Do Fraudulent Users Form a Community in a Social Network?

Manim Tirkey
Computer Science
Virginia Tech
Falls Church Virginia USA
mtirkey@vt.edu

Abhimanyu Bhagwati
Computer Science
Virginia Tech
Falls Church Virginia USA
abhimanyu@vt.edu

ABSTRACT

This paper explores the phenomenon of community formation among fraudulent users within the Bitcoin OTC trust weighted signed network, a platform where users trade Bitcoin anonymously. The network comprises 5,881 nodes and 35,592 directed edges, with weights ranging from -10, indicating total distrust, to +10, signifying complete trust. Notably, 89% of these edges carry positive weights, highlighting a predominance of trust among users. Our study leverages the metrics of 'goodness' and 'fairness' from existing research on weighted signed networks (WSNs) to identify fraudulent behaviors effectively. These metrics evaluate how users are perceived (goodness) and how equitably they rate others (fairness), providing a robust framework for detecting anomalous patterns. Our findings reveal that fraudulent users tend to form tightly-knit communities with higher average clustering coefficients and centrality measures compared to the broader network. These communities are not only more interconnected but also wield disproportionate influence within the network, suggesting strategic formation for malicious purposes. Additionally, by analyzing the structural properties of these communities, our study sheds light on the social dynamics and trust relationships that facilitate fraudulent activities in digital trading environments. The insights gained from this analysis are critical for developing targeted strategies for fraud detection and prevention on platforms like Bitcoin OTC, where trust is a fundamental currency. This research contributes to the broader discourse on network theory and cyber-security by demonstrating the utility of WSN metrics in understanding and combating online fraud.

CCS CONCEPTS

•Networks ~ Network properties ~ Network dynamics

KEYWORDS

Bitcoin OTC network, Fraudulent user detection, Community detection, Network centrality measures, Goodness and fairness metrics, Weighted signed networks (WSNs), Trust dynamics in digital trading, Social network analysis, Fraud prevention

1 INTRODUCTION

The rapid evolution of digital currency platforms has transformed economic interactions, fostering new communities and networks wherein trust and reputation are of the utmost priority. The Bitcoin OTC (Over-the-Counter) trading network is a prominent example, where users rate each other on a scale from -10 (total distrust) to +10 (total trust). This study leverages the Bitcoin OTC network's rich data to explore whether fraudulent users form identifiable communities within this trust-based network [5].

Weighted signed networks (WSNs) offer a framework for capturing the complex nature of human relationships, particularly in contexts where interactions can be both supportive and adversarial. Unlike traditional networks that might only capture the presence or absence of a relationship, WSNs accommodate both positive and negative edge weights, making them ideal for analyzing situations where trust and distrust coexist. This dual-weight system enables the representation of a wide range of sentiments in interactions as seen in platforms like Bitcoin OTC. The incorporation of these weights adds a layer of depth to the analysis, allowing us to not only observe but also quantify the strength and polarity of interpersonal dynamics within a network.

The concepts of "goodness" and "fairness" within WSNs provide critical insights into the behavior of network participants, extending beyond simple measures of network centrality or connectivity. "Goodness" refers to the overall trustworthiness or reputation of a user, as perceived by others within the network, and is typically inferred from the positive ratings received. Conversely, "fairness" evaluates how judiciously a user rates others, offering a measure of the integrity with which a user engages in the network's rating system. These metrics are essential for distinguishing between users who are genuinely trusted and those who might exploit the rating system to appear trustworthy (i.e. fraudulent users). By applying these measures, we can uncover subtle patterns of behavior indicative of trust building or erosion, thereby shedding light on the mechanisms through which trust is established in the aforementioned network.

The next part of our analysis extends towards the most common social media network algorithms which help us investigate our main question: Do fraudulent users form communities? We use a community detection algorithm to identify clusters or groups within the network that exhibit distinct patterns of trust interactions. By applying the best partition approach to the network's structure, we separate communities within the Bitcoin OTC network. Subsequently, we perform a comprehensive analysis of these communities to detect and characterize fraudulent activities. This includes calculating the percentage of fraudulent users in each community, community size, clustering coefficients, community density, average community degree, and the overall network's clustering coefficient compared to that of fraudulent nodes. Furthermore, we examine the network centrality measures such as average centrality, betweenness centrality, and closeness centrality, contrasting these metrics with those observed for fraudulent users to draw correlations between network position and fraudulent behavior. Additionally, the study investigates the presence of fraudulent nodes within strongly and weakly connected components of the network.

