**REPORT ON NETWORK ANALYSIS OF TRUST-BASED TRANSACTIONS IN THE BITCOIN OTC NETWORK**

Table of Contents

[Introduction 2](#_Toc193448532)

[Data Source 2](#_Toc193448533)

[Tools 2](#_Toc193448534)

[Data Loading & Preprocessing 2](#_Toc193448535)

[Research Questions and Methodologies 3](#_Toc193448536)

[Network Analysis Results 4](#_Toc193448537)

[Conclusion and Recommendations 8](#_Toc193448538)

# Introduction

In online financial ecosystems, user interactions and transaction security are greatly influenced by reputation and trust. Having knowledge about the development of trust and mistrust in these kinds of networks can help identify possible fraud trends, user conduct, and systemic hazards. With a focus on trust dynamics, network topologies, and reconciliation patterns, this study examines transaction and rating data from a financial platform that employs Bitcoin.

Aim: Through the use of network analysis techniques, **we want to find hidden connections, spot irregularities, and pinpoint important trends that affect user engagement and platform stability within the Bitcoin OTC.**

This report presents a structured analysis, including dataset descriptions, research questions, methodologies, and findings derived from network metrics and visualization techniques.

The objective is to:

1. study how trust is distributed, user interactions and transaction relationships over time.
2. interpret the pattern of trust-based transactions and their implication for financial reconciliation processes.

# Data Source

The dataset used for this analysis originates from a Bitcoin-based transaction platform, where users provide trust or distrust ratings for each other. The dataset was sourced from **SNAP (Stanford Large Network Dataset Collection)** dataset which is publicly available financial transaction records and has been pre-processed to remove personally identifiable information (PII), ensuring compliance with ethical data usage guidelines. This analysis was conducted as part of a task from the HNG Tech internship program, providing hands-on experience in real-world network analysis and data-driven decision-making.

# Tools

1. **Python**: The primary programming language used for data handling and analysis.
2. **Pandas**: For data manipulation and preprocessing.
3. **NetworkX**: To construct and analyse the trust network.
4. **Matplotlib & Seaborn**: For data visualization and exploratory analysis.
5. **Seaborn**: To create visuals of network graphs.
6. **Jupyter Notebook**: For implementing and documenting the entire workflow.

# Data Loading & Preprocessing

1. **Loading the Dataset**  
   The dataset was loaded into a Pandas DataFrame from a CSV file, ensuring proper encoding and parsing of timestamps.
2. **Handling Missing Values**  
   The dataset was thoroughly examined for missing or incomplete entries, and no missing values were found. As a result, no imputation techniques or record removals were necessary, ensuring that the dataset remained intact for analysis.
3. **Constructing the Trust Network**  
   A directed graph was built using NetworkX, where:
   * 1. Nodes represent users.
     2. Edges represent trust or distrust ratings.

This preprocessing stage ensures that the data is clean, structured, and ready for advanced network analysis techniques. The next sections will delve into the research questions, methodologies, and insights derived from this analysis. For a detail analysis and breakdown of the code used in each of the preprocessing step, refer to appendix 1.

# Research Questions and Methodologies

1. What is the distribution of trust scores in the Bitcoin OTC network?

The dataset is examined using statistical metrics including mean, median, standard deviation, skewness, and kurtosis in order to answer the research question. Histograms chart and kernel density estimation (KDE) plots are used to map the distribution of trust ratings in addition to these measures.

The research question aim is to ascertain whether trust scores show skewed or clustered patterns or follow a normal distribution. Also, give insight to systemic biases or polarization in user behaviour, such as disproportionate trust accumulation among particular user groupings, by measuring the prevalence of high-trust versus low-trust ratings.

1. How do high-trust and low-trust transactions differ in terms of network structure?

The predetermined thresholds are used to classify transactions into low-trust (<-5) and high-trust (>5) groups. Through degree distributions, clustering coefficients, and connectivity patterns, the structural distinctions between these groupings are investigated. Using an optimal parameter called modularity, which gauges the density of links between the communities, Louvain algorithms are used to find core communities of trusted users.

This method demonstrates how high-trust interactions show increased centrality and dense subgraphs, whereas low-trust transactions form isolated clusters with weak connection. For detail analysis and breakdown of the code used to implement the Louvain algorithm refer to appendix 2.

