GROUP 3

Anti-Money Laundering Detection with Graph Neural Networks (GNN)

TABLE OF CONTENTS

I. Introduction

- **Sophisticated financial crime: Money laundering is a complex and widespread financial crime.**

- **Global threat: It poses a significant threat to the global financial system.**

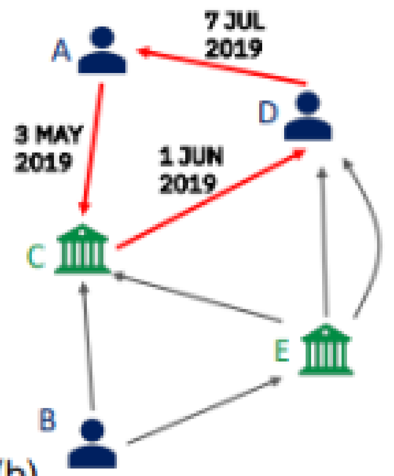
- **Concealment of illegal origins: Involves processes to hide the origins of money obtained through illegal activities.**

- **Examples of illegal activities: Such activities include drug trafficking, terrorism, fraud, and corruption.**

• Example

(a)

Trans. ID	Timestamp	Source bank ID	Source Account	Target bank ID	Target Account	Amount	Currency	Payment type
0	3 MAY 2019 12:45	1	A	1	C	1400	USD	Cheque
1	15 MAY 2019 07:34	2	B	1	C	710	EUR	ACH
2	18 MAY 2019 16:55	3	E	1	C	950	USD	Credit card
3	1 JUN 2019 10:06	1	C	3	D	1200	CHF	Wire
4	27 JUN 2019 13:18	3	E	3	D	2300	EUR	Credit card
5	7 JUL 2019 11:14	3	D	1	A	1100	USD	Credit card
6	14 JUL 2019 09:37	2	B	3	E	650	USD	ACH
7	20 JUL 2019 14:02	3	E	3	D	2500	USD	Wire



(b)

Figure 1 Financial transactions in (a) tabular format and in (b) graph format.

All laundering in the data follows one of these 8 patterns. As with other aspects of this data noted above, knowing all the transcation involved in particular laundering patterns is an immense challenge with real data.

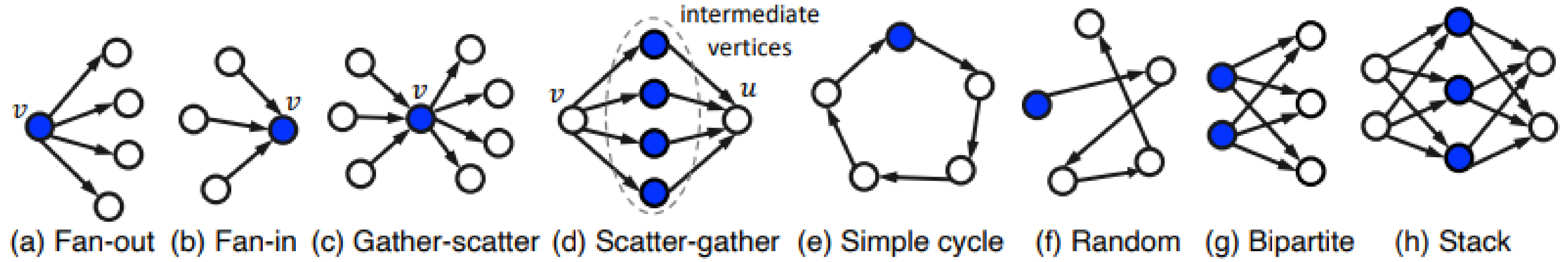


Figure 2: Laundering Patterns Modelled

II.Survey

This survey examines key studies that have contributed to the understanding and development of GNNs in the context of AML detection:


arXiv > cs > arXiv:2306.16424

Search...
Help |

Computer Science > Artificial Intelligence

[Submitted on 22 Jun 2023 (v1), last revised 25 Jan 2024 (this version, v3)]

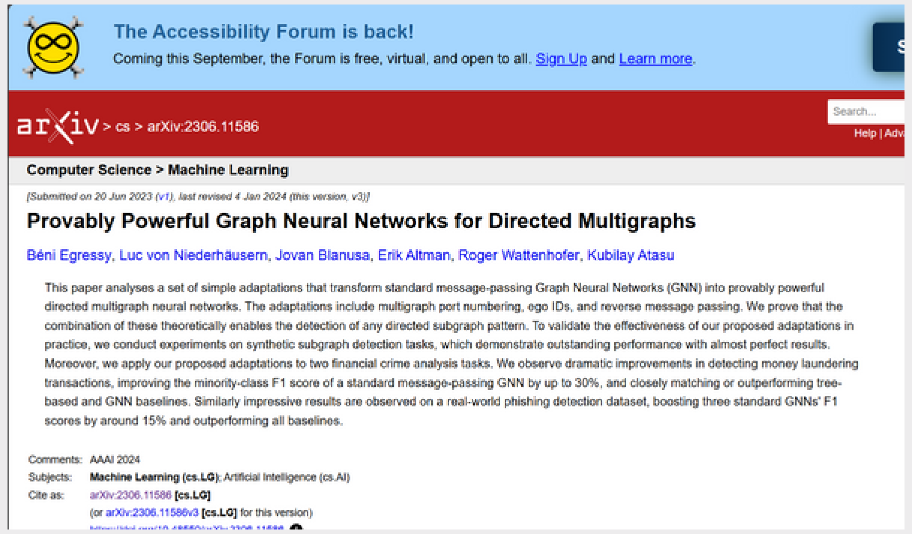
Realistic Synthetic Financial Transactions for Anti-Money Laundering Models

Erik Altman, Jovan Blanuša, Luc von Niederhäusern, Béni Egressy, Andreea Anghel, Kubilay Atasu

With the widespread digitization of finance and the increasing popularity of cryptocurrencies, the sophistication of fraud schemes devised by cybercriminals is growing. Money laundering -- the movement of illicit funds to conceal their origins -- can cross bank and national boundaries, producing complex transaction patterns. The UN estimates 2-5\% of global GDP or $0.8 - $2.0 trillion dollars are laundered globally each year. Unfortunately, real data to train machine learning models to detect laundering is generally not available, and previous synthetic data generators have had significant shortcomings. A realistic, standardized, publicly-available benchmark is needed for comparing models and for the advancement of the area.
To this end, this paper contributes a synthetic financial transaction dataset generator and a set of synthetically generated AML (Anti-Money Laundering) datasets. We have calibrated this agent-based generator to match real transactions as closely as possible and made the datasets public. We describe the generator in detail and demonstrate how the datasets generated can help compare different machine learning models in terms of their AML abilities. In a key way, using synthetic data in these comparisons can be even better than using real data: the ground truth labels are complete, whilst many laundering transactions in real data are never detected.

