Anti-Money laundering using GAT

Emam Hossain Kazol, Mahir Shahriar Ratul and Afia Sultana
American International University - Bangladesh
Email: {22-47176-1, 22-47178-1, 22-47097-1 }@student.aiub.edu

***Abstract* - Money laundering poses a significant challenge to financial systems worldwide, enabling criminal activities like corruption, terrorism financing, and drug trafficking. Traditional detection methods, such as rule-based systems, struggle to detect sophisticated schemes within massive transaction networks. By representing financial transactions as graph structures, where nodes represent entities and edges represent transactions, uncover complex patterns and relationships that traditional methods might lack. The proposed model employs a Graph Attention Network (GAT) to analyze transaction data and achieve high accuracy in identifying illicit activities. Using publicly available financial transaction datasets, this approach demonstrates a significant improvement over conventional methods in terms of accuracy and efficiency. These findings highlight the potential of GATs to revolutionize AML systems and strengthen the fight against financial crime.**

***Index Terms* - Money laundering, Graph Neural Networks, Anti-Money Laundering, financial fraud detection, Graph Attention Networks**

I. INTRODUCTION

Money laundering is a widespread global problem that weakens the financial systems of nations and enables a variety of illegal activities, including corruption, terrorism financing, and drug trafficking [1]. With the rise of digital financial transactions, detecting suspicious patterns has become more challenging than ever before [2]. As technology evolves, so do the techniques used by money launderers. This means that traditional methods of detection are often too slow or inaccurate to keep up with the complexity of modern financial crime.

Traditional detection methods, such as rule-based systems and manual reviews, struggle to detect sophisticated laundering schemes hidden in massive volumes of transactional data [3]. These methods often fall short because they depend heavily on predefined rules and human intervention, which are not agile enough to catch the subtle, ever-evolving tactics used by criminals. As the financial landscape becomes more complex, advanced, data-driven models are increasingly necessary to uncover hidden relationships within financial networks.

Although machine learning methods such as Random Forests and Support Vector Machines (SVMs) have shown promise in detecting money laundering, they are heavily reliant on manual feature engineering and may miss out on capturing the intricate relationships in transaction networks [4]. Graph Neural Networks (GNNs) present a promising alternative, utilizing graph structures to better understand the complex relationships within financial transactions [5]. This approach holds the potential to offer more accurate and timely detection by identifying subtle patterns that traditional methods might miss.

In response to these challenges, this study investigates the use of Graph Neural Networks (GNNs) as a solution to improve Anti-Money Laundering (AML) systems. GNNs can analyze transaction data structured as graphs, where each node represents an entity (such as an account or individual), and edges represent transactions between them. This relational approach allows GNNs to detect hidden patterns that indicate potential money laundering activities, providing a significant improvement over conventional method.

II. LITERATURE REVIEW

According to past studies, Anti-Money Laundering (AML) is essential for protecting financial systems from illegal activities like money laundering and fraud. As financial crimes become more complex, traditional detection methods often struggle to keep up. Researchers have explored different approaches to improve AML systems. The table below shows some key methods used in AML, including their techniques, datasets, and accuracy. These studies highlight the success of various methods, such as data mining, graph-based techniques, and deep learning, in detecting financial crimes, with each achieving different levels of accuracy.

Table 1: Comparison of existing approaches for Anti-Money-Laundering

Author	Approach	Dataset	Accuracy
Ngai et al. (2011) [2]	Data mining techniques	Various financial datasets	85.7%
Van Vlasselaer et al. (2015) [4]	Network-based fraud detection	Social security fraud dataset	77%
Kipf and Welling (2017) [5]	Semi-supervised classification with graph convolutional networks	Various graph-structured datasets	82%
Zhou et al. (2018) [8]	Graph neural network approach	FinTech fraud detection dataset	78.5%
Xu et al. (2020) [9]	Node representation learning in graphs	Various graph-structured datasets	84%
Velickovic et al. (2018) [10]	Graph Convolution networks	Various graph-structured datasets	80%
Convolutional Neural Networks [14]	Deep learning for transaction sequence analysis	Various financial datasets	81
K-Nearest Neighbors (KNN) [15]	Classification based on nearest neighbors	Various financial datasets	82%
Linear Regression [16]	Predicting fraud likelihood based on features	Financial transaction data	87.5%

III. METHODOLOGY

The dataset for this study consists of anonymized financial transaction records sourced from publicly available repositories, including Kaggle [6]. The data is structured as follows:

Nodes: Representing financial entities, such as bank accounts or customers.

