Synthetic Graph Data Generation to Mitigate Class Imbalance in Money Laundering Detection

Shahram Ghahremani*, Uyen Trang Nguyen†
Department of Electrical Engineering & Computer Science, York University, Toronto, Canada
Email: *shg@yorku.ca, †unguyen@yorku.ca

***Abstract*—We present a Conditional Variational Autoencoder (CVAE) designed to generate realistic money laundering (ML) patterns in the form of transaction subgraphs. The synthesized patterns are employed to balance class distribution in ML datasets. By jointly modeling relational structures and transactional attributes, the CVAE produces graph-based samples that capture established ML typologies. Experiments on the AMLWorld dataset demonstrate that our method outperforms conventional oversampling methods such as SMOTE and ADASYN, as well as generative models like GAN, in downstream ML detection tasks. Notably, graph neural network (GNN) models trained on CVAE-augmented data achieve significantly higher recall while reducing false alert rates. To the best of our knowledge, our proposed approach is the first that generates ML data at the pattern level, rather than at the level of individual transactions.**

***Index Terms*—Money laundering detection, synthetic data generation, transaction graphs, conditional variational autoencoder, graph neural networks, class imbalance**

I. Introduction

Money laundering (ML) is the process of disguising illicit funds as legitimate assets, often involving proceeds from crimes such as tax evasion, trafficking, illegal gambling, and terrorist financing. Financial institutions must implement anti-money laundering (AML) measures such as customer verification, risk assessment, transaction monitoring, and reporting suspicious activities. Non-compliance has resulted in large regulatory fines worldwide [1]. This work focuses on *transaction monitoring* for ML detection.

Most AML systems rely on rule-based thresholds [2], such as flagging cash withdrawals above $10,000 within 24 hours. These rules are conservative and lead to extremely high false alert rates (FAR), typically 95%–98% [3]. Every alert requires manual review, making the process costly. Machine learning can detect more subtle ML behaviors [4], but extreme class imbalance remains a major obstacle. In public datasets such as SynthAML [5], SAML-D [6], and AMLworld [7], illicit transactions account for only 0.05%–0.125% of all transactions.

Two main approaches address class imbalance: (a) *data-level* methods such as undersampling or oversampling [8], [9]; and (b) *algorithm-level* methods such as cost-sensitive learning or anomaly detection [7], [10]. Given the extreme imbalance (1:800 to 1:2,000), undersampling or cost-sensitive learning alone is insufficient. In this paper, we focus on oversampling.

A money laundering case usually involves several accounts and multiple linked transactions. Suzumura and Kanezashi [11] identify eight common ML patterns, shown in Figure 1. However, popular oversampling methods such as SMOTE [12] and ADASYN [13] treat each transaction independently and cannot capture the relational or sequential structures of laundering patterns. Table I illustrates examples from the AMLworld dataset: SMOTE is unaware of these structures and generates isolated transactions that do not form meaningful ML patterns.

After identifying the limitations of SMOTE and ADASYN, which operate at the transaction level, we adopt a different approach: generating complete ML patterns as transaction subgraphs. In our previous work [14], we proposed a GAN-based model that generates such realistic graph patterns, substantially improving money laundering detection performance compared to transaction-level oversampling methods such as SMOTE and ADASYN. However, our preliminary results indicate that augmenting the dataset with a CVAE achieves even greater improvements in detection performance.

The main contribution of this paper is the development of a generative model using CVAE that learns money laundering (ML) typologies [11] and generates realistic ML patterns by jointly modeling graph structures and transaction attributes. We evaluate the CVAE on the AMLWorld dataset, comparing its performance with SMOTE, ADASYN, and CGAN in terms of downstream ML detection across multiple graph neural network (GNN) models. Experimental results show that the CVAE consistently outperforms these baselines, with GNNs trained on CVAE-augmented data achieving higher recall and significantly lower false alert rates (FAR) than those trained on data augmented by SMOTE, ADASYN, or CGAN.