Subjects: **Artificial Intelligence (cs.AI)**; Machine Learning (cs.LG); Computational Finance (q-fin.CP)
Cite as: arXiv:2306.16424 **[cs.AI]**
(or arXiv:2306.16424v3 **[cs.AI]** for this version)
https://doi.org/10.48550/arXiv.2306.16424

Submission history
From: Kubilay Atasu [view email]


The Accessibility Forum is back!
Coming this September, the Forum is free, virtual, and open to all. Sign Up and Learn more.

arXiv > cs > arXiv:2306.11586

Search...
Help | Adv

Computer Science > Machine Learning

[Submitted on 20 Jun 2023 (v1), last revised 4 Jan 2024 (this version, v3)]

Provably Powerful Graph Neural Networks for Directed Multigraphs

Béni Egressy, Luc von Niederhäusern, Jovan Blanusa, Erik Altman, Roger Wattenhofer, Kubilay Atasu

This paper analyses a set of simple adaptations that transform standard message-passing Graph Neural Networks (GNN) into provably powerful directed multigraph neural networks. The adaptations include multigraph port numbering, ego IDs, and reverse message passing. We prove that the combination of these theoretically enables the detection of any directed subgraph pattern. To validate the effectiveness of our proposed adaptations in practice, we conduct experiments on synthetic subgraph detection tasks, which demonstrate outstanding performance with almost perfect results. Moreover, we apply our proposed adaptations to two financial crime analysis tasks. We observe dramatic improvements in detecting money laundering transactions, improving the minority-class F1 score of a standard message-passing GNN by up to 30%, and closely matching or outperforming tree-based and GNN baselines. Similarly impressive results are observed on a real-world phishing detection dataset, boosting three standard GNNs' F1 scores by around 15% and outperforming all baselines.

Comments: AAAI 2024
Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI)
Cite as: arXiv:2306.11586 **[cs.LG]**
(or arXiv:2306.11586v3 **[cs.LG]** for this version)


The Accessibility Forum is back!
Coming this September, the Forum is free, virtual, and open to all. Sign Up and Learn more.

arXiv > cs > arXiv:1812.00076

Search...
Help

Computer Science > Social and Information Networks

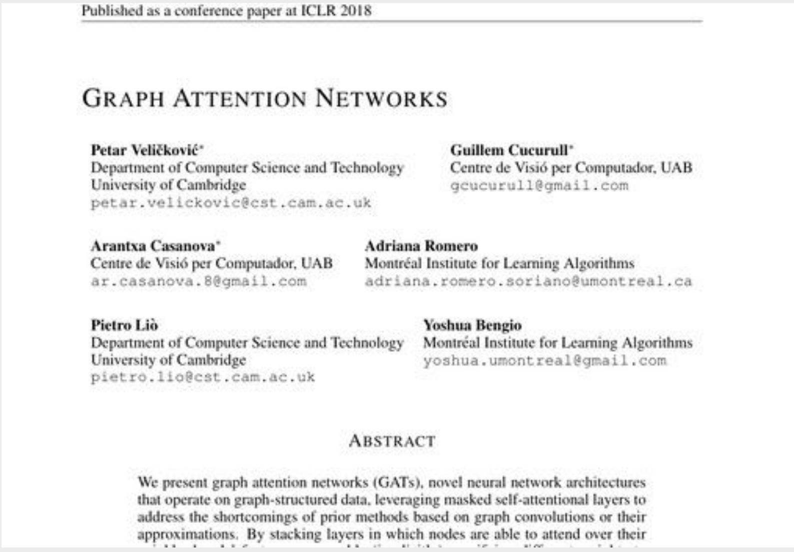
[Submitted on 30 Nov 2018]

Scalable Graph Learning for Anti-Money Laundering: A First Look

Mark Weber, Jie Chen, Toyotaro Suzumura, Aldo Pareja, Tengfei Ma, Hiroki Kanezashi, Tim Kaler, Charles E. Leiserson, Tao B. Schar

Organized crime inflicts human suffering on a genocidal scale: the Mexican drug cartels have murdered 150,000 people since 2006, upwards of 700,000 people per year are "exported" in a human trafficking industry enslaving an estimated 40 million people. These nefarious industries rely on sophisticated money laundering schemes to operate. Despite tremendous resources dedicated to anti-money laundering (AML) only a tiny fraction of illicit activity is prevented. The research community can help. In this brief paper, we map the structural and behavioral dynamics driving the technical challenge. We review AML methods, current and emergent. We provide a first look at scalable graph convolutional neural networks for forensic analysis of financial data, which is massive, dense, and dynamic. We report preliminary experimental results using a large synthetic graph (1M nodes, 9M edges) generated by a data simulator we created called AMLSim. We consider opportunities for high performance efficiency, in terms of computation and memory, and we share results from a simple graph compression experiment. Our results support our working hypothesis that graph deep learning for AML bears great promise in the fight against criminal financial activity.

Comments: NeurIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, Montreal, Canada
Subjects: **Social and Information Networks (cs.SI)**; Artificial Intelligence (cs.AI); Machine Learning (cs.LG)
Cite as: arXiv:1812.00076 **[cs.SI]**


Published as a conference paper at ICLR 2018

GRAPH ATTENTION NETWORKS

Petar Veličković*
Department of Computer Science and Technology
University of Cambridge
petar.velickovic@cst.cam.ac.uk

Guillem Cucurull*
Centre de Visió per Computador, UAB
gcucurull@gmail.com

Arantxa Casanova*
Centre de Visió per Computador, UAB
ar.casanova.8@gmail.com

Adriana Romero
Montréal Institute for Learning Algorithms
adriana.romero.soriano@umontreal.ca

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
pietro.lio@cst.cam.ac.uk

Yoshua Bengio
Montréal Institute for Learning Algorithms
yoshua.umontreal@gmail.com

ABSTRACT

We present graph attention networks (GATs), novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations. By stacking layers in which nodes are able to attend over their

Realistic Synthetic Financial Transactions for Anti-Money Laundering Models: discusses synthetic financial datasets that enhance AML model training without privacy concerns, allowing for better exploration of complex laundering patterns.