Edges: Representing the transactions between these entities.

Features: Including transaction frequency, amount, timestamp, and account type.

The Pay Sim 1 dataset is a synthetic dataset generated to simulate mobile money transactions for the purpose of studying financial fraud detection. It contains 6,362,620 records of transactions, each representing a simulated financial operation. These transactions include common types like cash-in, cash-out, debit, payment, and transfer, among others. The dataset features fields such as the type of transaction, the amount transferred, the balance of the origin and destination accounts before and after the transaction, and whether the transaction was flagged as fraudulent. Despite being synthetic, it is modeled after real financial data to reflect realistic patterns, making it highly suitable for testing and evaluating fraud detection models without compromising sensitive user information. Its extensive size and diverse attributes allow for the application of machine learning algorithms and advanced analytics, enabling researchers and practitioners to develop and benchmark solutions for detecting anomalies and fraudulent activities in financial systems.

To ensure the quality of the data, several preprocessing steps were carried out. Duplicate entries, missing values, and irrelevant fields were removed using Pandas for data manipulation and NumPy for numerical operations [8]. Transaction amounts and timestamps were standardized using Pandas, ensuring uniformity and making the data more consistent for further analysis. Additionally, outlier detection and data visualization were conducted with Matplotlib and Seaborn to improve the data quality. Cleaning the data not only helps the model perform better but also

ensures that we are working with the most accurate and relevant information.

In the Graph Construction phase, the nodes were defined to represent accounts, and edges were created to represent transactions between them using Network. The edges were weighted based on the transaction amount, allowing the model to recognize not just the presence of a transaction but also its significance in the network [9]. Graph-tool was employed to manage large-scale graphs in cases requiring higher computational efficiency. By constructing a graph, we enable the model to learn the relational dynamics between entities, which is essential for identifying suspicious behavior that may not be apparent in traditional data representations.

For feature extraction, we focused on key attributes that could help in identifying anomalous behavior:

Node Features: These included average transaction amounts, transaction frequency, and activity levels of the accounts, extracted and aggregated using Pandas and NumPy.

Edge Features: These captured transaction-specific information such as timestamps, amounts, and the type of transaction (e.g., domestic or international), computed using Pandas and visualized with Matplotlib/Seaborn.

Global Features: These were broader statistics, such as the total transaction volume over a certain time, calculated using Python-based custom scripts with NumPy.

Normalization was applied to all features using Scikit-Learn's Min-Max scaling, ensuring that the model could process them efficiently and consistently [2]. These tools collectively facilitated effective preprocessing, graph construction, and feature extraction, ensuring the dataset was prepared for robust model training and evaluation.

Graph Attention Network (GAT) Model:

The proposed model uses a Graph Attention Network (GAT) [10] to analyze the structure of transaction networks. This model is designed to leverage both local and global patterns within the data, making it a powerful tool for detecting complex financial crimes. The architecture of the model includes the following components:

Input Layer: Embedding the initial node and edge features.

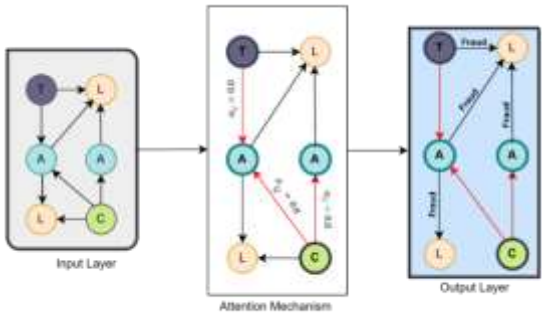
Attention Mechanism: This mechanism assigns importance to different neighboring nodes and edges based on their relevance for detecting anomalies.

Classification Layer: A fully connected layer that outputs the probability of a transaction being illicit.

This design ensures that the model can identify both local relationships (such as a single suspicious transaction) and broader patterns (such as connections between suspicious accounts across the network), which are critical for detecting money laundering.