The remainder of this paper is organized as follows. Section II reviews related work on class imbalance handling in anti-money laundering (AML) systems. Section III presents the proposed CVAE-based synthetic data generation model. Section IV describes the experimental setup, including datasets, baselines, and evaluation metrics. Section V discusses the results of our experiments, comparing CVAE with other oversampling techniques. Finally, Section VI concludes the paper and outlines directions for future research.

II. Related Work

Most research in AML has focused on money laundering detection [15]–[17]. Our focus, however, is the class imbalance problem of AML data, which significantly affects classification performance of ML detection models. Prior surveys have reviewed class imbalance solutions across domains such as medical diagnosis, fraud detection, and image recognition [18]–[21]. In AML, illicit transactions are extremely rare,

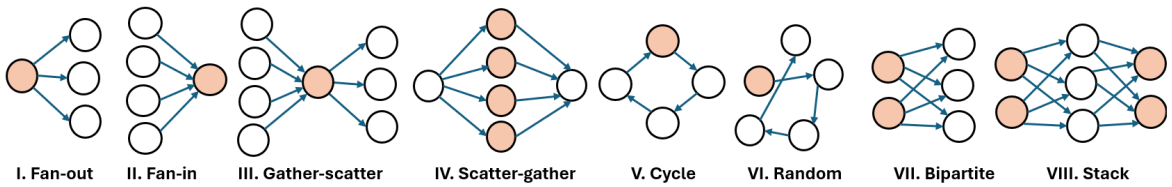
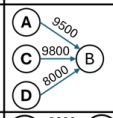
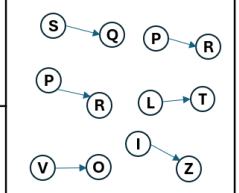
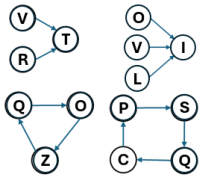
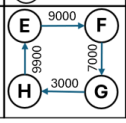


Fig. 1: Common money laundering patterns [7]. Shaded nodes are key or central nodes in each ML structure.

TABLE I: Examples of transaction-based vs. pattern-based synthetic data generation

Money Laundering Transactions	Pattern	Type	Synthetic Money Laundering Transaction Generator	
			SMOTE	Pattern-based
Date, Sender, Amount, Receiver 2022/10/20, A, USD 9500, B 2022/10/22, C, EUR 9800, B 2022/10/23, D, USD 8000, B		Fan-in		
2022/11/20, E, USD 9000, F 2022/12/22, F, EUR 7000, G 2022/12/23, G, CAD 3000, H 2022/12/27, H, CAD 9900, F		Cycle		

often representing less than 0.1% of the data, which has been identified as a major challenge [17], [22].

To mitigate class imbalance, prior work in AML uses either *data-level* [8], [9]or *algorithm-level* methods [7], [10]. Data-level methods modify the class distribution of training data. Undersampling removes majority class samples using random or near-miss techniques [8], [12], [23], while oversampling increases minority class representation using methods like SMOTE [24], which interpolates between neighbors, or ADASYN [13], which generates more samples in sparse minority class regions. Some prior works have combined undersampling and oversampling to balance AML datasets [9], [12].

Algorithm-level methods instead adjust the learning process. These include the use of class-weighted loss functions to penalize misclassifications of the minority class [7], [25]. Focal loss is used in DTPAN [26] to focus on hard-to-classify samples. Other studies reframe AML detection as an anomaly detection problem to avoid explicit resampling. For example, El-Kilany et al. [27] and Baltoiu et al. [28] use one-class SVM, while Tertychnyi et al. [10] use isolation forests to detect anomalous behavior in transaction graphs.

In this paper, we focus on oversampling to mitigate class imbalance in AML data. While SMOTE and ADASYN are effective for numeral or tabular data, they are not capable of replicating pattern structures in graph data like AML data. To address this gap, we propose a deep learning model based on a conditional variational autoencoder (CVAE) that synthesizes complete graph patterns representing real-world money laundering typologies. While CVAE has been used for generating synthetic data [29], [30], to our knowledge, this work is the first that generates graph patterns tailored to money laundering detection and studies their impact on ML detection models.