This research could have an impact on trust management systems for online trading platforms. By studying how users behave on these platforms using statistics and network analysis, the goal is to better predict and prevent fraudulent activity, making peer-to-peer trading safer. This study adds to our understanding of network science and gives practical advice for creating stronger online communities.

2 RELATED WORK

The detection of fraudulent users in social networks and other online platforms has garnered significant attention, largely due to the influence these users can exert on public perception and trust. Previous research has often focused on various methods to identify and mitigate the impact of such users.

One notable approach is presented by Kumar et al. in their study of weighted signed networks (WSNs), where edges between nodes represent trust or distrust with positive or negative weights, respectively [1]. They introduce the concepts of "goodness" and "fairness" of nodes to evaluate the trustworthiness and impartiality of individuals within the network. These metrics are utilized to predict the weight of edges in WSNs, offering a novel method to assess relationships in social and trust networks.

Another significant contribution is the REV2 algorithm developed by Kumar et al., which identifies fraudulent users on rating platforms by evaluating the intrinsic quality of users, ratings, and products through metrics like fairness, reliability, and goodness [2]. REV2's iterative algorithm integrates behavioral data and user interaction patterns to enhance prediction accuracy and has been validated through extensive real-world application, demonstrating superior performance over existing methodologies.

Both approaches underscore the importance of understanding user behavior and network dynamics to effectively detect and counteract fraudulent activities in online environments. Using comprehensive user metrics and algorithms, these studies provide foundational methodologies that inform ongoing research in the field.

3 APPROACH

3.1 Preprocessing the Data

The dataset we are using is the bitcoin OTC WSN. The format of the schema is Source, Target, Rating and Time. We will not use the temporal aspect of the network (although it can be an extension of this study by studying how the network and the fraudulent users in it grow by time). This is a signed weighted network, so, the source and target show the direction of the edge and rating shows the weight of the edge.

For preprocessing the data we kept in mind the recursive algorithm for goodness and fairness of the nodes in the network. So, we scaled the range of weights (or ratings) from [-10, 10] to [-1, +1].

3.2 Fairness and Goodness scores

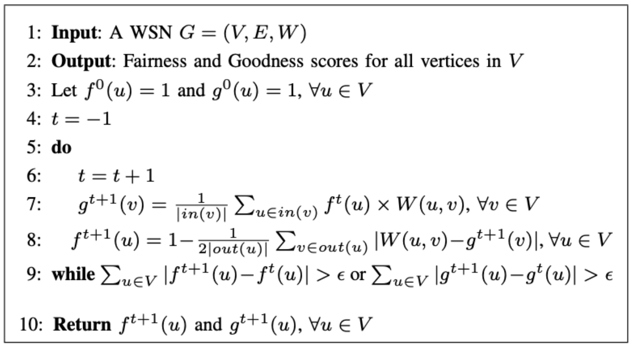
The fairness and goodness is calculated in a mutually recursive manner.

$$g(v) = \frac{1}{|\mathrm{in}(v)|} \sum_{u \in \mathrm{in}(v)} f(u) \times W(u, v) \qquad (1)$$

$$f(u) = 1 - \frac{1}{|\mathrm{out}(u)|} \sum_{v \in \mathrm{out}(u)} \frac{|W(u, v) - g(v)|}{R} \qquad (2)$$

Fairness scores are always within the range of 0 to 1, while goodness scores fall within the range of -1 to 1, which corresponds to the edge weights in our scenario. The maximum possible difference between an edge weight and a goodness score is 2, represented by the range of difference, denoted as R.