1. What role do central users play in transaction reconciliation, and how does their influence change over time?

**PageRank, betweenness, and degree are centrality measurements**. A node's degree centrality indicates how many direct connections it has within the network. The frequency with which a user serves as a bridge or intermediary along the shortest paths between other nodes is measured by betweenness centrality. A node's pagerank is determined by how important the nodes it is connected to are. These are all calculated to determine who are the ecosystem's key users.   
High-centrality users' transaction histories are examined for patterns of mediation, and temporal analysis monitors changes in their centrality ratings over time periods. According to the analysis, central users serve as crucial go-betweens, promoting the spread of trust and resolving disputes. As the network decentralizes or consolidates during times of increased transactional risk, their impact may eventually wane, reflecting changing dynamics of trust.

1. How does transaction volume and trust score evolve over time?

To examine temporal trends, the dataset is divided into monthly and quarterly intervals. To identify variations, line charts and moving averages are used to plot transaction volume and average trust scores.

This methodology makes it clear whether trust stability and transaction frequency are related. For example, cyclical or event-driven trust patterns may be revealed when high activity times coincide with unstable trust scores brought on by platform stress or outside market forces.

1. Do some users frequently engage in inconsistent or fluctuating trust ratings?

Standard deviations (measures the amount of variation or dispersion) of received scores are used to identify users with unstable trust ratings. Users are categorized into stability groups (e.g., "stable," "moderate," and "volatile") using rolling-window analysis and clustering algorithms. Because uneven trust signals may indicate unresolved arguments or antagonistic behaviour, compromising network reliability, the study hypothesizes that users with inconsistent ratings may contribute to reconciliation inefficiencies.

1. Are transactions with lower trust scores more likely to involve users with fewer connections?

Pearson correlation analysis is used to assess the association between trust scores and node degree, and scatter plots are used to show the results. Results reveals those peripheral users those with fewer connections are more vulnerable to low-trust interactions, pointing to possible weaknesses in the edge structure of the network and guiding the development of initiatives to improve user onboarding procedures.

# Network Analysis Results

In order to answer the Research questions, this section provides the network analysis's findings together with important graph metrics, visualizations, and statistical insights. The results show how the Bitcoin OTC trust network is structured and how its reconciliation patterns work.

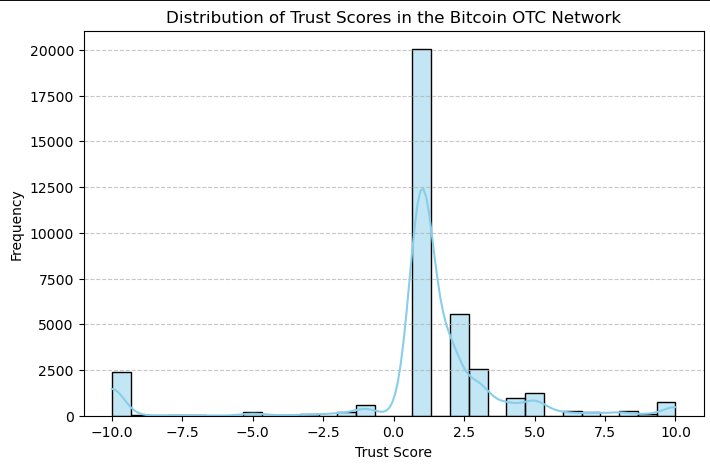


Figure 1: Distribution of Trust Scores in the Bitcoin OTC Network

From the visual it can be observed that the concentration around **zero** suggests that **trust is not universally established**, and many transactions occur without strong opinions on trustworthiness. The presence of **extreme negative scores** may indicate fraudulent activities or unresolved disputes in the network. The dominance of **positive ratings** over negative ones could suggest that once trust is established, it is reinforced over time.