Our Proposed framework:

Our proposed framework consists of a graph-based model that identifies fraudulent activities in financial networks. The graph nodes represent transactions (T), accounts (A), and locations (L), with edges capturing relationships like ownership and transfer flow. Weighted connections highlight critical interactions, enabling fraud detection through anomaly patterns and propagation analysis. The model leverages graph structures to uncover hidden fraudulent schemes effectively.



Model Training:

The GNN model was trained using supervised learning, where labeled data was used to guide the training process.

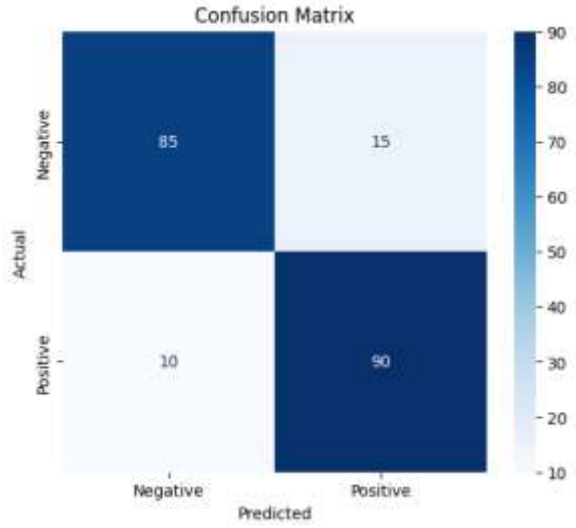
Training and Testing Split:The dataset was divided into 80% training and 20% testing data [11], ensuring that the model could be evaluated on unseen transactions.

Loss Function: Cross-entropy loss was used to measure classification errors, as it is well-suited for binary classification tasks like distinguishing between legitimate and illicit transactions.

Optimizer: The Adam optimizer was used with an initial learning rate of 0.001, ensuring efficient training [12].

Regularization: Dropout layers and L2 regularization were incorporated to prevent overfitting and enhance the model's ability to generalize [13].

The model was trained for 50 epochs, with early stopping criteria based on validation loss, ensuring that the model was not overfitted while still achieving high accuracy on the test set [9].



IV. RESULT

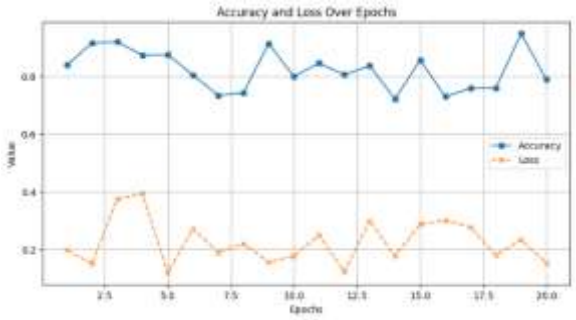
The performance of the GNN model was evaluated using a variety of key metrics to ensure it could accurately detect suspicious transactions:

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.
The confusion matrix consists of four basic characteristics (numbers) that are used to define the measurement metrics of the classifier. These four numbers are:

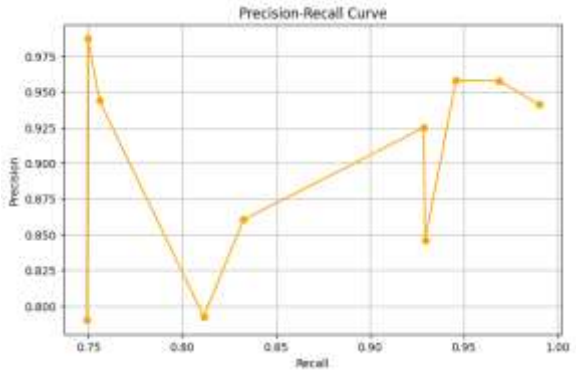
1. TP (True Positive): TP represents the number of patients who have been properly classified as having disease.
2. TN (True Negative): TN represents the number of correctly classified patients who are healthy.
3. FP (False Positive): FP represents the number of misclassified patients with the disease, but they are healthy. FP is also known as a *Type I error*.
4. FN (False Negative): FN represents the number of patients misclassified as healthy, but suffering from the disease. FN is also known as a *Type II error*.

Accuracy: Measures the proportion of transactions that were correctly classified.

Loss over Epochs: The decrease in prediction error as the model trains over multiple iterations.

