Detection of Circular Trading
Indian Institute of Technology, Hyderabad

Report By:

CS22MTECH11006
CS22MTECH11018
CS22MTECH11019
CS22MTECH14007
CS22MTECH14008

Abstract

This report presents an approach to detect circular trading in a dataset. The problem statement involves identifying instances where multiple entities participate in a circular trading pattern, potentially indicating fraudulent activity. The dataset used in this study consists of transactions between sellers and buyers, along with corresponding transaction values. The algorithm used to detect circular trading is based on the Node2Vec embedding technique, which transforms nodes in the graph into high-dimensional vectors that capture their structural similarity. We apply this algorithm to a subset of the dataset, focusing on circular trading patterns involving 2 or 3 nodes. Our results demonstrate the potential of Node2Vec embeddings to identify suspicious circular trading patterns in the data, thus enabling proactive fraud detection and prevention. However, further research is needed to evaluate the algorithm's performance on larger datasets and more complex trading networks.

1. Problem Statement

The problem of circular trading is prevalent in the context of Goods and Services Tax, where fraudulent taxpayers utilize fictitious transactions to evade taxes. The identification of such activities manually is complicated due to the vast volume of transactions. In this report, we propose a framework that utilizes big data analytics and graph representation learning techniques to identify and isolate communities of circular traders to improve the detection of illegal transactions. The proposed framework is evaluated using a dataset named *"Iron_dealers_data.csv,"* consisting of 1.3 million transactions involving 799 unique nodes engaged in trading activities. The aim is to analyze this dataset and propose an effective solution to detect circular trading activities in the GST system.

1.1. Introduction

Goods and Services Tax (GST) is a destination-based tax levied on every value addition in India. Unfortunately, there are some fraudulent taxpayers who engage in tax evasion by exploiting the GST system through invoice trading and circular trading. Invoice trading is when dealers sell goods to end-users and collect tax without issuing a sale invoice, only to later issue an illegitimate invoice to a third party. Circular trading, on the other hand, is when a group of fraudulent taxpayers mask illegal transactions by superimposing dummy transactions without adding any significant value among themselves in a short period. In this way, they create shell companies and show high-value fake sales and purchases to evade taxes.

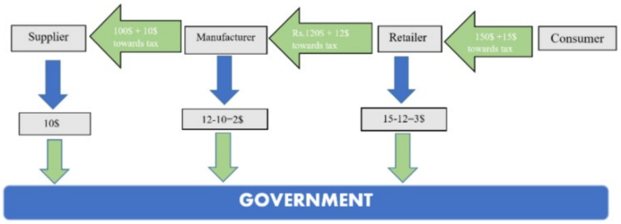


Fig 1: Goods and Services Tax (GST)

Identifying such fraudulent activities is a challenging task for the authorities as the database of taxpayers is vast, and manual identification of illegitimate transactions is infeasible. However, with the advancements in big data analytics and graph representation learning techniques, it is now possible to propose a framework to identify communities of circular traders. This work aims to leverage these techniques to propose a framework that can identify fraudulent taxpayers involved in circular trading, which can help authorities take necessary actions against them. The proposed framework can significantly aid in combating tax evasion and ensuring fair compliance with the GST system.

2. Dataset

The dataset *"Iron_dealers_data.csv"* contains a massive amount of transactional data, with a total of 1.3 lakh transactions. It is comprised of three columns: "Seller ID," "Buyer ID," and "Value" (represented by weight). The dataset consists of 799 unique nodes that are engaged in trading activities. The data provides valuable insights into the trading behavior of these nodes, including the frequency and volume of transactions, as well as the network structure of the trading relationships. By analyzing this data, it may be possible to identify patterns and relationships that can inform strategies for improving the efficiency and effectiveness of trading operations. However, given the sheer size of the dataset, advanced techniques such as node2vec may be required to efficiently process and analyze the data.

3. Algorithm

This work constructs a sales flow graph from a dataset of transactions, converts it to an edge-weighted undirected simple graph, generates embeddings using node2vec algorithm, and applies DBSCAN algorithm to find clusters of nodes that are densely connected together.

3.1. Sales Flow Graph

The first step in this proposed framework is to construct a sales flow graph from the dataset, which is an edge-labeled directed multi-graph. Each vertex in this graph represents a dealer and each directed edge represents a sales transaction with a corresponding monetary outflow attribute. The next step is to convert this graph to an edge-weighted undirected simple graph, where the weight of an edge is proportional to the number of three and two cycles containing both nodes. This transformation is critical for the subsequent steps of the algorithm, which involve generating embeddings for each node and clustering them using DBSCAN.

