considered are multigraphs $\mathcal{G}(V, E, \mathbf{W})$, comprising of $V$ nodes representing bank accounts, and $E$ directed edges capturing the transactions between them. Nodes are assigned account balances, and each edge a weight $w_{ij}$ with the transaction amount from node $i$ to

Julien Schmidt, Dimosthenis Pasadakis, and Olaf Schenk are with the Advanced Computing Laboratory at the Institute of Computing, Università della Svizzera italiana (USI), Lugano, Switzerland. email: {julien.schmidt, dimosthenis.pasadakis, olaf.schenk}@usi.ch. Madan Sathe, Partner, Forensics, Ernst & Young AG.

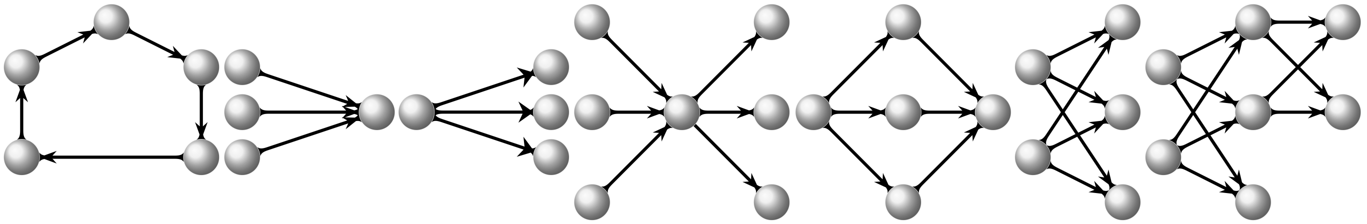


**Fig. 1:** Visualization of the anomalous topologies that are embedded in the data. Starting from the left: cycle, fan-in, fan-out, gather-scatter, scatter-gather, bipartite, and stacked bipartite.

node $j$, while the initial and final balance of each node are also known. The generated feature set is computed from these transaction amounts, the account balances, and the information encoded in the adjacency matrix $\mathbf{W}$. Embedded in this financial network are known money laundering topologies following the same transaction and temporal rules as benign accounts. The infused anomalous topologies are visualized in Figure 1, and the process of generating these synthetic graphs is summarized in Section III.

### A. Feature generation

The node feature generation process provides the model with a total of $l$ latent statistics that will be utilized in the learning and classification decisions. We initially consider the structurally informative in-, out-, and total degree $\mathbf{d}_i = \sum_{j=1}^{n} w_{ij}$ in both the multigraph and the simple graph setting, where the multiple edges connecting nodes $i, j$ are aggregated in a single directed edge. We then compute the degree frequency of a node $i$ as $\mathbf{q}_i = \mathbf{d}_i^{\mathrm{s}}/\mathbf{d}_i^{\mathrm{m}}$, where $\mathbf{d}^{\mathrm{s}}$ is the total degree in the simple graph, and $\mathbf{d}^{\mathrm{m}}$ the total degree in the multigraph. The next structural features are based on [10], and are the geometric average of weights of a node, $\mathrm{GAW}_i = \left(\prod_{j\in N(i)} w_i\right)^{(\lceil r*\mathbf{d}_i^{\mathrm{out}}\rceil)^{-1}}$, where $N(i)$ is the neighborhood of node $i$, $w_i$ the arranged in ascending order incoming and outgoing weights of node $i$, $\mathbf{d}_i^{\mathrm{out}}$ the simple graph out-degree, and $r = 1$ is a scalar parameter controlling the percentage of considered out-degrees. The GAW is additionally computed for the heaviest 10% and 20% of the connected edges of each node, with $r$ being 0.1 and 0.2, respectively. As a last structural statistic, we include the z-score of the node degree of the simple graph based on its mean and standard deviation, in order to capture when the degree is larger or smaller than expected at random.