III. The Proposed Model

Figure 2 provides an overview of the proposed model and the downstream task of money laundering (ML) detection. The following sub-sections detail each step of the pipeline.

A. Pre-processing

Since our goal is to generate ML patterns, the first step is to extract these patterns (shown in Figure 1) from the AMLWorld dataset. Each ML case is represented by a graph (see two example graphs in Table I). For each extracted pattern graph, we constructed an *adjacency matrix A* and a *feature matrix F*, each of size $v \times v$, assuming the pattern graph has v nodes (vertices). In the adjacency matrix A, $A_{i,j} = 1$ indicates that there is a directed edge (a transaction) from node i to node j, and $A_{i,j} = 0$ otherwise. The feature matrix F contains the attributes of the transactions (the edges) such as transaction amount, currency type, and transfer method.

Because the pattern graphs have different sizes, the resulting square matrices have different sizes. Therefore, we padded all matrices with zeros to match the size of the largest pattern graph in the dataset, 45×45. This way, all matrices input into the CVAE model have the same size. Each pair of matrices (A, F) is labeled with the corresponding pattern (fan-in, fan-out, gather-scatter, scatter-gather, cycle, random, bipartite or stack). These matrices and their labels are then fed into a conditional variational autoencoder to learn the underlying distribution of each pattern type.

B. Conditional Variational Autoencoder (CVAE)

Conditional variational autoencoders [31] are generative models designed to learn complex data distributions and generate new samples based on a given condition. Unlike standard VAEs [32], CVAE incorporate *conditional variables* to guide the generation process, making them well-suited for

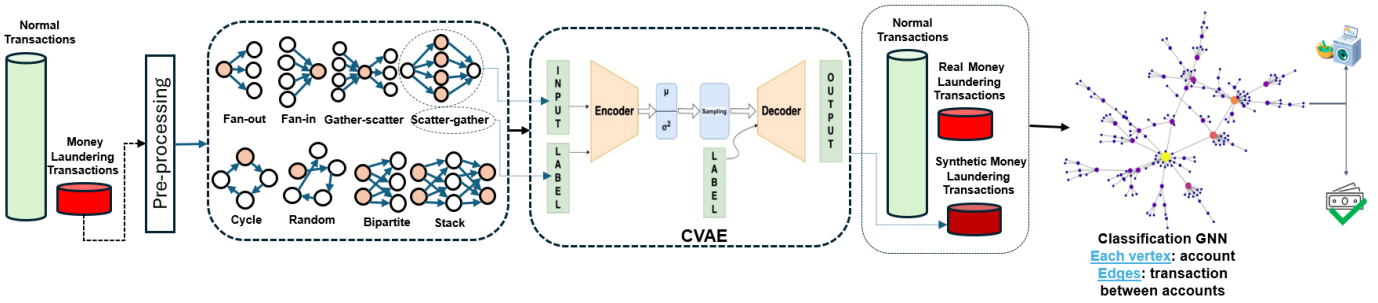


Fig. 2: An overview of CVAE and the downstream task of money laundering detection

structured data like ML patterns. A CVAE consists of two main components:

1. *Encoder:* The encoder maps the input pattern (A, F) into a latent space representation z, conditioned on an auxiliary variable c (e.g., the pattern type). The encoding process is modeled as:

$$q_\phi(z|A, F, c) = \mathcal{N}(\mu, \sigma^2),$$

where μ and σ are learned parameters representing the mean and variance of the latent distribution.

2. *Decoder:* The decoder reconstructs the original input pattern (A, F) from the latent variable z and condition c: $p_\theta(A, F|z, c)$

The reconstruction loss (e.g., binary cross-entropy for adjacency matrices and mean squared error for feature matrices) ensures that the generated patterns resemble real ML patterns.

