CS6890 Fraud Analytics Using Predictive and Social Network Techniques

---

Computing Trust Rank Assignment 1 Report

Group Members:

- **CS24MTECH14003** Anurag Sarva
- **CS24MTECH14006** Gulshan Hatzade
- **CS24MTECH14018** Mainak Adhikari

---

1. Introduction

In modern digital transaction networks, identifying fraudulent activities is a critical challenge. **Trust Rank**, an adaptation of the **PageRank algorithm**, is used to evaluate the trustworthiness of entities within a system.

This assignment focuses on implementing a **DistrustRank model** using the **Pregel framework**, which efficiently computes **trust propagation** across a transaction network.

The objective of this project is to compute **weighted trust scores** for entities involved in transactions, ensuring that **high-value transactions contribute more** to trust calculations. The model also integrates a **Pregel-based distributed framework**, enhancing **scalability and efficiency** for processing large datasets.

---

2. Problem Statement

Given the datasets:

1. **Payments.xlsx** – Contains records of transactions between entities.
2. **bad_sender.xlsx** – Contains a list of fraudulent entities.

The goal is to:
Build a transaction graph, where each node represents an entity, and each edge represents a financial transaction.
Compute trust scores using a **Pregel-based parallel processing framework**.
Implement a weighted approach, where **trust scores are influenced by transaction**

values.
Visualize trust score distribution using **histograms**.

---

3. Dataset Information

This project uses **two datasets** to build a transaction network and compute trust scores.

3.1. Transactions Dataset (`Payments.xlsx`)

This dataset contains **130,535 transaction records** with **three columns**:

Column Name	Description
Sender ID	Unique identifier for the sender of the transaction
Receiver ID	Unique identifier for the receiver of the transaction
Transaction Amount	The monetary value transferred from sender to receiver

- Each row in the dataset represents a **directed transaction** between two entities.
- This data is used to **construct a graph**, where **nodes represent users** and **edges represent financial transactions** weighted by the transaction amount.

3.2. Malicious Senders Dataset (`bad_sender.xlsx`)

This dataset contains **20 unique fraudulent nodes** that have been **flagged as malicious**.

- These fraudulent nodes serve as **initial seed points** for the **DistrustRank algorithm**, influencing **trust propagation**.
- The fraudulent nodes are assigned **initial distrust scores**, which then **propagate through the network** during computation.

3.3. Unique Node Count

- **Total Unique Nodes: 799**
- **Total Bad Nodes (Fraudulent Entities): 20**

This structured dataset helps in initializing **DistrustRank**, ensuring that **malicious nodes are correctly identified**, and trust scores are computed effectively.

---

4. Methodology

4.1. Data Preprocessing

- **Loading Data**: The transaction data from `Payments.xlsx` and fraudulent senders from `bad_sender.xlsx` are read using Pandas.
- **Node Mapping**:
  Each **unique entity (sender or receiver)** is assigned a **unique index** for fast processing.
  Fraudulent senders are **mapped to their respective indices**.
- **Transaction Parsing**:
  The **sender-receiver pairs** form **directed edges** in a **graph**.
  The **transaction amount** is used as the **edge weight**, ensuring **transactions with higher monetary value contribute more to trust propagation**.

4.2. Graph Construction

The **network graph** is structured as follows:
* **Nodes:** Represent individual entities (senders and receivers).
* **Edges:** Represent financial transactions, **weighted by transaction amounts**.
* **Fraudulent Nodes:** Nodes listed in `bad_sender.xlsx` are **flagged for special processing**.

Edge weight calculation:

$$w_{(i,j)} = \frac{\text{transaction amount}}{\sum \text{transactions from } i}$$

where $w_{(i,j)}$ represents the **proportion of trust transferred** from node i to node j.