Provably Powerful Graph Neural Networks for Directed Multigraphs: explores GNNs for directed multigraphs, showcasing their effectiveness in modeling complex financial relationships and detecting hidden illicit activities.

Scalable Graph Learning for Anti-Money Laundering: focuses on the scalability of Graph Convolutional Networks (GCNs), showing how they reduce computational burdens while effectively modeling financial networks.

Graph Attention Networks: how attention mechanisms improve detection accuracy by focusing on key parts of the network.

III. DATA

- HI-Large_Patterns.txt
- HI-Large_Trans.csv
- HI-Medium_Patterns.txt
- HI-Medium_Trans.csv
- HI-Small_Patterns.txt
- HI-Small_Trans.csv
- LI-Large_Patterns.txt
- LI-Large_Trans.csv
- LI-Medium_Patterns.txt
- LI-Medium_Trans.csv
- LI-Small_Patterns.txt
- LI-Small_Trans.csv

6 datasets here divided into two groups:

- Group HI (Higher Illicit Ratio)
- Group LI (Lower Illicit Ratio)

Two files for each of the six datasets:

- .csv: Transactions
- .txt: Laundering Pattern Transactions

Timestamp	From Bank	Account	To Bank	Account	Amount Received	Receiving Currency	Amount Paid	Payment Currency	Payment Format	Is Laundering
1/1/2019 0:22	800319940	8004ED620	808519790	872ABC810	120.92	US Dollar	120.92	US Dollar	Credit Card	0
1/1/2019 0:05	8021ADE00	80238F220	9A7F59FA0	A23691240	33.97	US Dollar	33.97	US Dollar	Credit Card	1
1/1/2019 0:14	801946100	8023F0980	83585F5A0	948893910	79.20	US Dollar	79.20	US Dollar	Credit Card	0
1/1/2019 0:05	80010C840	800122AA0	80010C840	800122AA0	8,834.09	Euro	10351.64	US Dollar	ACH	0
1/1/2019 0:05	80010C840	800122AA0	80010CF20	80012DA00	8,834.09	Euro	8834.09	Euro	ACH	0
1/1/2019 0:08	80010CF20	80012DA00	80010CF20	80012DA00	9,682.16	US Dollar	8262.75	Euro	ACH	0
1/1/2019 0:08	80010CF20	80012DA00	80010BD60	80011E460	9,682.16	US Dollar	9682.16	US Dollar	ACH	0
1/1/2019 0:03	800319940	800466670	80029A010	8002F6F20	9,125.22	US Dollar	9125.22	US Dollar	ACH	0

Other Currencies

Yuan
Yen
Indian Rupee
Ruble
UK Pound
Canadian Dollar
Australian Dollar
Mexican Peso
Brazilian Real
Swiss Franc
Shekel
Saudi Riyal
Bitcoin

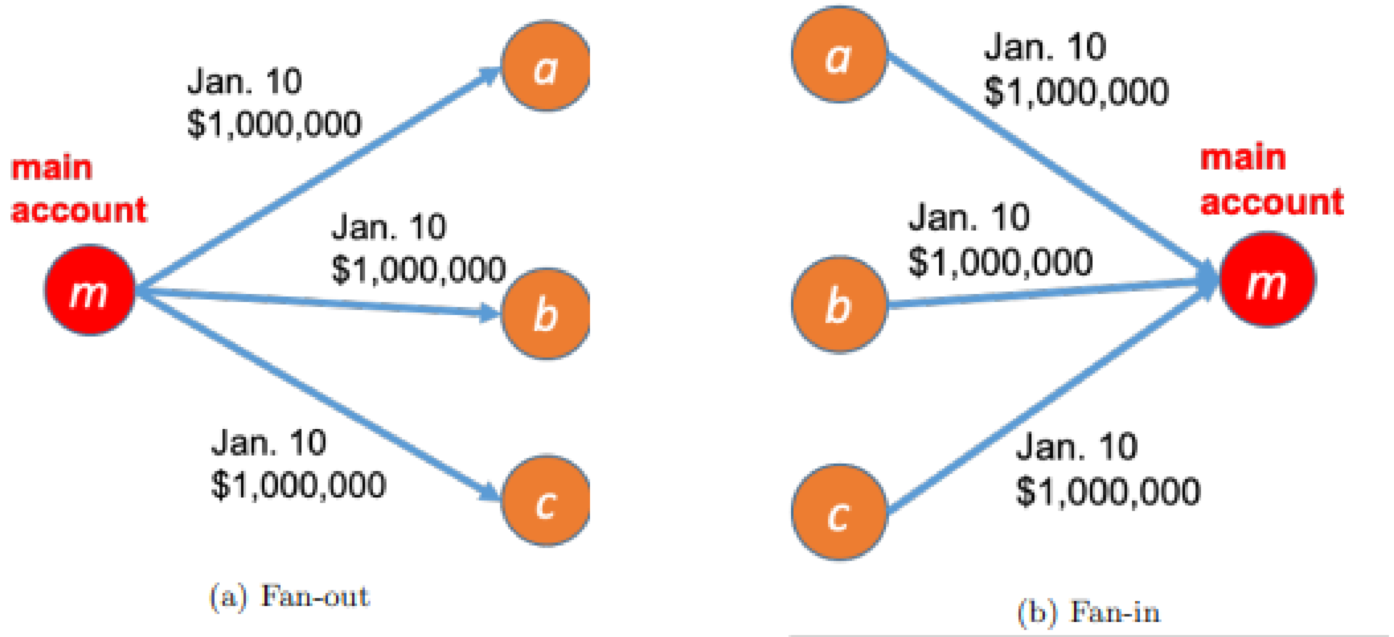
Other Formats

Wire
Cheque
Cash

AML – SAMPLE TRANSACTION

Methods

1. Why Apply Graph Neural Networks (GNNs)?



Graph Neural Networks (GNNs) provide a powerful alternative to traditional methods by leveraging the structure of financial transaction networks.