The training objective is to minimize the *evidence lower bound (ELBO)*:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|A,F,c)}[\log p_\theta(A, F|z, c)] - D_{KL}(q_\phi(z|A, F, c)||p(z)),$$

where the first term represents the reconstruction loss, and the second term is the *Kullback-Leibler (KL) divergence*, which regularizes the latent space to follow a prior distribution (typically a standard normal distribution).

Once trained, the CVAE can generate *new synthetic ML pattern graphs* by sampling z from the latent space and decoding it using a given condition c. This enables the model to create diverse pattern graphs representative of real-world ML behavior while preserving statistical properties of real data.

C. Data Augmentation and Transaction Classification

For each dataset D_i used in this study (described in Section IV-B and Table II), the CVAE learned from the pattern graphs in the minority class of D_i, and generated synthetic ML samples for each pattern type. The synthetic ML samples were then added to the minority class of D_i to obtain a more balanced dataset. Each dataset (original or augmented) was then converted into a (large) global graph, in which nodes and directed edges represent accounts and transactions between accounts, respectively. Three graph neural network (GNN) models (described in Section IV-C) were trained on the datasets to classify transactions as licit or suspicious of ML.

IV. Experiment Settings

This section describes the baseline synthetic data generation methods to compare against CVAE, datasets used in the experiments, and evaluation metrics.

A. Baseline Synthetic Data Generation Methods

To evaluate the quality of our synthetic data and the effectiveness of pattern-based generation versus individual transaction generation, we compare our approach with three widely used methods: SMOTE [33], ADASYN [34], and CGAN [14]. SMOTE and ADASYN create synthetic samples by interpolating between existing transactions, while CGAN, like our proposed CVAE model, learns to generate structured data patterns. SMOTE and ADASYN are the most popular oversampling techniques in financial fraud and money laundering detection [7], [13], [24], [33]–[37]. CGAN [14] is the closest competitor to CVAE due to its ability to synthesize structured data patterns. Collectively, these methods provide a diverse benchmark for evaluating CVAE.

B. Datasets

In this paper, we use AMLworld [7], a synthetic dataset developed by IBM that models a virtual financial ecosystem and the full money laundering cycle. AMLworld provides two variants based on the percentage of illicit transactions: high illicit ratio (HI) and low illicit ratio (LI).

In this paper, we evaluate our model using only the **Small** AMLworld datasets. A 60–20–20 temporal split is used for training, validation, and testing, where transactions are divided chronologically.

TABLE II: Sizes of the original and augmented Small AMLworld datasets. HI: high illicit ratio; LI: low illicit ratio; M: million; K: thousand.

Statistics	**HI**		**LI**	
	Original	Augmented	Original	Augmented
Total Transactions	5.1M	5.2M	6.9M	7.0M
Laundering Transactions	5.2K	94.8K	3.6K	65.3K
Minority-to-Majority Ratio	1:981	1:54	1:1942	1:107

C. Money Laundering Detection Models

To assess the impact of CVAE-generated data on downstream classification performance, we follow the experimental setup used for the AMLworld benchmark [7]. Specifically, we use three GNN architectures for the task of ML detection (which are also used in [7]): Graph Isomorphism Network (GIN) [38]; GIN with edge updates (GIN+EU) [39], which extends GIN by incorporating edge update mechanisms during training; and GNN with principal neighborhood aggregation (PNA) [40], which aggregates node neighborhoods using multiple statistics (mean, max, standard deviation) and applies degree-scalers to improve generalization across graphs. Details about the architecture, hyperparameters, and implementation of these GNN models can be found in [7].

D. Money Laundering Detection Performance Metrics

Common metrics like accuracy and F1-score are not suitable for AML tasks due to the extreme class imbalance of AML data. For example, in datasets with a 1:1000 positive-to-negative sample ratio, a naïve model that always predicts the negative class would still achieve an accuracy of 99.9%. F1-scores favor a balance of recall and precision. However, in AML, recall (the ability to identify true ML transactions) takes priority over precision to avoid regulatory penalties caused by missed ML cases.

