Representation Learning on Graphs to Identifying Circular Trading in Goods and Services Tax

1st Priya Mehta
Welingkar Institute of Management Development and Research
Mumbai, India
priya.mehta@welingkar.org

3rd Sandeep Kumar
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Sangareddy, India
sandeep@iith.ac.in

5th Ch. Sobhan Babu
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Sangareddy, India
sobhan@cse.iith.ac.in

2nd Sanat Bhargava
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Sangareddy, India
sanat@iith.ac.in

4th Ravi Kumar
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Sangareddy, India
CS18MTECH11028@iith.ac.in

Abstract—Circular trading is a form of tax evasion in Goods and Services Tax where a group of fraudulent taxpayers (traders) aims to mask illegal transactions by superimposing several fictitious transactions (where no value is added to the goods or service) among themselves in a short period. Due to the vast database of taxpayers, it is infeasible for authorities to manually identify groups of circular traders and the illegitimate transactions they are involved in. This work uses big data analytics and graph representation learning techniques to propose a framework to identify communities of circular traders and isolate the illegitimate transactions in the respective communities. Our approach is tested on real-life data provided by the Department of Commercial Taxes, Government of Telangana, India, where we uncovered several communities of circular traders.

Index Terms—Graph Representation Learning, Graph Clustering, Fraud Detection, Circular Trading, Goods and Services Tax, Value Added Tax

I. INTRODUCTION

Taxes can be broadly classified into two categories, based on how they are collected, namely, direct taxes and indirect taxes. This work proposes an approach to identify groups of taxpayers (traders) that engage in evasive activity called circular trading in the indirect taxation system. In indirect taxation systems, the tax is imposed successively at every point of the value chain, and the amount of tax levied is contingent on the value added to goods at the corresponding point in the value chain. Tax is levied at each point of the value chain, such that tax paid on the requisite goods and services (input tax) for that point offsets the tax levied on the sales (output tax). Figure 1 shows how the tax is collected incrementally in this system.

In this example, the manufacturer purchases some raw material of value $1000 from the raw material dealer by paying $100 as tax at a rate of 10%. The raw materials dealer remits this tax to the government. The retailer purchases the processed goods from the manufacturer for, say, $1200 and pays $120 to the manufacturer as tax at the same rate of taxation. The manufacturer pays the government the difference between the tax he had collected from the retailer and the tax he has paid to the raw materials producer, i.e., $120 - $100 = $20. The consumer then buys the finished goods from the retailer for $1500 by paying a tax of $150. Following the same argument, as given in the previous steps, the retailer pays the government $30(i.e.,$150 - $120).

It is easy to see that the total tax received by the government is $150, and the entirety of the tax burden is indirectly shifted onto the consumer of goods. Hence, the raw materials dealer, the manufacturer, and the retailers collect the tax on behalf of the government.

A. Circular Trading in GST and VAT

In most of the detected tax evasion cases, business dealers (taxpayers) defrauded the authorities by deliberately manipulating the actual business transactions in their tax returns to maximize the amount of profit gained by evading tax. Invoice trading is a tax evasion method, in which a dealer sells their goods to the end-user and collects the corresponding tax without issuing an invoice of sale. Later, they issue an illegitimate invoice to a third party, implying that the end-user’s tax is
Most of the prior work focuses on detecting circular trading in the stock market. In [1], a graph clustering algorithm is presented for detecting collusion sets in the stock market using Dempster–Schafer’s theory of evidence. In [2], authors proposed a method to detect potential collusive cliques by computing the coefficient of correlation between two suitable unified aggregated time series of signed order volume and then combining the connected components from multiple sparsified weighted graphs constructed by using correlation matrices where each correlation coefficient is higher than a threshold specified by the user. In [3], authors proposed an approach to detect collusion sets using Markov Clustering Algorithm. This method can detect purely circular collusion as well as cross-trading collusion. In [4], authors proposed the Bottleneck edge computation algorithm, which is applied to remove cycles from circular trading communities provided that the communities are known beforehand. Machine learning tasks over nodes and edges in graph require careful effort in extracting features used by machine algorithms. In machine learning, feature learning or representation learning is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. In [5], authors proposed node2vec algorithm, which learn a mapping of nodes to a low-dimensional space. In [6], authors provided a conceptual review of key advancements in representation learning on graphs, including matrix factorization-based methods, random-walk based algorithms, and graph convolutional networks.