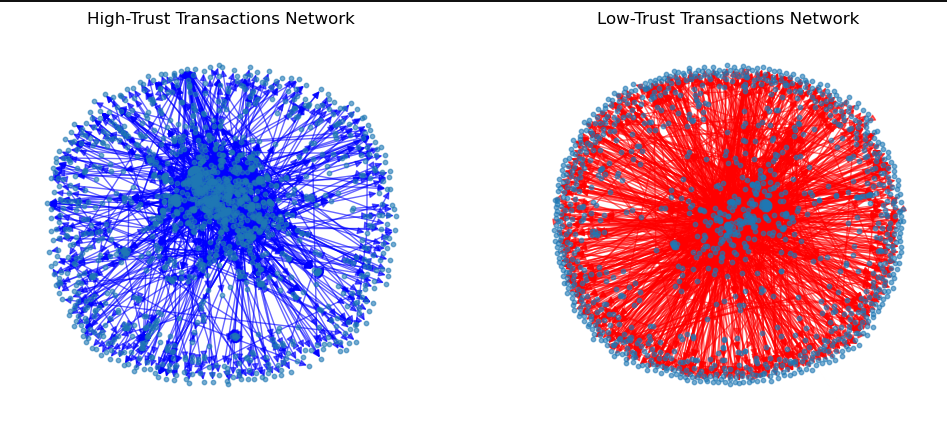


Figure 2:Structural Differences in High-Trust vs. Low-Trust Transactions: A Cohesive Core vs. a Dispersed Web of Distrust

The visualization compares the network structure of **high-trust transactions (left, blue)** and **low-trust transactions (right, red)** in the Bitcoin OTC network.

* The high-trust network exhibits **a dense core**, where well-connected users interact frequently, indicating a strong and cohesive trust structure. The edges (transactions) are relatively balanced, suggesting mutual confidence in transactions.
* The low-trust network, in contrast, appears **more dispersed yet highly interconnected**, with a large number of connections. This suggests widespread distrust but also implies that low-trust users still actively engage in transactions, potentially forming clusters of unreliable interactions.

Further analysis using the Louvain algorithm to identify inner trust communities reveals the size of high trust communities is twice as large compare to the low trust communities.

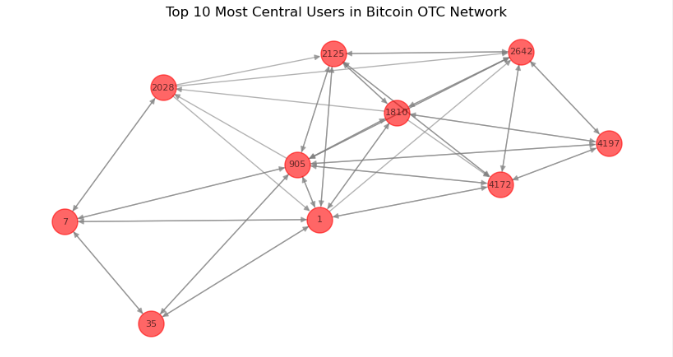


Figure 3:Key Players in the Bitcoin OTC Network: Top 10 Most Central Users and Their Connections

This network graph highlights the directional relationship and transaction strength, revealing key players based on centrality measures. Thicker edges indicate a higher volume of transactions or stronger trust relationships between certain users.

The central users (user 1, **Users 2125, and 2642**) identified in this network likely play a crucial role in facilitating transactions, possibly **acting as intermediaries** between less connected users. Their influence could stem from **high trust ratings**, making them reliable participants, or from **high transaction volume**, positioning them as dominant figures in the network. Meanwhile, users (**Users 7 and 35)** on the periphery may have **limited influence**, interacting selectively with the core network members.