In this context, we use *recall* (detection rate) and the *false alert rate* (FAR) as metrics to evaluate the classification performance of money laundering detection models. The objective is to achieve a target high recall value, e.g., 95%, while lowering the FAR. (While precision is widely used in machine learning literature, the FAR is more common in industry and helps quantify the operational cost of manually investigating alerts.)

V. Experimental Results

In this section, we evaluate money laundering detection performance of GNN models trained on synthetic data generated by SMOTE, ADASYN, CGAN and CVAE. All results are reported with *standard deviations* to ensure statistical reliability.

For ML detection, we use three GNN models described in Section IV-C: GIN, GIN+EU and PNA, trained on the original datasets (natural distributions) and the augmented datasets whose class distributions are listed in Table II. The test sets in all experiments retain the natural class distributions.

The metrics to evaluate ML detection performance are recall and false alert rate (FAR = 1 - precision), as explained in Section IV-C. When the intersection-over-union (IoU) threshold is calibrated, higher recall will lead to a higher FAR (or, equivalently, lower precision), and vice versa. To fairly compare the AML models, a metric should be fixed so that we can observe the other metric changing. For instance, given the same target recall (e.g., 95%), a model that offers a lower FAR is the better model. Conversely, given the same target FAR (e.g., 60%), a model that yields higher recall is the better model. In this paper, we present the results for the former case.

In our experiments, we use a recall value of 95%, following the industry practice of prioritizing recall over FAR (or precision). We calibrated the intersection-over-union (IoU) threshold to obtain the desired recall value for each GNN model. Given the same recall, a model that yields a lower FAR (i.e., incurring lower operational costs) is considered the higher performer. We also ran experiments with a threshold of 0.5, which is used in many machine learning tasks, including the ML detection models used to evaluate the AMLworld dataset in [7], just for comparison purposes.

The results in terms of recall and false alert rates (FAR) are presented in Table III and IV for the HI and LI dataset, respectively. The top three results of each row are highlighted in green color, and the darker the shade, the higher the performance.

Following are the findings from the results in **Table III** for the **HI dataset** and a **target recall of 95%**. First, all four data augmentation methods give higher classification performance than the original data (natural distributions). For instance, the FARs of the original data are very high, ranging from 93.50% to 93.80%. The FARs of the models trained with augmented data are much lower.

Second, CVAE-augmented training data yield the best performance, i.e., giving the lowest FARs, ranging from 57.90% to 65.20%. The second best performer is CGAN, with the FAR ranges from 63.70% to 70.85%. The FAR of the third best performer, SMOTE, ranges from 73.10% to 78.10%. (These FARs are significantly lower than the current industry standard of around 95%.) These results show that pattern-level data generated by CVAE and CGAN allow the AML models to learn more effectively than isolated transactions synthesized by SMOTE and ADASYN.

Third, PNA is the best classifier, followed by GIN+EU, based on the FARs they yield, thanks to these models' awareness of graph structures.

When using the **standard threshold of 0.5**, it takes more effort to compare the classification performance of the different classifiers and data augmentation methods, because both the recall and FAR now vary. A closer examination shows that, for each classifier, CVAE-augmented data yield the highest recall among the data augmentation methods, and the original data yield the lowest recall. However, all recall values in this set of experiments are very low, below the industry acceptable level. For example, the highest recall is 67.57%, produced by CVAE-augmented data and PNA classifier. This implies that 32.43% of ML transactions were misclassified in this case!

Table IV presents the results for the **LI (low illicit transaction ratio)** dataset. The overall trends match those observed in the HI datasets, with one exception: **GIN+EU outperforms PNA**. This suggests that when the minority class is extremely low, incorporating edge-update mechanisms (GIN+EU) is more effective than aggregating node neighborhoods using multiple statistics (PNA).

We compare the results in Table III (high illicit ratio) with those in Table IV (low illicit ratio). For example, the FAR given by the combination {CVAE, 95% recall, PNA} in Table III

TABLE III: Recall and FARs in percentages for the **HI dataset**. The results highlighted in yellow are taken from [7].