Subsequently, we record incoming and outgoing node transaction statistics with the minimum and maximum transaction amounts, along with their means and standard deviations, and with the aggregated totals of these transactions. Node features are also generated based on the minimum, maximum, starting, and ending balance of each node, and based on the maximum balance shift, i.e., the minimum account balance subtracted from the maximum one, and on the limit balance shift, i.e., the ending balance subtracted from the starting one.

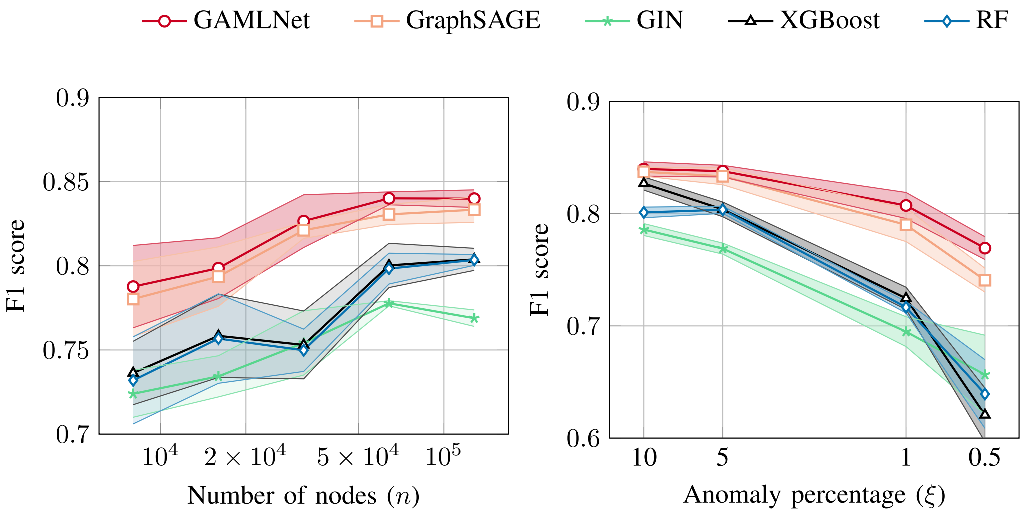
### B. Model description

We propose GAMLNet, a GNN architecture that combines the strengths of both GIN [8] and GraphSAGE [9] for node classification. The motivation of our model is based on the nature of the directed graphs under question, i.e., monetary transaction networks with scattered anomalous topologies that exhibit similar structure. The advantage of GIN over other GNN variants lies in its ability to discern structural isomorphisms between subgraphs. Using as input only a limited set of structural features of the graph, and following layers of propagation and aggregation, it maps nodes with identical anomalous neighborhood structure to the same embedding space, thus promoting their discovery. These new latent node embeddings are concatenated to the full list of generated node statistics, outlined in Subsection II-A, and form an enhanced, structurally informed feature set. Subsequently, GraphSAGE succeeds at learning the statistical nature of anomalies in the graph, with its mean aggregator function capable of learning feature distributions. In a feature rich environment such as the one we provide it with, it excels at identifying anomalous nodes based on their features, and on those of their neighbors.

**Algorithm 1** GAMLNet forward pass algorithm

```
Input:   Graph $\mathcal{G}(V, E, \mathbf{W})$, feature matrix $\mathbf{X} \in \mathbb{R}^{|V|\times l}$,
         degree structural features $s_{\mathrm{idx}}$,
         number of layers $K_1, K_2$
Output:  $\hat{\mathbf{y}} \in \mathbb{R}^{|V|\times 1}$ predicted node classification
$\overline{\mathbf{h}}^{(0)} = \mathbf{X}[:, s_{\mathrm{idx}}]$
for $k = 1$ to $K_1$ do
    for $i \in V$ do
        $\overline{\mathbf{h}}_i^{(k)} =$
            $\tanh\left(\mathrm{MLP}^{(k)}((1+\epsilon^{(k)})\overline{\mathbf{h}}_i^{(k-1)} + \sum_{j\in N(i)} \overline{\mathbf{h}}_j^{(k-1)})\right)$
$\mathbf{h}^{(0)} = \mathrm{concat}(\mathbf{X}, \phi_1(\mathbf{h}^{(K_1)}))$
for $k = 1$ to $K_2$ do
    for $i \in V$ do
        $\mathbf{h}_i^{(k)} =$
            $\mathrm{ReLU}\left(\mathbf{\Theta}_1^{(k)}\mathbf{h}_i^{(k-1)} + \mathbf{\Theta}_2^{(k)} \cdot \operatorname*{mean}_{j\in N(i)} \mathbf{h}_j^{(k-1)}\right)$
$\hat{\mathbf{y}} = \mathrm{argmax}(\phi_2(\mathbf{h}^{(K_2)}))$
return $\hat{\mathbf{y}}$
```