### III. Description of Data Set and Graph Construction Procedure

The Goods and Services Tax Return 1 (GSTR-1) is a document that each registered tax payer needs to file every month/quarter. It must contain the details of all sales and supply of goods and services made by the tax payer during the tax period. We have taken GSTR-1 invoices data from March 2021 to December 2022. Number of invoices are 77837538, number of taxpayers are 388448 and size of the data set is 7GB.
Table I depicts a few fields of the sales transactions data set (GSTR-1) used. Construct an edge labelled directed multigraph called sales flow graph using this data set. Each vertex in this graph corresponds to an individual dealer. Each directed edge denotes one sales transaction in the data set. Each edge contains two attributes, where the first one is the time of sales of the corresponding transaction and the second one is the monetary outflow associated with this transaction. Due to the complexity and the size of this graph, community detection is a non-trivial task.

IV. IDENTIFYING COMMUNITIES IN SALES FLOW GRAPH

A. Node2vec

Node2vec is a semi-supervised algorithm which assigns an embedding (essentially a vector in n-dimensional space) for every node in the graph, such that the relative position of the node is preserved. These embeddings allow us to quantify the similarity or the dissimilarity between nodes, which is crucial for clustering nodes of the graph. It is worth noting that by tuning the hyperparameters appropriately we can get node embeddings such that the nodes which belong to the same neighbourhood, have similar embeddings. The quality of embeddings is influenced by the choice of following hyperparameters.

- **Return parameter (p):** Controls the probability of revisiting a node. We set p=1 to ensure moderate exploration and steer clear of two-hop redundancy.
- **In-out parameter (q):** Controls the bias towards "inward" and "outward" nodes. We set q=1/2 to increase the likelihood of visiting nodes further away from the current node. Our choice of p and q ensures a Depth-First Sampling of the neighbourhood.
- **Length of RandomWalk:** The number of nodes to traverse from the current node to sample probabilities of every node in the neighbourhood of the current node.
- **Number of RandomWalks:** Number of Random Walks to carry out for each node.
- **Dimensions:** Number of dimensions of the vector space containing the embedding vectors. Conventionally, this value is taken to be $\sqrt{\text{number of nodes}}$.

B. Clustering using DBSCAN

Once we have obtained the embeddings for every node, we find densely connected nodes within the graph. Mathematically, this reduces to clustering embedding vectors that are close to each other in n-dimensional space. For this, we use DBSCAN algorithm to find dense regions within the vector space using cosine similarity measure. As with node2vec, the effectiveness of clustering depends on the choice of hyperparameters. Two important hyper parameters required for DBSCAN are $\text{epsilon}$ ("eps") and minimum points ("MinPts"). The parameter $\text{eps}$ defines the radius of neighborhood around a point. The parameter $\text{MinPts}$ is the minimum number of neighbors within "eps" radius.

C. Algorithm to find communities

1) Create an edge labelled directed multi graph from the sales transactions data set as explained in Section III.
2) Convert the above graph to an edge-weighted undirected graph as explained in [7].
3) Generate embeddings for each node of the undirected graph using node2vec algorithm.
4) Apply DBSCAN algorithm to find clusters of nodes that are densely connected together.

V. RESULTS AND CONCLUSION

The Figure 3 shows the clusters obtained after applying the DBSCAN algorithm. The Figure 4 shows one cluster in detail. Here, we present a state-of-the-art methodology to obtain communities from a large and convoluted graph. Once we have obtained the communities, we can use the algorithms described in [4] to isolate fraudulent transactions within the community.

ACKNOWLEDGMENT

We express our sincere gratitude to the Telangana state Government, India, for sharing the commercial tax data set, which is used in this work.

REFERENCES

[1] G. K. Palshikar and M. M. Apte, “Collusion set detection using graph clustering,” Data mining and knowledge Discovery, vol. 16, no. 2, pp. 135–164, 2008.
[2] J. Wang, S. Zhou, and J. Guan, “Detecting potential collusive cliques in futures markets based on trading behaviors from real data,” *Neurocomputing*, vol. 92, pp. 44–53, 2012.

[3] M. N. Islam, S. R. Haque, K. M. Alam, and M. Tarikuzzaman, “An approach to improve collusion set detection using mcl algorithm,” in *2009 12th International Conference on Computers and Information Technology*. IEEE, 2009, pp. 237–242.

[4] P. Mehta, J. Mathews, S. B. Ch et al., “A graph theoretical approach for identifying fraudulent transactions in circular trading,” 2017.

[5] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[6] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications,” *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017. [Online]. Available: http://sites.computer.org/debull/A17sep/p52.pdf

[7] P. Mehta, J. Mathews, K. S. Kumar, K. Suryamukhi, C. S. Babu, and S. V. K. V. Rao, “Big data analytics for nabbing fraudulent transactions in taxation system,” ser. Lecture Notes in Computer Science, K. Chen, S. Seshadri, and L. Zhang, Eds., vol. 11514. Springer, 2019, pp. 95–109.