4.3. Trust Score Initialization

* **All nodes** start with an **initial trust score of 0**.
* **Fraudulent nodes** are assigned an **initial score based on the number of malicious entities** to simulate **trust leakage** in the system.

$$T_i = \frac{1}{\text{total fraudulent nodes}}, \quad \forall i \in \text{fraudulent nodes}$$

4.4. DistrustRank Algorithm (Pregel Implementation)

The algorithm follows a **message-passing approach**, where **trust scores are exchanged iteratively between nodes**.

Trust Score Computation

Each node updates its trust score as:

$$T_{\text{new}} = \text{damping factor} \times \sum_{\text{incoming nodes}} w_{(i,j)} \times T_{\text{old}} + (1 - \text{damping factor}) \times T_{\text{initial}}$$

where:

- **Damping Factor** $(\alpha = 0.85)$ ensures that trust leakage is accounted for.
- **Incoming nodes** contribute their trust **proportionally based on edge weights**.

Parallel Processing with Pregel

* **Workers are assigned subsets of nodes**, ensuring balanced workload distribution.
* **Each worker processes trust updates independently**, improving efficiency.
* **Nodes continue to exchange trust scores** until a **convergence threshold or max iterations (100) is reached**.

---

5. Results and Visualization

5.1. Computed Trust Scores

- The **final computed trust scores** are stored in `DistrustRank_Results.txt`.
- Nodes with **higher trust scores** represent entities with **strong financial influence**, whereas **low trust scores** indicate potential **fraudulent activity**.

5.2. Trust Score Histograms

Figure 1: Distribution of Trust Rank Values

* **File Name:** `DistrustRank_Distribution.png`

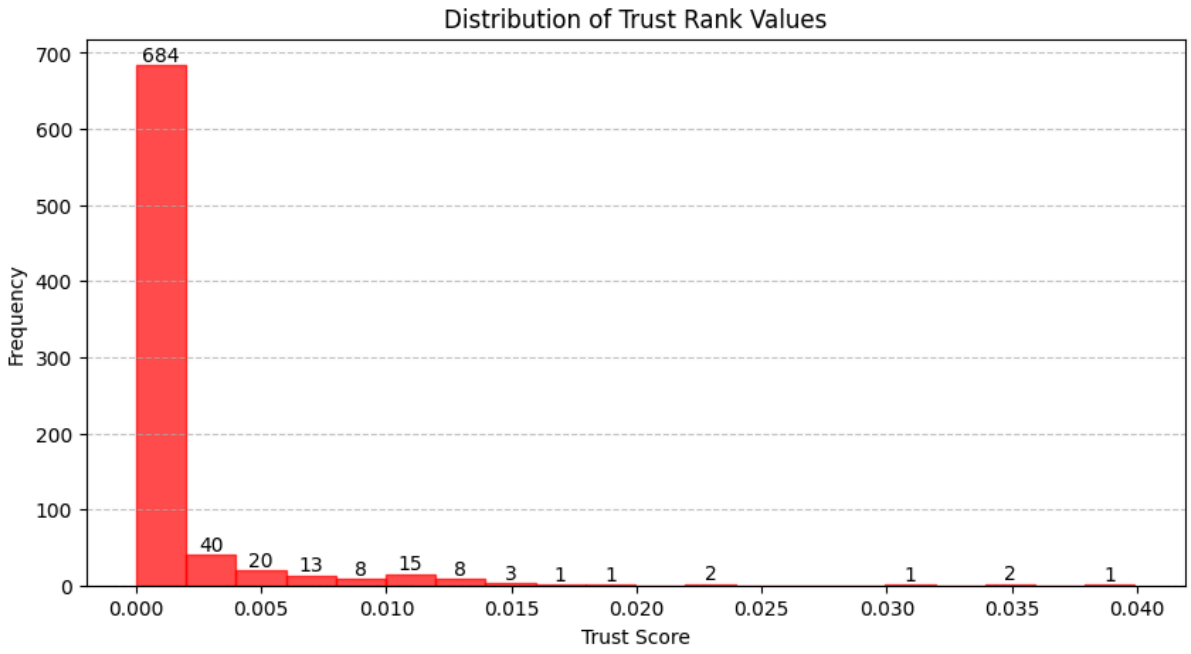

Figure 1: Distribution of Trust Rank Values

This figure represents the **distribution of trust rank values** across all nodes. The **X-axis denotes trust scores**, while the **Y-axis represents frequency (number of nodes with that trust score)**.