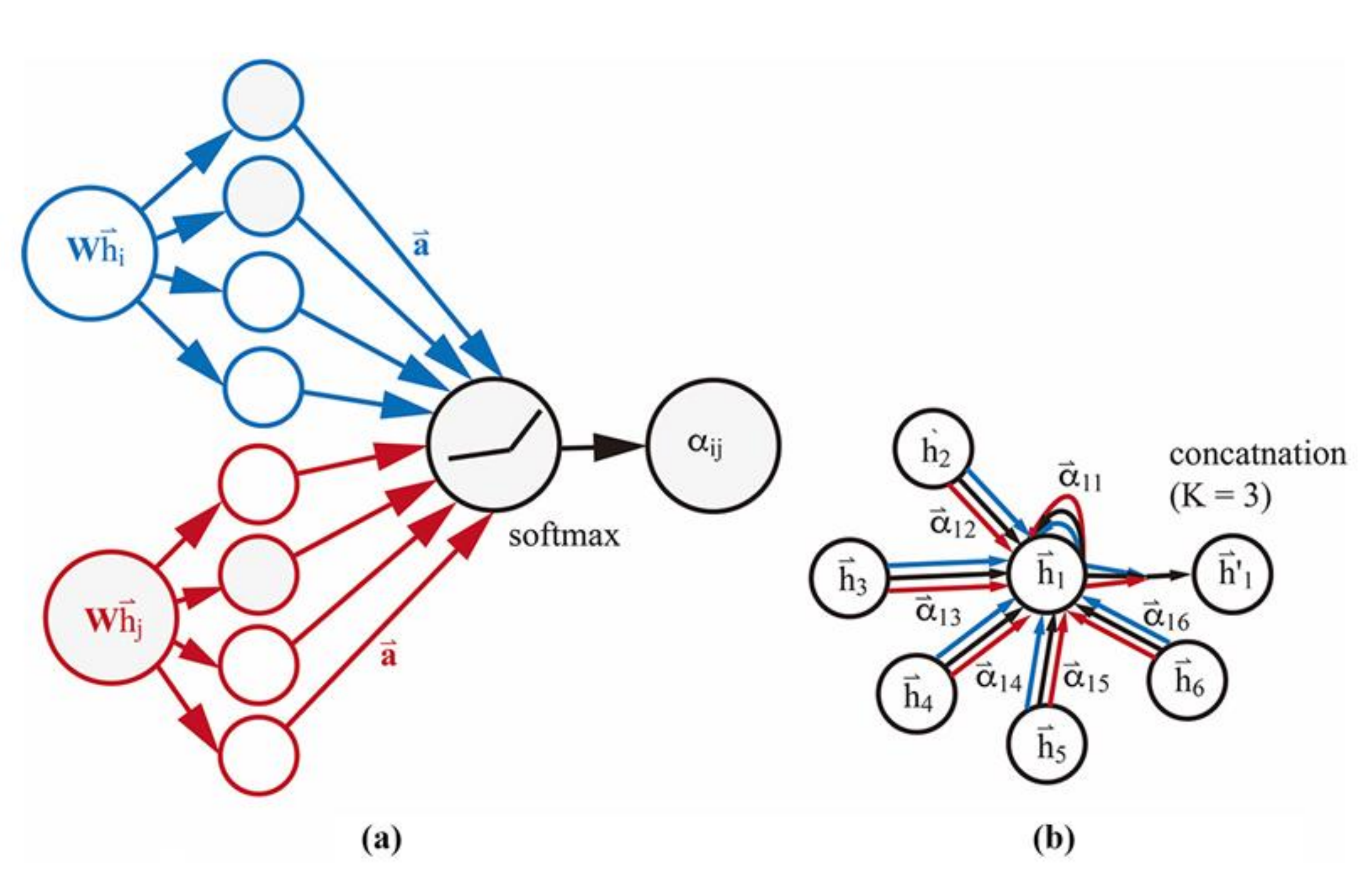
Connectivity of Financial Transactions:

- Nodes: nodes represent entities such as bank accounts, individuals, or companies.
- Edges: Edges represent the financial transactions between these entities.

Typical Patterns in Financial Networks:

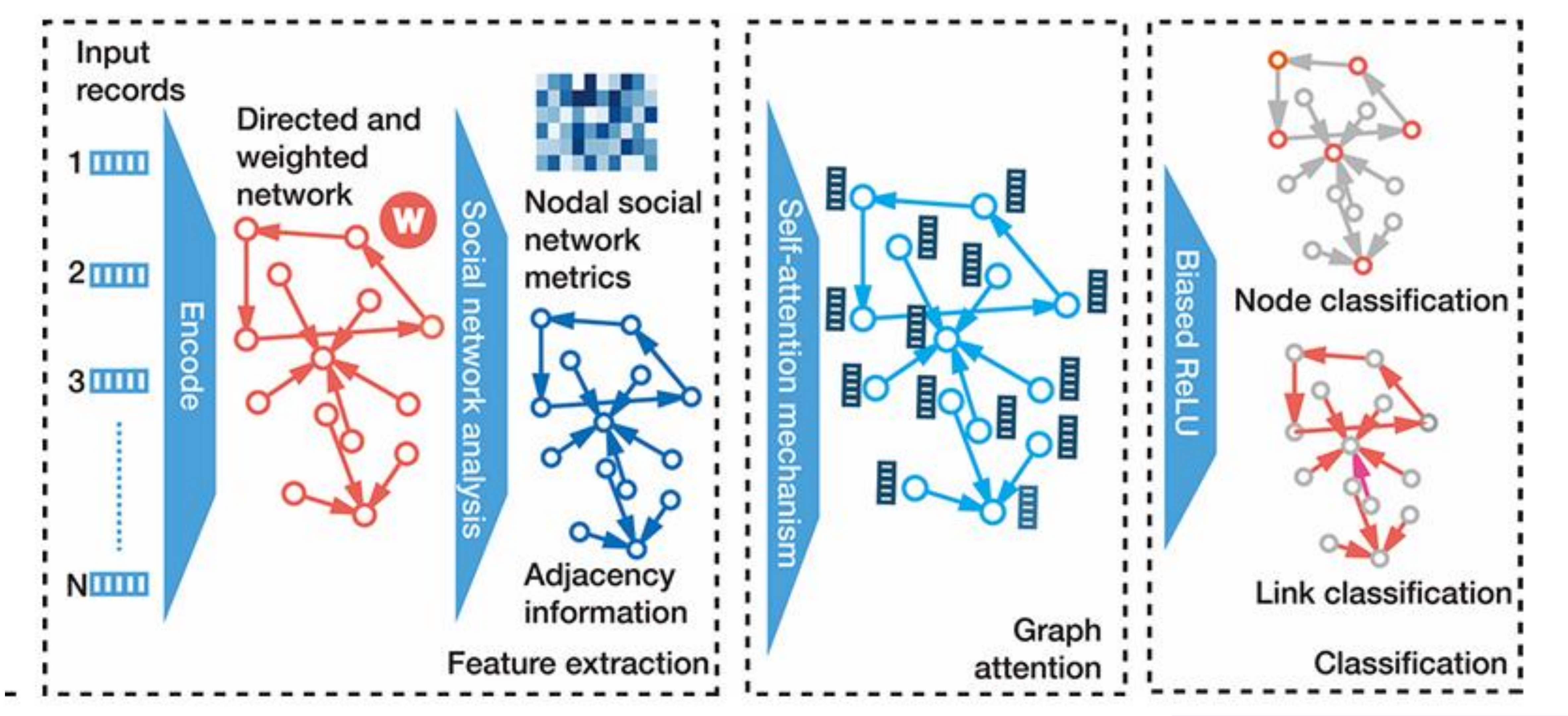
- Fan-Out Patterns: This occurs when a single account disperses funds across multiple other accounts.
- Fan-In Patterns: This pattern is observed when multiple accounts funnel funds into a single account.
- Circular Patterns: funds are cycled between accounts, often through multiple layers of transactions