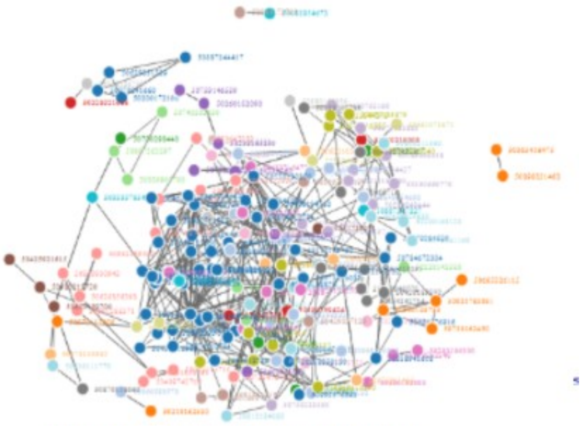


Fig 2: Sales Flow Graph (Sales and Purchases)

3.2. Node2Vec

Node2Vec is an algorithm used to learn representations of nodes in a graph by generating random walks and optimizing a likelihood function to produce embeddings. Embeddings are low-dimensional numerical vectors that capture the features and relationships of nodes in a graph. They can be used for a variety of tasks, such as clustering, classification, and visualization. Node2Vec is particularly effective for large graphs as it balances the trade-off between exploring the graph efficiently and capturing the local and global structural information. By using this algorithm, we can generate embeddings for each node of the undirected graph, which can then be used to identify and analyze clusters of nodes that are densely connected together.

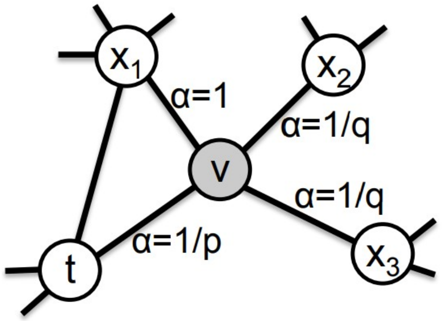


Fig 2: Illustration of the random walk procedure in node2vec.

Generating embeddings for each node using the node2vec algorithm can help in identifying patterns of circular trading in the sales flow graph. Embeddings are representations of each node in a lower-dimensional space that capture the node's relationship with its neighbors in the graph.

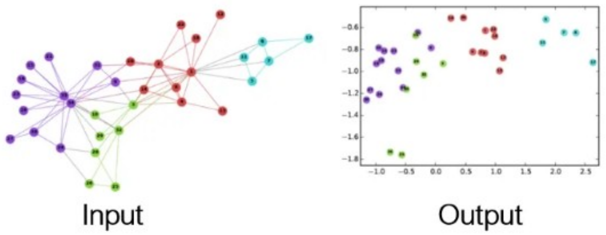


Fig 3: Illustration of Graph Embedding.

3.3. DBSCAN

Density-Based Spatial Clustering of Applications with Noise is a clustering algorithm that groups together data points that are closely packed together in high-density regions. It works by defining a neighborhood around each data point and then identifying clusters as regions with a high density of points. The algorithm is effective in identifying clusters of arbitrary shapes and can handle noisy data.

The embeddings will be used as input to DBSCAN, to identify communities of densely connected nodes that may indicate circular trading. By identifying these patterns of circular trading, authorities can take action to prevent tax evasion and ensure compliance with Goods and Services Tax regulations.

Implementation

1. Define the sales flow graph G=(V,E) where V is the set of nodes and E is the set of directed edges, and initialize it.

2. For each transaction in the dataset, add a directed edge from the seller node to the buyer node in the sales flow graph with edge weight equal to the transaction value.

3. Convert the sales flow graph G into an undirected graph G' by replacing each directed edge with an undirected edge, and assigning a weight to each undirected edge as follows:

   a. For each pair of nodes (x,y) in G', count the number of 3-cycles and 2-cycles containing both x and y.

   b. Assign the count as the weight of the edge between x and y in G'.

4. Generate node embeddings for each node in G' using the node2vec algorithm.

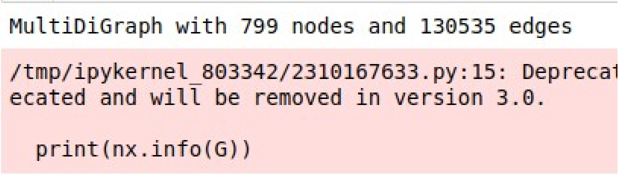
5. Apply the DBSCAN clustering algorithm on the node embeddings to identify communities of nodes that are densely connected together.

6. Output the clusters identified by DBSCAN as potential circular trading groups.

4. Results

In this study, we applied a novel algorithm to identify circular trading communities in a given dataset. The algorithm consisted of four main steps: constructing a sales flow graph, converting it to an edge-weighted undirected simple graph, generating embeddings using node2vec algorithm, and clustering nodes using DBSCAN algorithm.