Model	Natural Distribution		CVAE		CGAN		SMOTE		ADASYN	
	Recall	FAR	Recall	FAR	Recall	FAR	Recall	FAR	Recall	FAR
AML Threshold (High Recall of Approx. 95%)										
GIN	95.13±0.88	93.80±6.61	95.45±0.67	65.20±3.83	95.32±0.35	70.85±8.27	95.18±0.71	78.10±4.65	95.05±0.91	82.50±4.92
GIN + EU	95.12±0.74	93.70±7.11	95.48±0.59	61.10±3.04	95.35±0.84	67.50±7.96	95.22±0.22	74.60±4.50	95.18±0.43	79.20±4.83
PNA	95.08±0.65	93.50±7.75	95.50±0.09	57.90±2.48	95.38±0.11	63.70±7.25	95.20±0.38	73.10±3.94	95.30±0.77	70.80±3.79
Standard Threshold of 0.5										
GIN	38.16±5.92	72.60±7.98	58.24±3.12	57.40±4.25	53.78±4.47	62.90±6.21	33.10±3.21	68.45±3.09	30.50±3.33	72.50±3.18
GIN + EU	55.41±5.96	57.71±10.69	67.41±3.27	35.56±4.30	66.13±4.68	42.10±6.13	34.11±3.45	50.25±3.26	31.03±3.31	54.50±3.20
PNA	53.15±2.26	41.52±10.67	67.57±1.85	17.13±4.11	63.44±2.14	23.79±5.77	35.00±1.91	32.75±1.85	36.41±2.00	30.11±1.78

is 57.90%, lower than the FAR 74.40% given by the same combination in Table IV. That is, the dataset with a higher number of money laundering samples (high illicit ratio) yields higher performance than that with fewer money laundering samples, as expected, because a classifier can learn more effectively with more samples. This finding is consistent across all datasets, classifiers and recall targets/thresholds.

VI. Conclusion

We propose CVAE, a generative model that addresses a critical shortcoming in current ML detection systems: their inability to generate structurally faithful minority-class samples. Unlike traditional oversampling methods that treat transactions as independent and identically distributed samples, CVAE synthesizes graph-structured transaction patterns that more accurately reflect real-world ML behavior. Our preliminary results demonstrate that CVAE outperforms conventional oversampling methods in downstream detection performance. In future work, we aim to extend this study by investigating the statistical and structural quality of the samples generated by various oversampling methods (e.g., SMOTE, ADASYN, CGAN, CVAE) in greater depth. We will evaluate the proposed CVAE model using more datasets. We will also study combinations of oversampling, undersampling and algorithm-level methods to find the most effective solutions to class imbalance.

References

[1] O. Husain, "13 biggest aml fines ($500 million plus)," May 2024, accessed: 2024-09-05. [Online]. Available: https://www.enzuzo.com/blog/biggest-aml-fines

[2] S. Ross and M. Hannan, "Money Laundering Regulation and Risk-Based Decision-Making," *Journal of Money Laundering Control*, vol. 10, no. 1, pp. 106–115, 2007.

[3] J. Gerlings and I. Constantiou, "Machine learning in transaction monitoring: The prospect of xai," in *Proceedings of the 56th Hawaii International Conference on System Sciences*, T. X. Bui, Ed. Honolulu, United States: Hawaii International Conference on System Sciences (HICSS), 2023, pp. 3474–3483. [Online]. Available: https://doi.org/10.10125/103058

[4] A. Naser Eddin, J. Bono, D. Aparício, D. Polido, J. T. Ascensão, P. Bizarro, and P. Ribeiro, "Anti-money laundering alert optimization using machine learning with graphs," *arXiv preprint arXiv:2112.07508*, 2021. [Online]. Available: https://arxiv.org/abs/2112.07508

[5] R. I. T. Jensen, J. Ferwerda, K. S. Jørgensen, E. R. Jensen, M. Borg, M. P. Krogh, J. B. Jensen, and A. Iosifidis, "A synthetic data set to benchmark anti-money laundering methods," *Scientific data*, vol. 10, no. 1, p. 661, 2023. [Online]. Available: https://www.nature.com/articles/s41597-023-02569-2