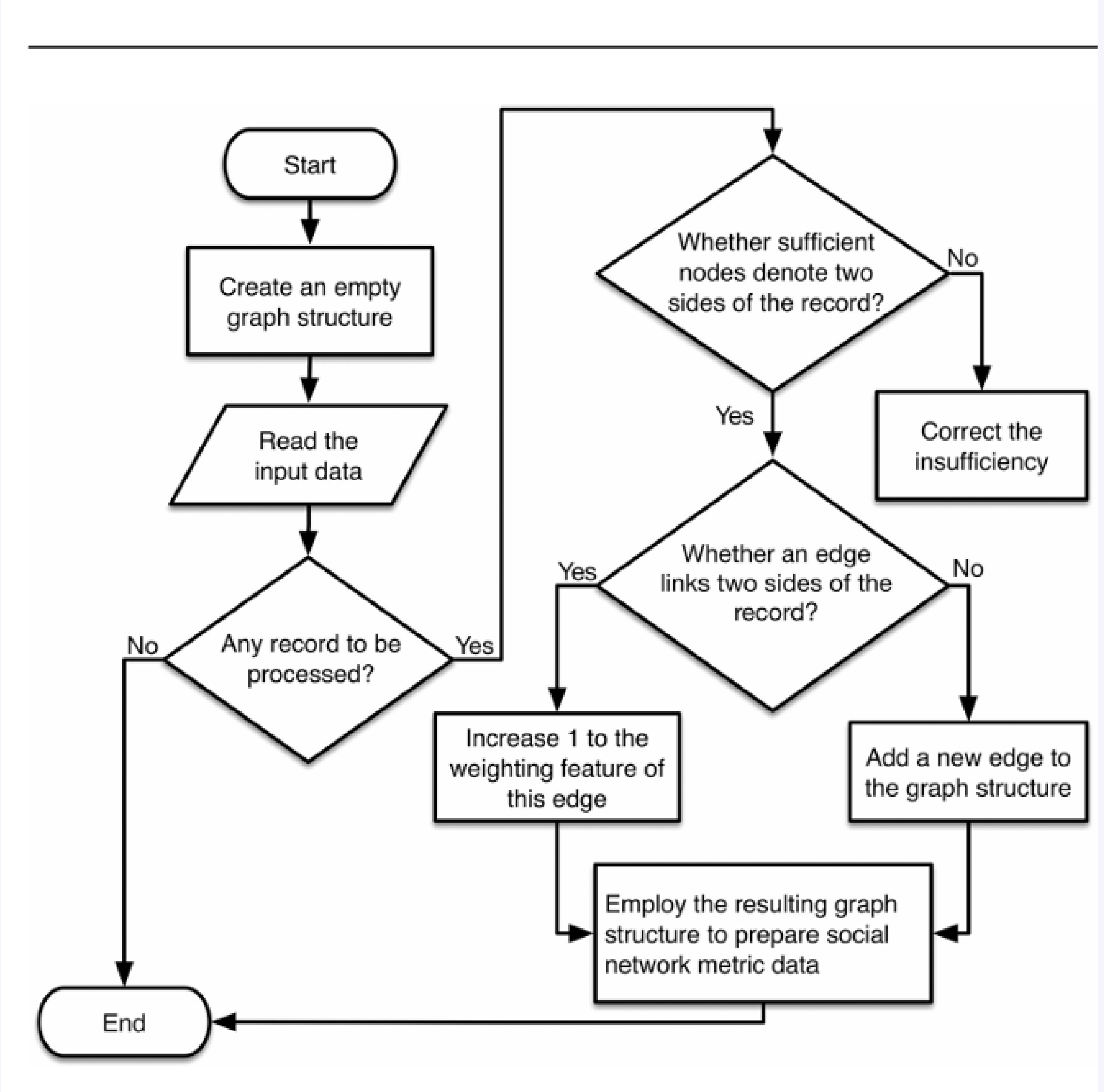
2. Overview of Graph Attention Networks (GAT)



01	Attention Mechanism: Dynamically weighs the importance of each edge, crucial for assessing the risk in financial transactions.
02	Information Aggregation: Gathers information from the most relevant edges to update node features, reflecting both the node's attributes and its key neighbors.
03	Multi-Head Attention: Uses multiple attention heads in parallel to enhance model robustness, capturing diverse perspectives and richer features.
04	Non-Linear Transformation: Applies a non-linear function (e.g., ReLU) to aggregated features, allowing the model to capture complex relationships within the graph.

3. Application of GAT to the AML





4. Comparison with GCN and GIN

GRAPH CONVOLUTIONAL NETWORKS (GCN):

1. Architecture: Aggregates neighbor information through convolution, applying a linear transformation and non-linear activation.
2. Limitations: Treats all neighbors equally, which may overlook crucial patterns in detecting money laundering.

GRAPH ISOMORPHISM NETWORKS (GIN):

- Architecture: GINs use summation for neighbor aggregation, followed by an MLP, enabling them to capture subtle differences in graph structures.
- Strengths: Highly expressive in distinguishing structural properties of graphs, though this can increase computational complexity, especially in large-scale graphs.