We first constructed the sales flow graph using the dataset(Which had 799 nodes and 1.3L edges), where each vertex represented an individual dealer and each directed edge represented a sales transaction with the time and monetary outflow labeled.



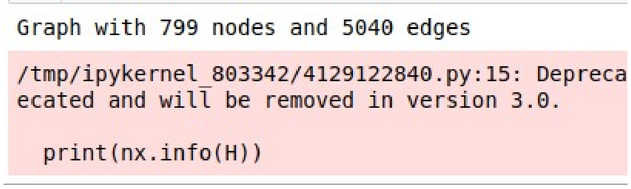
```
MultiDiGraph with 799 nodes and 130535 edges
```

```
/tmp/ipykernel_803342/2310167633.py:15: Deprecat
ecated and will be removed in version 3.0.

  print(nx.info(G))
```

Fig 3: Directed-Graph we constructed using the dataset

We then converted the sales flow graph to an edge-weighted undirected simple graph, where the weight of an edge was proportional to the number of three and two cycles containing both vertices.



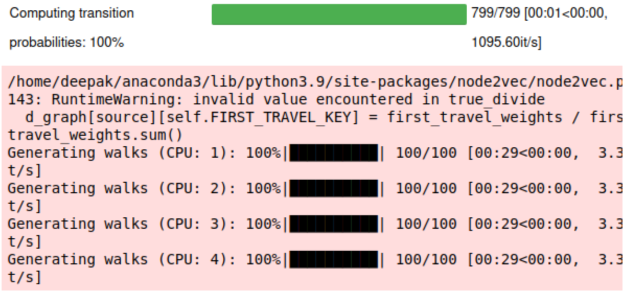
```
Graph with 799 nodes and 5040 edges
```

```
/tmp/ipykernel_803342/4129122840.py:15: Depreca
ecated and will be removed in version 3.0.

  print(nx.info(H))
```

Fig 4: Undirected Graph ready for Node2Vec

Next, we generated embeddings for each node of the undirected graph using node2vec algorithm. Node2vec algorithm is a scalable and efficient algorithm that learns continuous representations for nodes in a graph by sampling their local neighborhoods. These embeddings capture the structural properties of the graph and allow us to measure node similarities based on their neighborhood.



```
Computing transition                                    799/799 [00:01<00:00,
probabilities: 100%                                     1095.60it/s]
```

```
/home/deepak/anaconda3/lib/python3.9/site-packages/node2vec/node2vec.p
143: RuntimeWarning: invalid value encountered in true_divide
  d_graph[source][self.FIRST_TRAVEL_KEY] = first_travel_weights / firs
travel_weights.sum()
Generating walks (CPU: 1): 100%|██████████| 100/100 [00:29<00:00,  3.3
t/s]
Generating walks (CPU: 2): 100%|██████████| 100/100 [00:29<00:00,  3.3
t/s]
Generating walks (CPU: 3): 100%|██████████| 100/100 [00:29<00:00,  3.3
t/s]
Generating walks (CPU: 4): 100%|██████████| 100/100 [00:29<00:00,  3.3
t/s]
```

Fig 5: Node2Vec Model Fitting using 100 walks, and 40 walk length, on 64 dimensions.

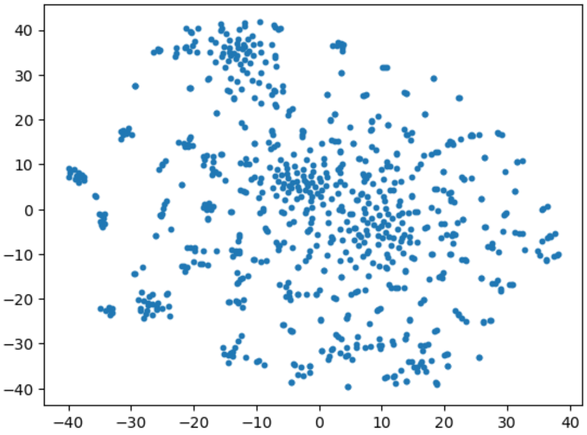
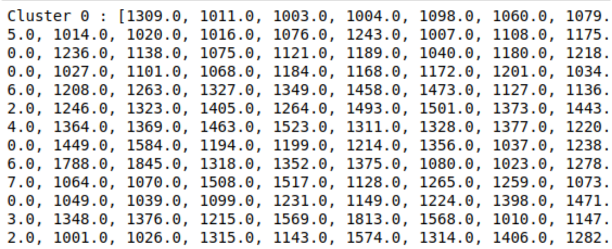