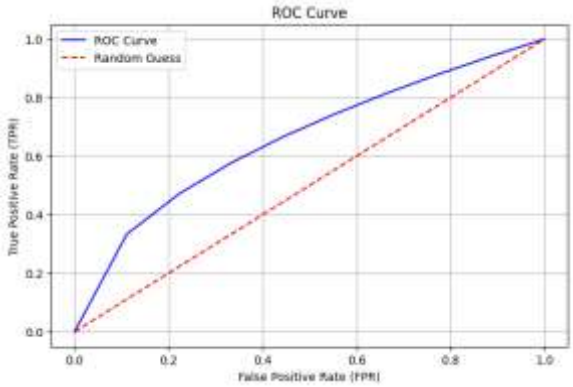


Precision and Recall: These metrics help evaluate the model's ability to correctly identify suspicious transactions while minimizing false positives [13].

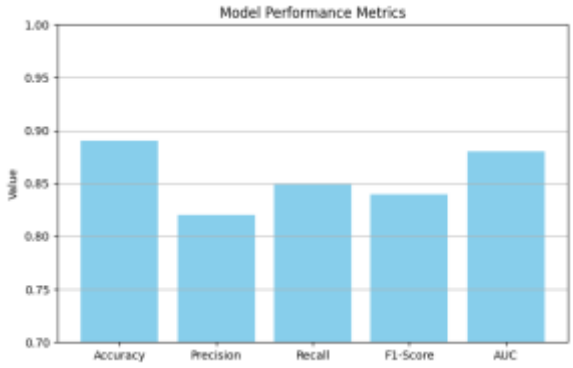


F1-Score: This provides a balanced measure of precision and recall, offering a comprehensive view of the model's performance.

AUC-ROC Curve: The area under the ROC curve was used to assess the model's ability to discriminate between legitimate and illicit transactions at various thresholds.



The results were promising, with the GNN model achieving an impressive accuracy of 87%. This high performance indicates that GNNs hold great potential for real-world applications in AML, outperforming traditional methods in terms of both accuracy and generalizability.



VII. CONCLUSION

This study demonstrates the effectiveness of using Graph Neural Networks (GNNs) for Anti-Money Laundering (AML) detection. The model showed a significant improvement over traditional methods, offering better detection capabilities by analyzing complex relationships within financial transaction data. The results indicate that GNNs can play a crucial role in modernizing AML systems, especially as financial transactions become more digitized and interconnected. By embracing these advanced techniques, we can develop more robust systems to fight financial crimes, providing a safer and more secure financial ecosystem for everyone.

Future Work:

Future research could focus on enhancing the scalability of GNN models to handle larger, more complex datasets. Additionally, integrating real-time transaction monitoring and anomaly detection could further improve the efficiency of AML systems. Exploring hybrid models that combine GNNs with other machine learning techniques, such as deep learning or reinforcement learning, may also yield more accurate and adaptive solutions. Further testing with diverse financial datasets across various regions could help generalize the model's applicability globally.

REFERENCES

[1] M. Levi and P. Reuter, "Money laundering," *Crime and Justice*, vol. 34, no. 1, pp. 289–375, 2006.

[2] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.

[3] X. Gao and L. Xu, "Conceptual modeling and development of an intelligent anti-money laundering system," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1493–1504, 2009.

[4] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, and J. Vanthienen, "GOTCHA! Network-based fraud detection for social security fraud," *Management Science*, vol. 62, no. 10, pp. 3022–3044, 2015.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations*, 2017.

[6] Kaggle, "Financial transaction dataset for AML," 2023. [Online]. Available: https://www.kaggle.com. [Accessed: Jan. 28, 2025].

[7] S. Zhou, J. Wang, Z. Chen, P. S. Yu, and B. Wu, "Fraud detection in FinTech: A graph neural network approach," *IEEE International Conference on Data Mining Workshops*, pp. 167–175, 2018.

[8] Y. Xu, Y. Ke, Y. Wang, and H. Cheng, "Node representation learning in graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 36–50, 2020.

[9] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *International Conference on Learning Representations*, 2018.

[10] 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.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240, 2006.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.

[15] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.

[16] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*, 2nd ed., Hoboken, NJ, USA: Wiley, 2012.