COMPARATIVE ANALYSIS:

- Performance:
- GAT: Excels in weighting important transactions, ideal for detecting AML.
- GCN: Faster and more scalable but may miss key patterns.
- GIN: Strong in distinguishing complex structures, but computationally intensive.
- Scalability:
- GCN: Best for large graphs due to faster training.
- GAT: Captures subtle patterns but may require more resources.
- GIN: Powerful but resource-heavy, especially with complex graphs.

5.Evaluation Metric:

F1 Score: Essential for imbalanced data, balancing precision (correctly flagged illicit transactions) and recall (accurately detecting illicit transactions). A high F1 score indicates effective detection with minimal false positives.

Implementation

1. Data Loading and Preprocessing
   + Label Encoding
   + Timestamp Normalization
   + Unique Identifiers

3. GAT Model Architecture
   + Input Layer
   + GAT Layers
   + Output Layer

```
class GAT(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels,
heads):
        super().__init__()
        self.conv1 = GATConv(in_channels, hidden_channels, heads,
dropout=0.6)
        self.conv2 = GATConv(hidden_channels * heads,
int(hidden_channels/4), heads=1, concat=False, dropout=0.6)
        self.lin = Linear(int(hidden_channels/4), out_channels)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x, edge_index, edge_attr):
        x = F.dropout(x, p=0.6, training=self.training)
        x = F.elu(self.conv1(x, edge_index, edge_attr))
        x = F.dropout(x, p=0.6, training=self.training)
        x = F.elu(self.conv2(x, edge_index, edge_attr))
        x = self.lin(x)
        x = self.sigmoid(x)

        return x
```

2. Graph Construction

+ Node features

```
def get_node_attr(currency_ls, paying_df,receiving_df, accounts):
        node_df = paid_currency_aggregate(currency_ls, paying_df,
accounts)
        node_df = received_currency_aggregate(currency_ls,
receiving_df, node_df)
        node_label = torch.from_numpy(node_df['Is
Laundering'].values).to(torch.float)
        node_df = node_df.drop(['Account', 'Is Laundering'], axis=1)
        node_df = df_label_encoder(node_df,['Bank'])
#         node_df = torch.from_numpy(node_df.values).to(torch.float)  #
comment for visualization
        return node_df, node_label
```

+ Edge features

```
def get_edge_df(accounts, df):
        accounts = accounts.reset_index(drop=True)
        accounts['ID'] = accounts.index
        mapping_dict = dict(zip(accounts['Account'], accounts['ID']))
        df['From'] = df['Account'].map(mapping_dict)
        df['To'] = df['Account.1'].map(mapping_dict)
        df = df.drop(['Account', 'Account.1', 'From Bank', 'To Bank'],
axis=1)

        edge_index = torch.stack([torch.from_numpy(df['From'].values),
torch.from_numpy(df['To'].values)], dim=0)

        df = df.drop(['Is Laundering', 'From', 'To'], axis=1)

#         edge_attr = torch.from_numpy(df.values).to(torch.float)  #
comment for visualization

        edge_attr = df  # for visualization
    return edge_attr, edge_index
```

Implementation

4. Training the Model
+ Attention Mechanism
+ Optimization

```
model = GAT(in_channels=data.num_features, hidden_channels=16,
out_channels=1, heads=8)
model = model.to(device)
criterion = torch.nn.BCELoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.0001)

split = T.RandomNodeSplit(split='train_rest', num_val=0.1, num_test=0)
data = split(data)

train_loader = loader = NeighborLoader(
    data,
    num_neighbors=[30] * 2,
    batch_size=256,
    input_nodes=data.train_mask,
)

test_loader = loader = NeighborLoader(
    data,
    num_neighbors=[30] * 2,
    batch_size=256,
    input_nodes=data.val_mask,
)
```

5. Model Evaluation

```
for i in range(epoch):
    total_loss = 0
    model.train()
    for data in train_loader:
        optimizer.zero_grad()
        data.to(device)
        pred = model(data.x, data.edge_index, data.edge_attr)
        ground_truth = data.y
        loss = criterion(pred, ground_truth.unsqueeze(1))
        loss.backward()
        optimizer.step()
        total_loss += float(loss)
    if epoch%10 == 0:
        print(f"Epoch: {i:03d}, Loss: {total_loss:.4f}")
        model.eval()
        acc = 0
        total = 0
        with torch.no_grad():
            for test_data in test_loader:
                test_data.to(device)
                pred = model(test_data.x, test_data.edge_index,
test_data.edge_attr)
                ground_truth = test_data.y
                correct = (pred ==
ground_truth.unsqueeze(1)).sum().item()
                total += len(ground_truth)
                acc += correct
            acc = acc/total
            print('accuracy:', acc)
```

VII. Results

1. Graph Convolutional Network (GCN)

```
Run summary:
 best_test_f1 0.71429
      f1/test 0.72093
     f1/train 0.47619
f1/validation 0.70229
```