The GAMLNet architecture is outlined in Algorithm 1.



**(a)** **(b)**

**Fig. 2:** Accuracy in the retrieval of fraudulent nodes for (a) datasets of increasing node size $n$, and (b) datasets of decreasing anomaly percentage $\xi$.

Initially, the reduced node feature matrix $\overline{\mathbf{h}}^{(0)}$, containing only the in- and out-degree features (Line 1), is used as input to the $K_1$ layers of the GIN model (Line 4). This involves a multilayer perceptron $\mathrm{MLP}^{(k)}$ for the $\overline{\mathbf{h}}_i$ node representation vectors, where $N(i)$ is again the neighborhood of node $i$, and a learnable parameter $\epsilon$ that scales the weight of each node feature vector at each layer. The output embedding of GIN is passed through a linear layer $\phi_1$, and concatenated with the full feature matrix $\mathbf{X}$ (Line 5) producing a new node representation $\mathbf{h}^{(0)}$, to then serve as input to the $K_2$ layers of the GraphSAGE model (Line 8). This involves a ReLU activation function with two learnable parameter matrices $\mathbf{\Theta}_1, \mathbf{\Theta}_2$, and the representation vectors $\mathbf{h}_i$ of node $i$, and $\mathbf{h}_j$ of its neighborhood. Last, the output $\mathbf{h}^{(K_2)}$ is passed though a linear layer $\phi_2$ to classify the nodes into fraudulent or bening (Line 10).

Our model is trained using an Adam optimizer [11] and a weighted binary cross entropy loss, where the majority class weight $\beta$ is scaled between 0 and 1. Optimal model hyperparemeters are selected via a grid search, and the model is built and trained using PyTorch and PyTorchGeometric [12], using the implementations of the GIN and GraphSAGE convolution layers under the message passing paradigm.

## III. Numerical results

In order to demonstrate the effectiveness of the GAMLNet model in AML tasks, in Subsection III-A we report the setup of our experiments, and in III-B the accuracy of classification assignments for datasets of increasing size and decreasing anomaly percentage. Then, in Subsection III-C we focus on its ability to identify the different fraudulent topologies.

Our source code is available at https://github.com/schmidtjulien/GAMLnet, and the synthetic financial datasets that were used in our experiments can be downloaded from https://drive.switch.ch/index.php/s/Sc5o5B7ASni9DHW.

### *A. Experimental setup*

We create synthetic test instances using the AMLSim multi-agent simulator [3], that generates synthetic banking transaction data together with a set of known money laundering topologies. These anomalies include the seven structural patterns, i.e., cycle, fan-in, fan-out, gather-scatter, scatter-gather, bipartite, and stacked bipartite, that are illustrated in Figure 1, and have a varying number of participating nodes. We consider cases with a fixed percentage of anomalies $\xi = 5\%$ and an increasing number of nodes $n \in \{8, 16, 32, 64, 128\} \times 10^3$. Then, for the case with $n = 128 \times 10^3$, we create datasets with a decreasing percentage of anomalies $\xi \in \{10, 5, 1, 0.5\}\%$. The classification accuracy of our approach is compared against (1) GraphSAGE [9], (2) GIN [8], and the decision-tree based methods, (3) Random Forests (RF) [13], and (4) XGBoost [14]. For all methods under consideration, the same set of node features was used, and a grid search was conducted to identify the optimal hyperparameter setting. We report the mean and standard deviation after 10 runs, and consider the train, validation, and test splits, being 50%, 15%, and 35% respectively, and randomly shuffled across the runs.