[6] B. Oztas, D. Cetinkaya, F. Adedoyin, M. Budka, H. Dogan, and G. Aksu, "Enhancing Anti-Money Laundering: Development of a Synthetic Transaction Monitoring Dataset," in *2023 IEEE International Conference on e-Business Engineering (ICEBE)*. IEEE, 2023, pp. 47–54. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10356193/

[7] E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, "Realistic synthetic financial transactions for anti-money laundering models," in *37th Conference on Neural Information Processing Systems (NeurIPS) - Datasets and Benchmarks Track*. NeurIPS, 2023.

[8] B. N. Pambudi, I. Hidayah, and S. Fauziati, "Improving money laundering detection using optimized support vector machine," in *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2023.

[9] Y. Zhang and P. Trubey, "Machine learning and sampling scheme: An empirical study of money laundering detection," *Computational Economics*, vol. 54, p. 1043–1063, 2019.

[10] P. Tertychnyi, T. Lindström, C. Liu, and M. Dumas, "Detecting group behavior for anti-money laundering with incomplete network information," in *IEEE International Conference on Big Data (Big Data)*. IEEE, 2022.

[11] T. Suzumura and H. Kanezashi, "Anti-money laundering datasets," GitHub repository, 2021. [Online]. Available: https://github.com/IBM/AMLSim

[12] N. Bakhshinejad, U. T. Nguyen, S. Ghahremani, and R. Soltani, "A graph-based deep learning model for the anti-money laundering task of transaction monitoring," in *16th International Joint Conference on Computational Intelligence (IJCCI 2024)*. SCITEPRESS – Science and Technology Publications, Lda, 2024, pp. 496–507.

[13] M. Caglayan and S. Bahtiyar, "Money laundering detection with node2vec," *Gazi University Journal of Science*, vol. 35, no. 3, pp. 854–873, 2022.

[14] S. Ghahremani and U. T. Nguyen, "Graph data augmentation using generative adversarial networks for imbalanced anti-money laundering datasets," in *Proceedings of the 2nd International Generative AI and Computational Language Modelling Conference (GACLM)*. Valencia, Spain: GACLM, 2025.

[15] J. J. Soria, R. Loayza Abal, and L. Segura Peña, "Machine learning models for money laundering detection in financial institutions. a systematic literature review," in *Proceedings of the 22nd LACCEI International Multi-Conference for Engineering, Education, and Technology*. San Jose, Costa Rica: LACCEI, 2024.

[16] J. Fan *et al.*, "Deep learning approaches for anti-money laundering on mobile transactions: Review, framework, and directions," *arXiv preprint arXiv:2503.10058*, 2025. [Online]. Available: https://arxiv.org/abs/2503.10058

[17] N. Bakhshinejad, "A Graph-Based Deep Learning Model for Anti-Money Laundering," Master's thesis, York University, Toronto, Ontario, April 2023.

[18] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.

[19] H. Wang and H. He, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[20] A. Gupta and R. Sharma, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, vol. 135, p. 110026, 2023.

[21] S. Ghosh, S. K. Kamila, S. Maity, and D. De, "Ensemble learning for class imbalance: a review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 2, p. e1441, 2022.

TABLE IV: Recall and FARs in percentages for the **LI dataset**. The results highlighted in yellow are taken from [7].