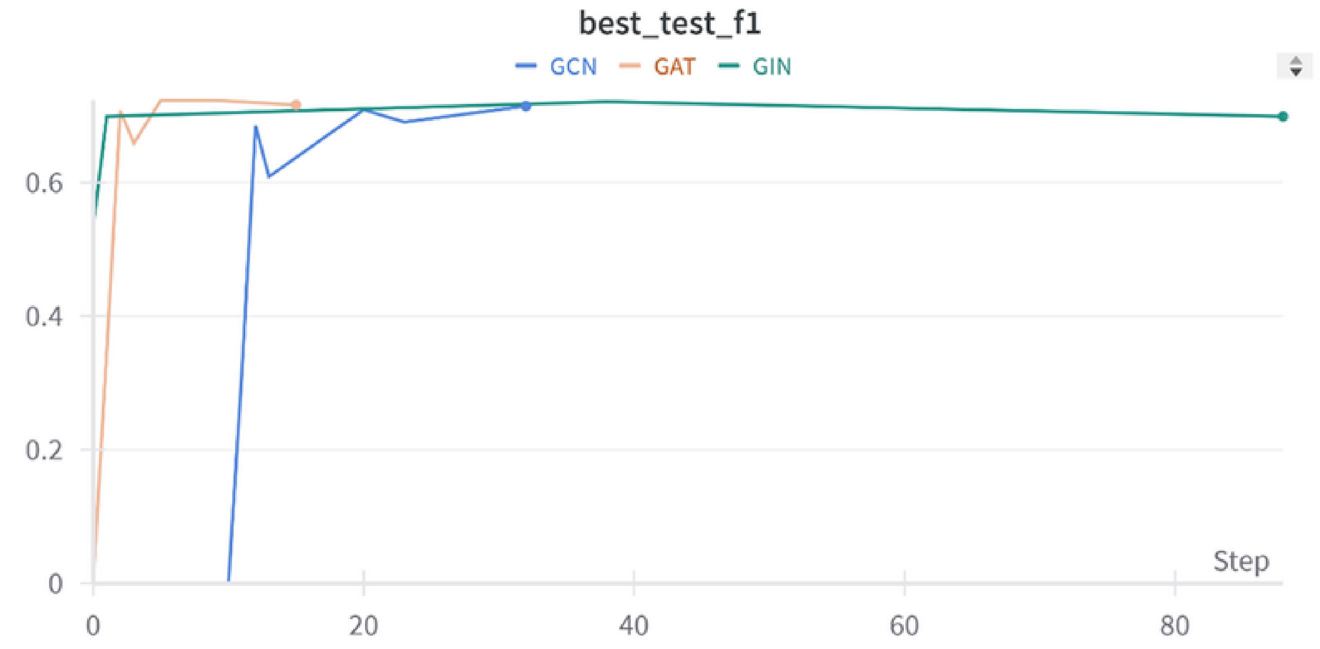
2. Graph Attention Network (GAT)

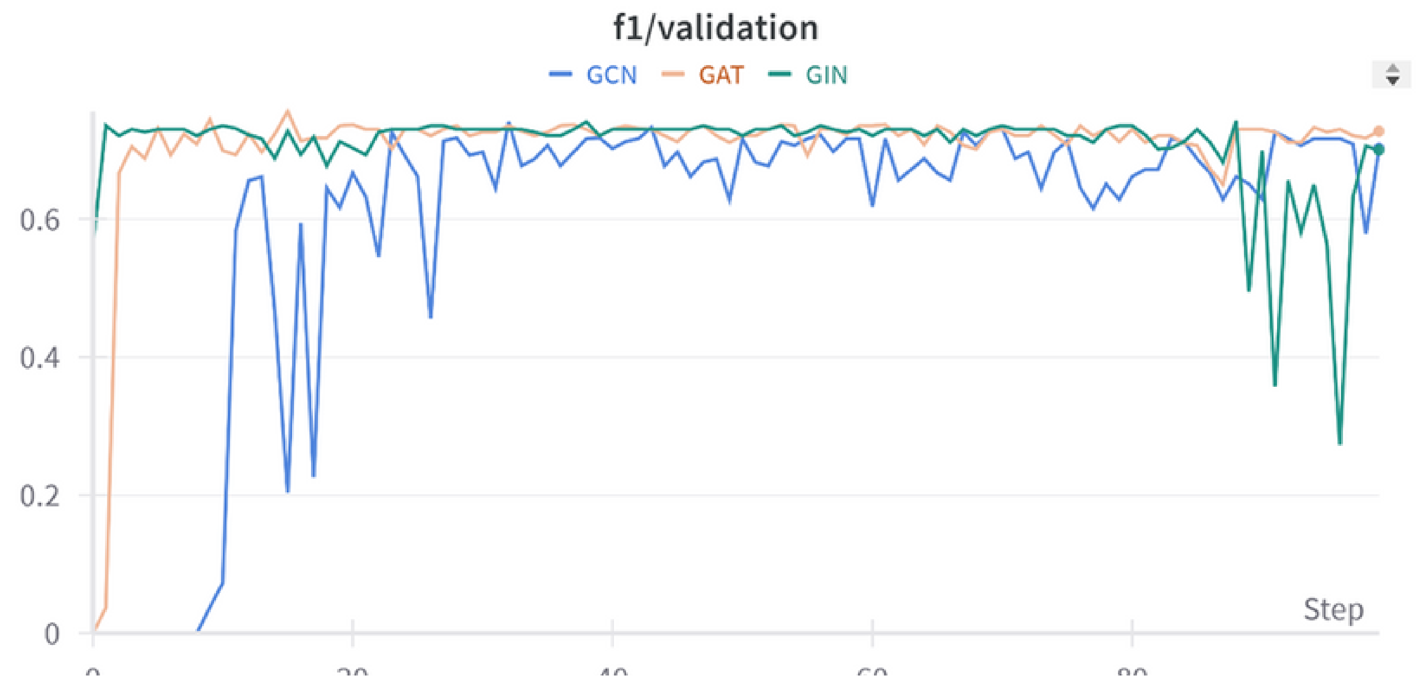
```
Run summary:
 best_test_f1 0.71605
      f1/test 0.72289
     f1/train 0.32353
f1/validation 0.72727
```