Results concerning the accuracy of classification are reported in terms of F1 score and in terms of the percentage of true positives, captured in the ACC metric [15]. For both F1 and ACC a score of 1 suggests a perfect classification accuracy, while smaller values suggest worse recovery success.

### *B. Anomaly detection accuracy*

We present in Figure 2 comparative results regarding the accuracy of our method in detecting accounts involved in money laundering activities. In Figure 2a, for datasets of increasing size, GAMLNet attains the best accuracy in terms of F1-score for all cases with $\mathrm{F1} = 0.79, 0.80, 0.82, 0.84$, and $0.84$, respectively, with the sole other method following closely being GraphSAGE, and reporting on average 1.2% worse recovery

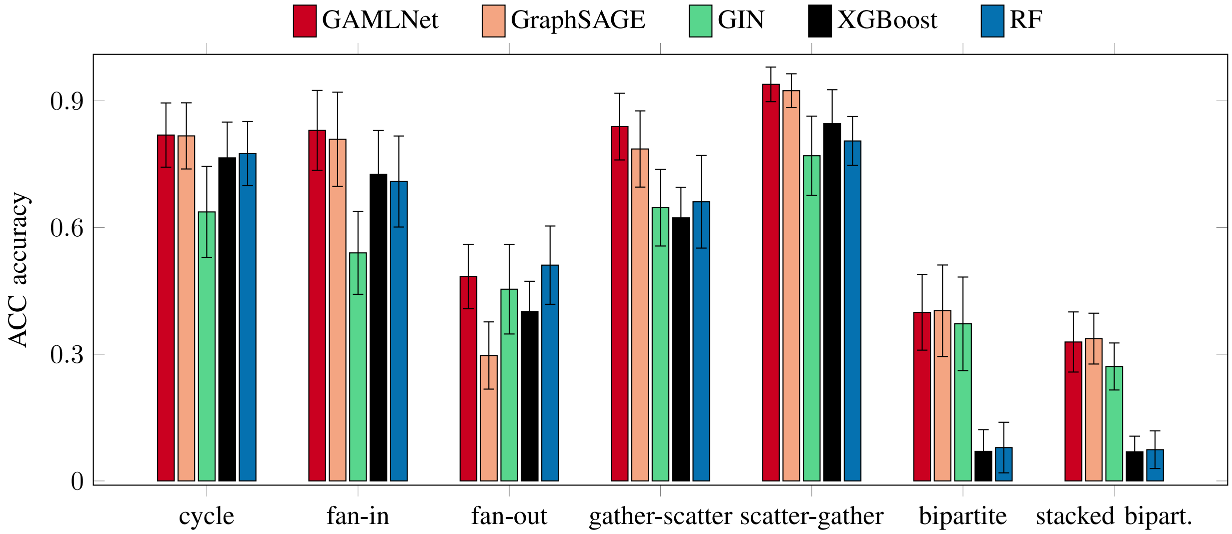


**Fig. 3:** True positive recovery rate of the different fraudulent topologies for all the methods under consideration for the dataset with size $n = 128 \times 10^3$, and anomaly percentage $\xi = 0.5$.

rates. The higher accuracy for larger datasets is anticipated due to the larger size of the training set the algorithms have access to. In Figure 2b we investigate the performance for test cases with a fixed number of nodes $n = 128 \times 10^3$, and a decreasing anomaly percentage. The classification accuracy decreases for smaller percentages of anomalies, as the datasets become more imbalanced. GAMLNet reports the best accuracy in all $\xi$ regimes with $\mathrm{F1} = 0.84, 0.84, 0.80,$ and $0.77$, respectively. The largest benefits over all other methods are notably observed for the most imbalanced case with $\xi = 0.5$, which is an anomaly ratio that accurately reflects real-world transaction networks [2]. There GAMLNet attains $\mathrm{F1} = 0.77$, and the second best GraphSAGE $\mathrm{F1} = 0.74$.