Figure 2: Normalized Trust Rank Distribution

* **File Name:** `Normalized_TrustRank_Distribution.png`

This figure shows the **probability density distribution** of trust ranks, ensuring that the **area under the graph sums to 1** instead of displaying raw frequency counts. This approach allows for **better comparison across datasets** by **normalizing values**.

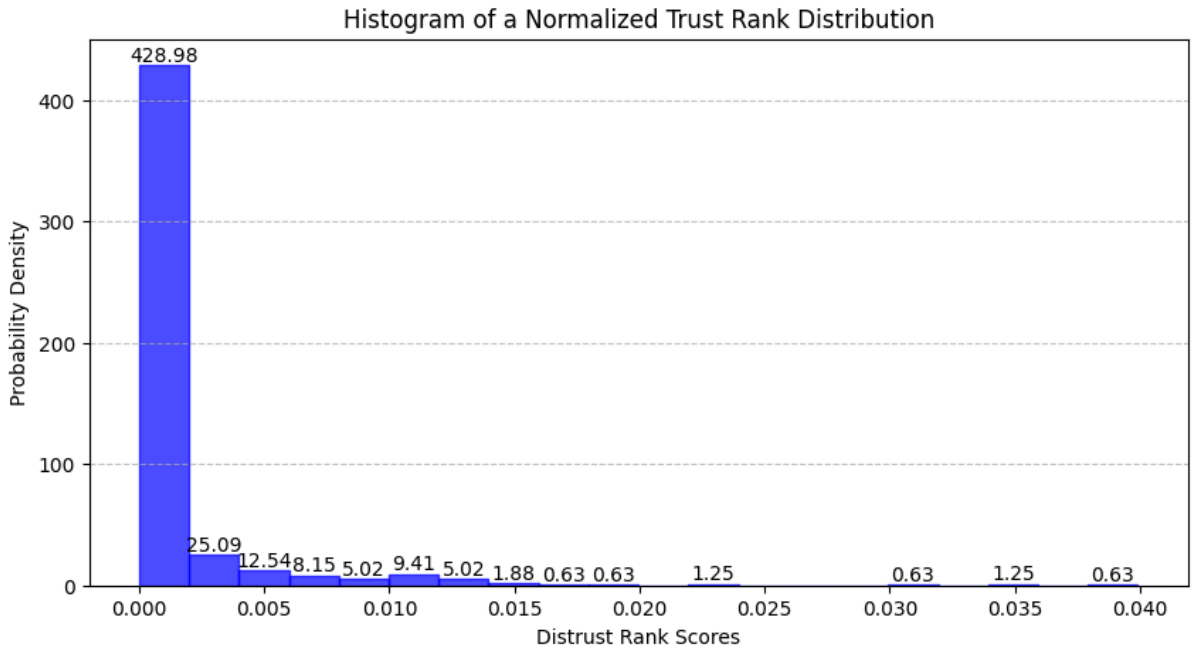

Figure 2: Normalized Trust Rank Distribution

This figure shows the **probability density distribution** of trust ranks, ensuring that the **area under the graph sums to 1** instead of displaying raw frequency counts. This approach allows for **better comparison across datasets** by **normalizing values**.

---

6. Conclusion

This project successfully implements **DistrustRank using the Pregel framework**, demonstrating:

Scalability: The Pregel-based distributed computation efficiently processes large-scale transaction networks.
Fraud Detection: The algorithm effectively identifies fraudulent nodes with low trust scores.
Weighted Trust Propagation: Transactions with higher values contribute more to the trust computation.
Graph-Based Visualization: The histogram provides insights into the trust distribution across the network.