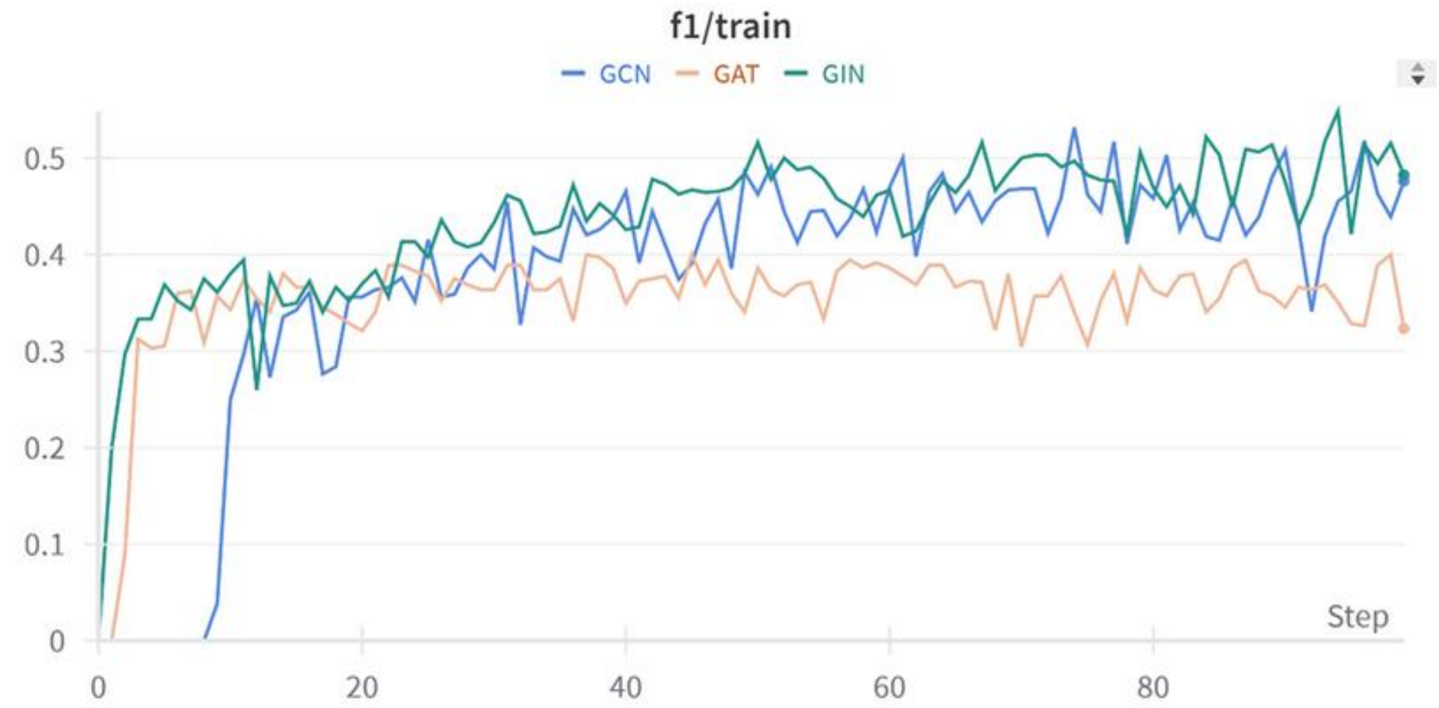
3. Graph Isomorphism Network (GIN)

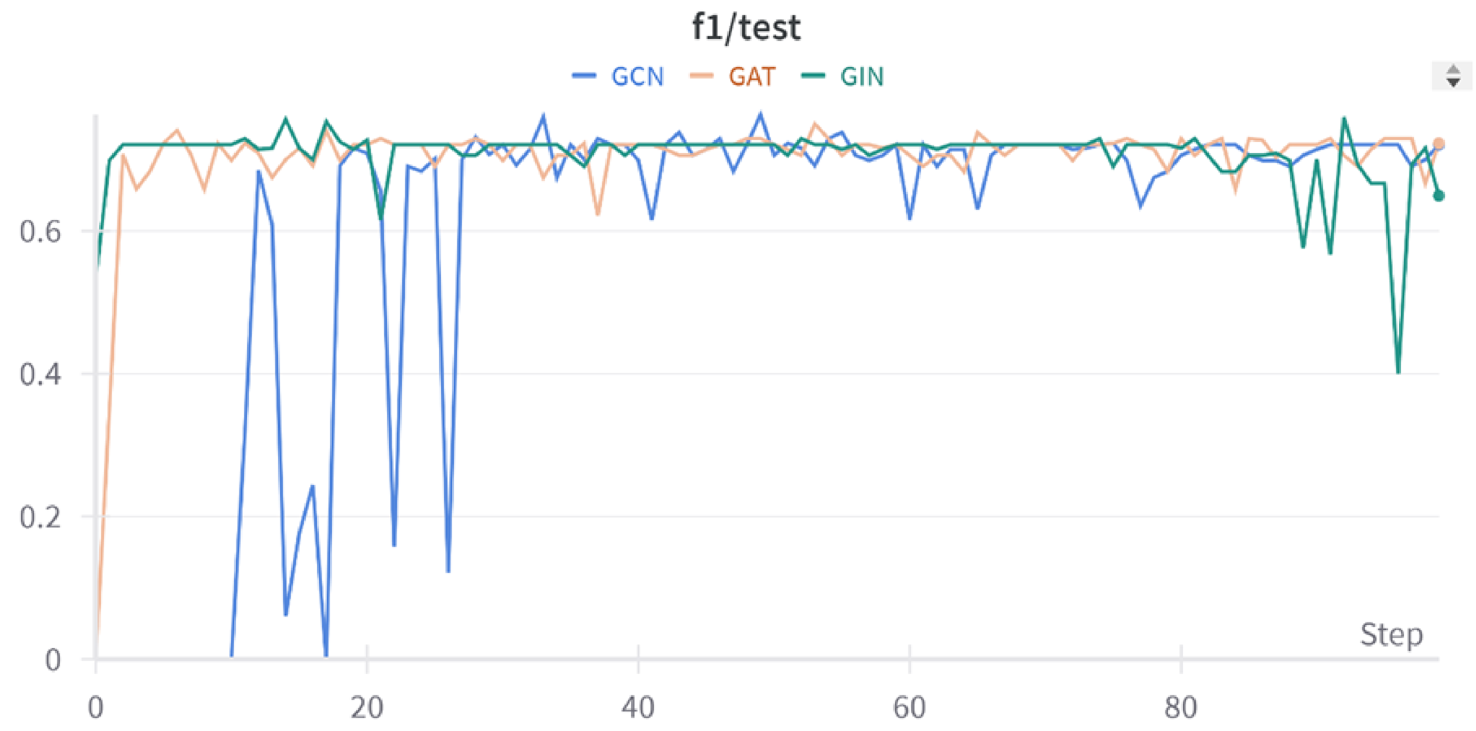
```
Run summary:
 best_test_f1 0.6988
      f1/test 0.64935
     f1/train 0.48241
f1/validation 0.7
```

4. Comparative Analysis









Conclusion

- **Potential of GNNs:** The study shows that GNNs can enhance Anti-Money Laundering (AML) systems beyond traditional methods.

- **Comparison of GNN Architectures**: GAT outperforms GCN and GIN in money laundering detection due to its ability to identify complex patterns.

- **Successful Application:** GNNs help reduce false positives and improve regulatory compliance in financial institutions.

- **Synthetic Data from IBM:** Using synthetic data is a safe and effective way to develop GNN models.

- **Future Research:** Focus on improving GNNs for real-time financial environments and integrating them with other techniques.

- **Contribution to the Field:** The study makes a significant contribution to applying machine learning in finance and enhancing AML.

THANK YOU !