आई आई टी हैदराबाद
IIT Hyderabad

# CS6890
# Fraud Analytics

---

# Trust Rank

---


V Harikrishnan
CS23MTECH11008

Suryansh Gautam
CS23MTECH11020

Patel Heetkumar D.
CS23MTECH11029

Anil kumar Sharma
CS23MTECH13001

KR Anuraj
CS23MTECH13002

# 1. Problem Statement

Financial transactions take place in complex networks where reputation and trust are vital components. The truthfulness of these networks depends on the detection of dishonest or malicious operators, or ”bad senders.” In this assignment, we evaluate senders' trustworthiness based on their transaction history by utilizing the Trust Rank algorithm. Our objective is to reliably and efficiently identify malicious senders.

Directly identifying bad senders without any prior information can be difficult when analyzing financial transactions. The dataset might not be able to clearly identify instances of dishonest behavior or malevolent intent on its own. Because of this, we start with limited knowledge in order to identify more problematic senders from the final dataset.

# 2. Description of the Dataset

Two **Datasets** are given :

1. Payment : It consists of payment infromation where each row is a transaction described by sender, receiver, amount.
2. Bad Sender : It consists of subset of bad senders.

# 3. Algorithm Used

For this assignment, we are using modified version of Trust Rank Algorithm where instead of calculating trust score for the different scores we calculate fraud score.

The **Input** of the algorithm are :

1. Graph Form : It is a directed graph form taken from different transactions from the payment dataset.
2. Array : Used to intialized the scores corresponding all the nodes given in the bad-senders dataset as one by number of bad-senders in that datasets and all other values as zero.
3. No of bad senders : No of senders listed in the bad-sender dataset.

The **Parameters** that are used in the algorithm are :

1. Epsilon : It is used to check whether difference from the scores from the previous iteration to the current iteration is more than a threshold.
2. Max Iteration : This is the another condition we imposed so that we loop over the graph to update the scores for limited number of times.
3. Alpha : This is a decay factor to introduce bias towards the given bad senders.

**Algorithm Trust Rank**

**Input :** directed graph g, bad_senders_arr [ ], no_bad_senders

**Output :** i_fraud_score [ ]

1. Create an i_fraud_score [ ]. Initialize it with the bad_senders_arr [ ].
2. Initialize an integer variable i to keep a track of number of iterations and initialize epsilon with 0.0001, max_itr to 100, no_bad_senders as number of bad senders in bad-senders dataset and alpha with 0.85.
3. Initialize a variable diff as a large positive number.
4. **while** diff $>$ epsilon and i $>$ max_itr
   (a) Initialize the n_fraud_score [ ] of length equal to no of nodes in the graph with zero.
   (b) **for** n in nodes in the graph g.
      i. **for** e in edges which originates from node n.
         A. Let h be the head of edge e and t be the tail of the edge e.
         B. n_fraud_score [h] = n_fraud_score[head] + (weight of e * i_fraud_score [t])
         **end for**
      **end for**
   (c) diff = max(abs(i_fraud_score - n_fraud_score))
   (d) Initialize a variable forgot_score = (1 - sum(n_fraud_score)) / no_bad_senders.
   (e) Initialize forgotten_score [ ] by distributing forgot_score to all the bad senders in bad_senders dataset.
   (f) n_fraud_score [ ] = n_fraud_score [ ] + forgotten_score [ ]
   (g) i_fraud_score [ ] = (alpha * n_fraud_score [ ]) + ((1 - alpha) * bad_senders_arr [ ])
   **end while**
5. return i_fraud_score [ ]

# 4. Results

Our algorithm gives fraud scores to all the nodes in the graph. Higher scores implies more probability of the node being fraud. Also, we want the original bad senders to attain the higher score. We use a minimum scoring node which was already present in the original bad senders as a threshold to find which other nodes in the graph are fraud.

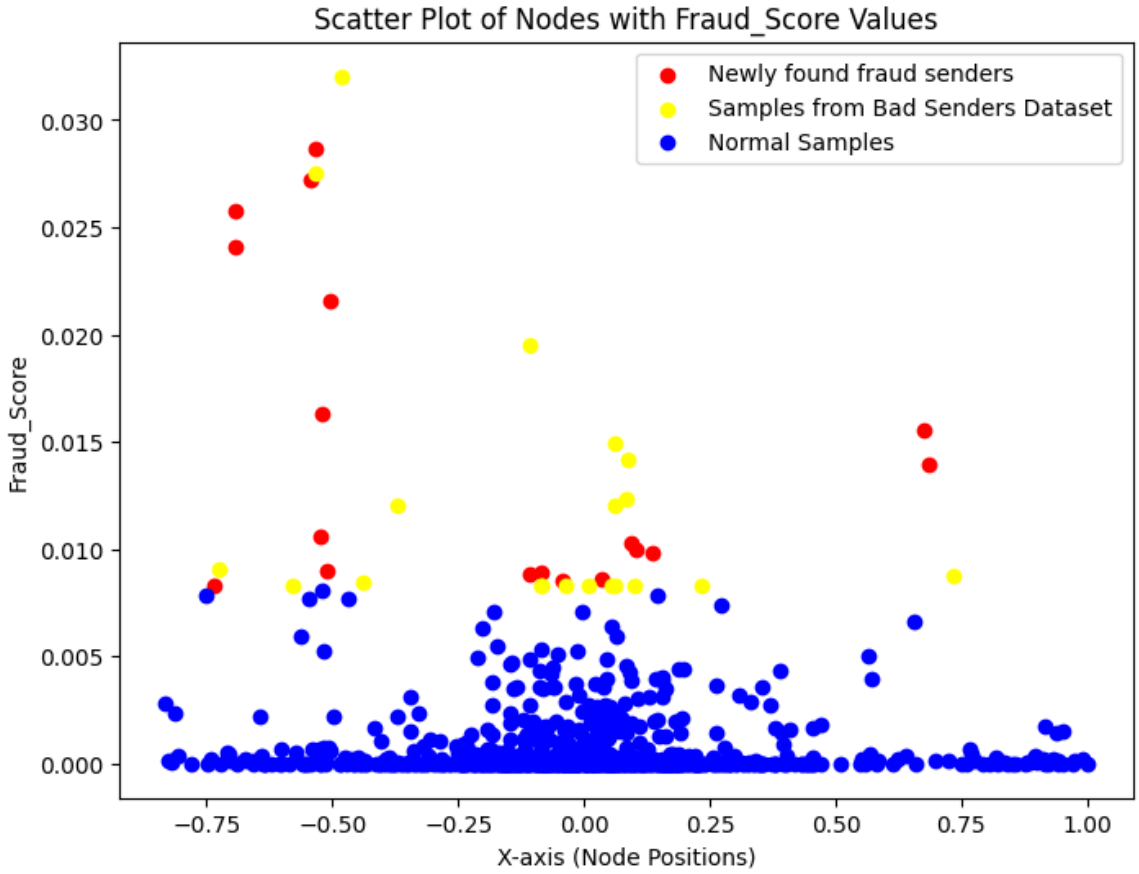


Figure 1: Final Output

Above Figure 1 shows the scatted plot of the final output where Blue points are showing the normal samples, Yellow points are showing the samples that are given in the bad_senders dataset and the Red points are showing the newly found fraud senders from the payment dataset.