Model	Natural Distribution		CVAE		CGAN		SMOTE		ADASYN	
	Recall	FAR	Recall	FAR	Recall	FAR	Recall	FAR	Recall	FAR
AML Threshold (High Recall of Approx. 95%)										
GIN	95.15±0.61	99.70±5.34	95.38±0.45	75.80±3.17	95.25±0.38	82.30±3.43	95.10±0.81	81.40±1.99	95.05±0.66	93.20±1.83
GIN + EU	95.08±0.45	98.40±5.84	95.30±0.58	70.90±3.01	95.20±0.91	77.30±3.19	95.05±0.47	79.10±1.87	95.00±0.65	90.10±1.72
PNA	95.12±0.22	98.30±6.84	95.28±0.94	74.40±2.77	95.18±0.61	79.40±2.94	95.10±0.76	82.70±1.69	95.05±0.33	91.30±1.61
Standard Threshold of 0.5										
GIN	14.59±2.37	94.86±3.42	26.94±1.42	74.25±2.18	24.42±1.94	79.31±2.89	18.95±1.38	78.89±2.12	17.32±1.49	81.10±2.25
GIN + EU	23.26±2.87	78.40±10.83	39.21±1.63	62.36±2.55	35.55±2.18	67.10±3.76	29.91±1.51	72.30±2.04	26.23±1.57	75.80±2.16
PNA	16.43±2.62	82.63±5.80	28.97±1.24	70.68±2.63	25.66±1.76	74.35±3.01	20.71±1.41	77.80±2.25	19.35±1.48	79.50±2.36

[22] B. Stein *et al.*, "Network analytics for anti-money laundering," *arXiv preprint arXiv:2405.19383*, 2024. [Online]. Available: https://arxiv.org/abs/2405.19383

[23] M. Jullum, A. Løland, R. B. Huseby, G. Ånonsen, and J. Lorentzen, "Detecting money laundering transactions with machine learning," *Journal of Money Laundering Control*, vol. 23, no. 1, pp. 173–186, 2020.

[24] G. K. K and B. Bhowmik, "Money laundering detection in banking transactions using rnns and hybrid ensemble," in *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024.

[25] J. Schmidt, D. Pasadakis, M. Sathe, and O. Schenk, "Gamlnet: a graph-based framework for the detection of money laundering," *Preprint*, 2025.

[26] X. Luo, X. Han, W. Zuo, Z. Xu, Z. Wang, and X. Wu, "A dynamic transaction pattern aggregation neural network for money laundering detection," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2022.

[27] A. El-Kilany, A. M. Ayoub, and H. M. E. Kadi, "Detecting suspicious customers in money laundering activities using weighted hits algorithm," in *5th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*. IEEE, 2024.

[28] A. Băltoiu, A. Pătrașcu, and P. Irofti, "Community-level anomaly detection for anti-money laundering," *Preprint, arXiv:1910.11313*, 2019.

[29] A. M. Venugopal, T. S. Tran, and M. Endres, "Synthetic data generation: A comparative study," in *International Database Engineering Applications Symposium (IDEAS)*. ACM, 2022.

[30] Y. Yao, X. Wang, Y. Ma, H. Fang, J. Wei, L. Chen, A. Anaissi, and A. Braytee, "Conditional variational autoencoder with balanced pre-training for generative adversarial networks," *arXiv preprint arXiv:2201.04809*, 2022.

[31] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[34] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2008, pp. 1322–1328.

[35] T. Al-Shehari, M. Kadrie, M. N. Al-Mhiqani, T. Alfakih, H. Alsalman, M. Uddin, S. S. Ullah, and A. Dandoush, "Comparative evaluation of data imbalance addressing techniques for cnn-based insider threat detection," *Scientific Reports*, vol. 14, no. 1, p. 24715, 2024.

[36] F. L. Becerra-Suarez, H. Alvarez-Vasquez, and M. G. Forero, "Improvement of bank fraud detection through synthetic data generation with gaussian noise," *Technologies*, vol. 13, no. 4, p. 141, 2025.

[37] K. Sintayehu and H. Seid, "Developing anti-money laundering identification using machine learning techniques," *Irish Interdisciplinary Journal of Science Research (IIJSR)*, vol. 7, no. 1, pp. 64–74, 2023.

[38] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://arxiv.org/abs/1810.00826

[39] W. Hu, M. Fey, H. Ren, M. Nakata, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 22 118–22 133, 2020. [Online]. Available: https://arxiv.org/abs/2005.00687

[40] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Velickovic, "Principal neighbourhood aggregation for graph nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [Online]. Available: https://arxiv.org/abs/2004.05718