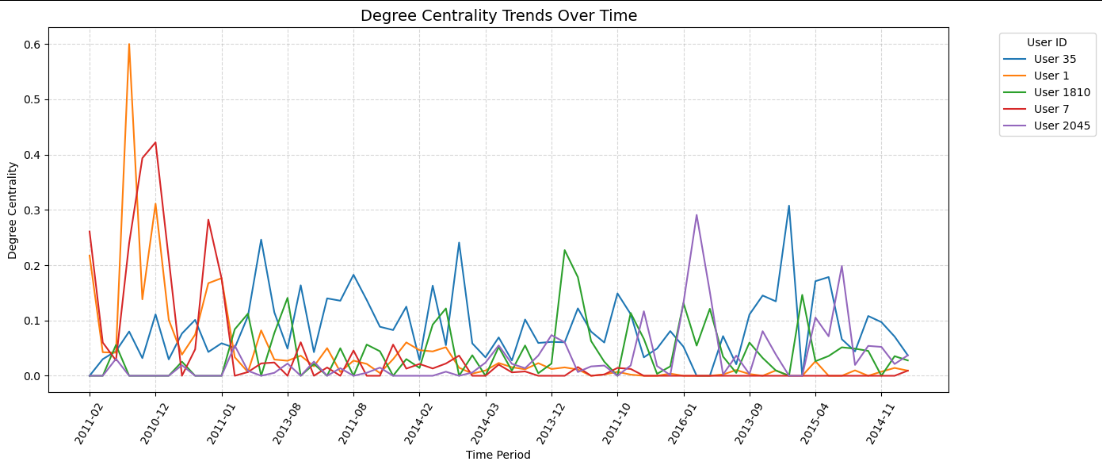


Figure 4: Fluctuations in User Influence: Degree Centrality Trends in the Bitcoin OTC Network Over Time

The graph illustrates degree centrality trends over time for top 5 selected users in the Bitcoin OTC network. Degree centrality measures how connected a user is within the network, reflecting their role in transaction reconciliation.

Some users (e.g., User 1 and User 7) exhibited early spikes in centrality, suggesting that they were highly active in facilitating transactions during specific periods but later declined in influence.

Also, other users (e.g., User 35 and User 2045) maintained sporadic or moderate centrality, indicating sustained but fluctuating participation in the network.

The overall declining trend in centrality suggests that no single user retained dominance over time, possibly due to shifts in trust or the entry of new participants in the network.

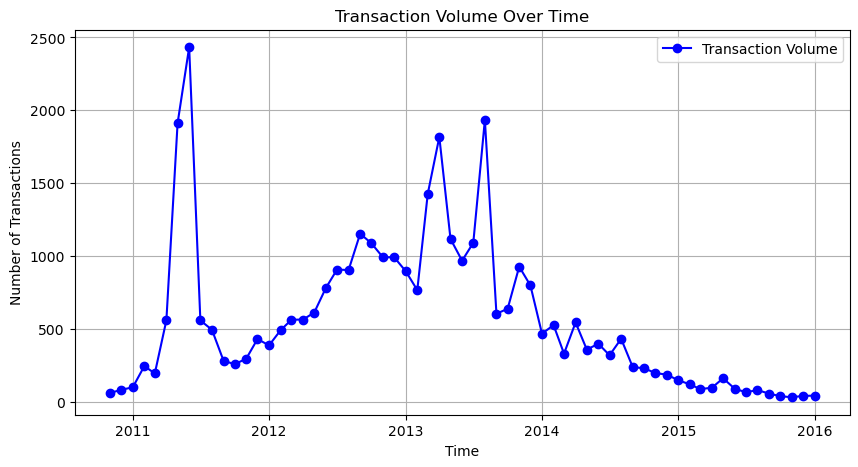


Figure 5:Fluctuations in Bitcoin OTC Transaction Volume: A Surge from 2011-2013 Followed by a Gradual Decline

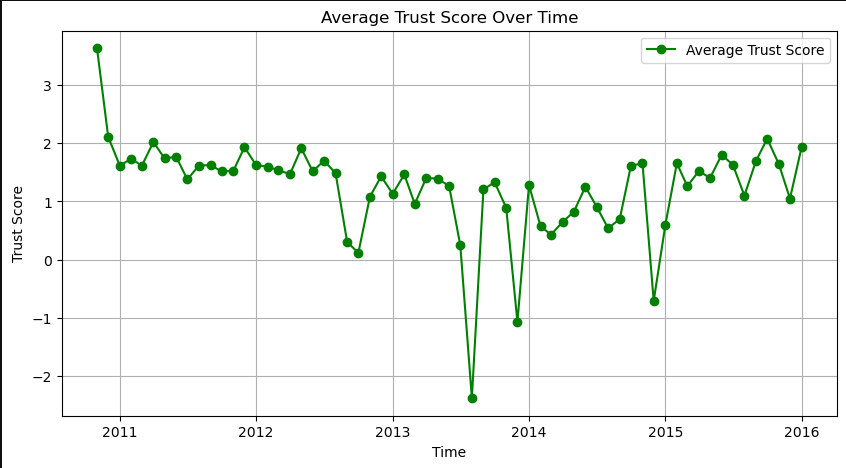


Figure 6: Trust Instability in Bitcoin OTC Transactions: Initial High Trust Declines Amid Market Volatility

The high transaction volume periods 2011-2012 might have been associated with both high engagement and an increase in unreliable interactions.