### C. *Topology identification accuracy*

We analyze further the rate of accurate retrieval for each of the anomalous topologies that are embedded in the test graphs. For the case with $n = 128 \times 10^3$ and $\xi = 0.5\%$, we present in Figure 3 the classification accuracy for the considered anomalies. GAMLNet achieves an ACC score of 0.82 (cycle), 0.83 (fan-in), 0.48 (fan-out), 0.84 (gather-scatter), 0.94 (scatter-gather), 0.40 (bipartite), and 0.33 (stacked bipartite). In contrast to every other model considered, GAMLNet reports either the top accuracy (cycle, fan-in, gather-scatter, and scatter-gather topologies), or is less than $1\%$ worse than the top method (RF for fan-out, GraphSAGE for bipartite and stacked bipartite). The efficacy of our model to combine the benefits of the GIN and the GraphSAGE architectures is thus highlighted by its consistent capability to maintain a top-tier classification score, regardless of the topology in question.

## IV. Conclusion & Outlook

In this work, we developed an anomaly detection method for financial data that are expressed in the form of a directed graph. We generated a set of structurally informed and statistically significant features for each node of the graph, and used this set as input to our GAMLNet model that combines layers from two state-of-the-art message passing architectures. Our model detects isomorphisms between graph substructures, and excels in taking advantage of the rich feature space that we supply it with. This enabled the accurate identification of accounts that were involved in money laundering schemes, manifested as anomalous topologies on the graph. Increased accuracy gains were exhibited over the competing methods for highly imbalanced datasets, that are accurate representations of real-world transaction networks. We intend to further explore the potential gains of estimating additional node and edge features that will augment the classification accuracy of our approach.

## Acknowledgment

We would like to acknowledge the financial support of the joint DFG - 470857344 and SNSF - 204817 project, and the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). NHR@FAU funding is provided by federal and Bavarian state authorities and by the German Research Foundation (DFG) - 440719683. This work was also supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID 32191.

## References

[1] United Nations Office on Drugs and Crime, "Money laundering," 2021. [Online]. Available: www.unodc.org/unodc/en/money-laundering/overview.html

[2] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12 012–12 038, 2023.

[3] T. Suzumura and H. Kanezashi, "Anti-Money Laundering Datasets: InPlusLab anti-money laundering datasets," http://github.com/IBM/AMLSim/, 2021.

[4] E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, “Realistic synthetic financial transactions for anti-money laundering models,” in Advances in Neural Information Processing Systems, vol. 36. Curran Associates, Inc., 2023, pp. 29 851–29 874.

[5] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, “One-class adversarial nets for fraud detection,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, Jul. 2019, pp. 1286–1293.

[6] M. Cardoso, P. Saleiro, and P. Bizarro, “Laundrograph: Self-supervised graph representation learning for anti-money laundering,” in Proceedings of the Third ACM International Conference on AI in Finance, ser. ICAIF ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 130–138.

[7] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in Proceedings of the 34th International Conference on Machine Learning - Volume 70, ser. ICML’17, 2017, p. 1263–1272.

[8] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?” in International Conference on Learning Representations, 2019.

[9] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1025–1035.

[10] A. Elliott, M. Cucuringu, M. M. Luaces, P. Reidy, and G. Reinert, “Anomaly detection in networks with application to financial transaction networks,” 2019. [Online]. Available: https://arxiv.org/abs/1901.00402

[11] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations (ICLR), San Diega, CA, USA, 2015.

[12] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[13] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, p. 5–32, 2001.

[14] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794.

[15] A. Tharwat, “Classification assessment methods,” Applied Computing and Informatics, vol. 17, no. 1, p. 168–192, Jul. 2020.