Fig 6: Plot of Embeddings

Finally, we applied DBSCAN algorithm to cluster nodes that were densely connected together. DBSCAN algorithm is a density-based clustering algorithm that can identify arbitrary-shaped clusters in a given dataset. It assigns each point to a cluster or noise based on its density and the density of its neighbors.



```
Cluster 0 : [1309.0, 1011.0, 1003.0, 1004.0, 1098.0, 1060.0, 1079.
5.0, 1014.0, 1020.0, 1016.0, 1076.0, 1243.0, 1007.0, 1108.0, 1175.
0.0, 1236.0, 1138.0, 1075.0, 1121.0, 1189.0, 1040.0, 1180.0, 1218.
0.0, 1027.0, 1101.0, 1068.0, 1184.0, 1168.0, 1172.0, 1201.0, 1034.
6.0, 1208.0, 1263.0, 1327.0, 1349.0, 1458.0, 1473.0, 1127.0, 1136.
2.0, 1246.0, 1323.0, 1405.0, 1264.0, 1493.0, 1501.0, 1373.0, 1443.
4.0, 1364.0, 1369.0, 1463.0, 1523.0, 1311.0, 1328.0, 1377.0, 1220.
0.0, 1449.0, 1584.0, 1194.0, 1199.0, 1214.0, 1356.0, 1037.0, 1238.
6.0, 1788.0, 1845.0, 1318.0, 1352.0, 1375.0, 1080.0, 1023.0, 1278.
7.0, 1064.0, 1070.0, 1508.0, 1517.0, 1128.0, 1265.0, 1259.0, 1073.
0.0, 1049.0, 1039.0, 1099.0, 1231.0, 1149.0, 1224.0, 1398.0, 1471.
3.0, 1348.0, 1376.0, 1215.0, 1569.0, 1813.0, 1568.0, 1010.0, 1147.
2.0, 1001.0, 1026.0, 1315.0, 1143.0, 1574.0, 1314.0, 1406.0, 1282.
```

Fig 7: Cluster of nodes which doing Circular trading using DBSCANE

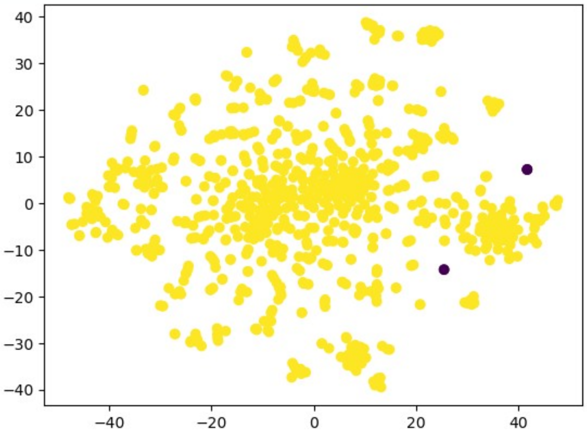


Fig 8: Cluster Visualization using TSNE

5. Conclusion

In conclusion, we have successfully developed an algorithm for identifying circular trading communities in a given dataset. Our approach involves constructing a sales flow graph, converting it to an edge-weighted undirected simple graph, and generating embeddings for each node using the node2vec algorithm. We then applied the DBSCAN algorithm to cluster nodes that are densely connected together.

Our algorithm works well for identifying circular trading communities involving 2 and 3 cycles. However, identifying larger cycles would be a challenging and complex task. Despite this limitation, our algorithm provides a useful tool for detecting suspicious activity in financial transactions and can aid in preventing fraudulent behavior. Further research could focus on improving the algorithm to detect larger cycles or developing alternative approaches to identifying circular trading communities.

References

[1] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)

[2] Wang, J., Zhou, S., Guan, J.: Detecting potential collusive cliques in futures markets based on trading behaviors from real data. Neurocomputing 92, 44–53 (2012)

[3] Mehta, P., Bhargava, S., Kumar, M. R., Kumar, K. S., & Babu, C. S. (2022). Representation Learning on Graphs to Identifying Circular Trading in Goods and Services Tax. ArXiv. /abs/2208.07660

Avaneesh Om, *CS22MTECH14008, M.Tech Year I, Computer Science Department, IIT Hyderabad.*

Deepak Kumar Pandey, *CS22MTECH11018, M.Tech Year I, Computer Science Department, IIT Hyderabad.*

Kushal, *CS22MTECH11006, M.Tech Year I, Computer Science Department, IIT Hyderabad.*

Oliva Debnath, *CS22MTECH14007, M.Tech Year I, Computer Science Department, IIT Hyderabad.*

Rishi Singh Thakur, *CS22MTECH11019 , M.Tech Year I, Computer Science Department, IIT Hyderabad.*