Additionally, trustworthiness in transactions seems to deteriorate as volume peaks, potentially due to fraudulent users or increased risk-taking behaviour in a growing network.

The eventual decline in transactions may be due to users losing confidence in the system, as indicated by fluctuating trust scores.

# Conclusion and Recommendations

The Bitcoin OTC network analysis shows a complicated relationship between suspicion and trust that is influenced by changing network topologies, transaction volumes, and user behaviour. A thorough analysis of the distribution of trust ratings reveals a significant concentration of neutral (zero) trust values, indicating that participants frequently proceed cautiously when making purchases. Extremely low ratings, on the other hand, point to unresolved conflicts or possible fraudulent activity, highlighting the significance of network reputation management. This dynamic is further highlighted by the structural distinctions between transactions with high and low levels of confidence.   
  
In order to mediate confidence and facilitate transactions within the network, central users are essential. But over time, no single user consistently holds a dominant position, suggesting a changing dynamic of power and trust. Additional information about the network's evolution can be gleaned from temporal variations in transaction volumes and trust scores. Overall activity decreased and trust scores stabilized as the network grew in the following years, indicating a shift toward fewer but more dependable transactions.

Also, users who regularly receive erratic trust ratings are also identified by the analysis; these users may be disruptive actors or be responding to unstable market conditions. These people add to the network's difficulties with reconciliation. Furthermore, because of their peripheral position and poor track record, low-degree nodes—users with fewer connections—frequently obtain lower trust scores. This result emphasizes how vulnerable users with fewer connections are to upholding trustworthy reputations.   
  
Finally, these results emphasize how critical it is to comprehend trust networks' social and structural components. Low-trust clusters expose enduring weaknesses that may jeopardize network integrity, whereas high-trust communities promote consistency and dependability in interactions. Stakeholders can foresee hazards and put policies in place to improve systemic resilience in these networks by keeping an eye on key users and analysing changes in trust over time.

Recommendations

Extremely unfavourable trust scores require investigation of both transactions and people since such negative ratings often suggest potential fraud or unmatched disputes. Established dispute resolution systems along with escrow services serve as proven methods to decrease the dangers which accompany these transactions.

Finally, low-trust clusters expose enduring weaknesses that may jeopardize network integrity, whereas high-trust communities promote consistency and dependability in interactions. Stakeholders can foresee hazards and put policies in place to improve systemic resilience in these networks by keeping an eye on key users and analysing changes in trust over time.

Appendix 1.

*# Step 1: Handle Missing Values*

*data.dropna(inplace=True) # Drop rows with any missing values*

*# Step 2: Remove Duplicate Edges (Keep highest trust rating per transaction)*

*data.sort\_values(by=["source", "target", "rating"], ascending=[True, True, False], inplace=True)*

*data.drop\_duplicates(subset=["source", "target"], keep="first", inplace=True)*

*# Step 3: Convert UNIX Timestamp to Datetime*

*data["timestamp"] = pd.to\_datetime(data["timestamp"], unit='s')*

*# Step 4: Check and Remove Self-loops (Users rating themselves)*

*data = data[data["source"] != data["target"]]*

*display(data.sample(5))*

*# Create Directed Graph Using*

*G = nx.from\_pandas\_edgelist(data, source="source", target="target", edge\_attr=["rating", "timestamp"], create\_using=nx.DiGraph())*

*# Step 7: Print Summary*

*print(f"Number of nodes: {G.number\_of\_nodes()}")*

*print(f"Number of edges: {G.number\_of\_edges()}")*

*print(f"Date Range: {data['timestamp'].min()} to {data['timestamp'].max()}")*

Appendix 2

*import community.community\_louvain as community\_louvain*

*# Detect communities in the high-trust network (now undirected)*

*high\_trust\_partition = community\_louvain.best\_partition(G\_high\_trust\_undirected)*

*# Detect communities in the low-trust network (now undirected)*

*low\_trust\_partition = community\_louvain.best\_partition(G\_low\_trust\_undirected)*

*# Print the number of communities*

*print(f"Number of communities in High-Trust Network: {len(set(high\_trust\_partition.values()))}")*

*print(f"Number of communities in Low-Trust Network: {len(set(low\_trust\_partition.values()))}")*