In the goodness formula for a vertex v, the incoming edge weights are adjusted according to the fairness of the vertices providing the ratings, prioritizing ratings from fair vertices. The goodness of v is determined by averaging these weighted products across all predecessors. When assessing the fairness of a vertex u, a smaller difference between the actual edge weight and the recipient's goodness indicates greater fairness. Once again, an average of all the ratings given by vertex u is computed to determine its fairness. [1]



```
Input: A WSN $G = (V, E, W)$
Output: Fairness and Goodness scores for all vertices in $V$
Let $f^0(u) = 1$ and $g^0(u) = 1$, $\forall u \in V$
$t = -1$
do
    $t = t + 1$
    $g^{t+1}(v) = \frac{1}{|in(v)|} \sum_{u \in in(v)} f^t(u) \times W(u, v), \forall v \in V$
    $f^{t+1}(u) = 1 - \frac{1}{2|out(u)|} \sum_{v \in out(u)} |W(u, v) - g^{t+1}(v)|, \forall u \in V$
while $\sum_{u \in V} |f^{t+1}(u) - f^t(u)| > \epsilon$ or $\sum_{u \in V} |g^{t+1}(u) - g^t(u)| > \epsilon$
Return $f^{t+1}(u)$ and $g^{t+1}(u)$, $\forall u \in V$
```

Figure 1: The algorithm for Fairness and Goodness

The algorithm (Figure 1) provides a method for calculating Fairness and Goodness scores for all vertices in a weighted signed network (WSN) characterized by vertices V, edges E, and weights W. Starting with initial values of 1 for both scores for all vertices, the algorithm iteratively adjusts these scores based on interactions within the network. For each vertex v, the Goodness score at iteration $t + 1$, $g^{t+1}(v)$, is updated by averaging the product of the neighboring vertices' Fairness scores and the weights of the connecting edges. The Fairness score, $f^{t+1}(u)$ is updated based on the outgoing edges, considering the difference between the edge weight and the product of the neighbor's Goodness score and the vertex's new Goodness score. This iterative process continues until the change in scores between two consecutive iterations is less than a small threshold $\epsilon = 10^{-6}$, ensuring convergence to stable Fairness and Goodness values across the network. Finally, the algorithm returns these converged scores for each vertex.

3.3 Detecting Fraudulent Users

The choice of thresholds for categorizing users as fraudulent in a network using "fairness" and "goodness" metrics relies on a balanced approach between median and mean values of these metrics. The formulas given:

$$\text{fairness}_{threshold} = \text{median}_{fairness} - \frac{\text{median}_{fairness} - \text{mean}_{fairness}}{2} \tag{3}$$

$$\text{goodness}_{threshold} = \text{median}_{goodness} - \frac{\text{median}_{goodness} - \text{mean}_{goodness}}{2} \tag{4}$$

is being used to leverage both the robustness of the median—which is less sensitive to extreme outliers—and the mean. This threshold is particularly useful in environments where data may be skewed or contain extreme values that could distort a pure mean or median-based approach. By adjusting the median closer towards the mean, the threshold aims to capture a more 'central' tendency which represents the typical behavior within the network. This method helps to differentiate between ordinary users and those who consistently exhibit behavior (either in fairness or goodness) that deviates significantly from the norm, thereby potentially identifying fraudulent activities.

One question that we haven't addressed yet is: **why not simply use the average rating of each user given by their predecessor and not rely on 'goodness' and 'fairness for detecting a fraudulent node'?** The answer is simple. The fraudulent users tend to give fraudulent ratings to others and we have no particular metric to verify that the current average rating associated with a user is the actual rating of that user. As an example, in real-world networks like Bitcoin, scammers often create multiple fake accounts to manipulate the trust system. They use these accounts to artificially boost their own ratings while undermining the ratings of honest users. 'fairness' and 'goodness' ratings leverage on the recursive approach to converge to a particular value using both the edge weight and direction. These values are **guaranteed** to converge to one fairness and goodness value given enough iterations or to a termination value (which is usually less than 10e-6) [1].

3.4 Community Detection

The choice of community detection algorithm was the Louvain method provided by the 'community' API in python. We have also used a widely used network analysis library Networkx for creating a graph.

Since, the 'community' API requires an undirected graph for detecting communities (implying two-way connection between two nodes) we made a copy of our original network and converted it to an undirected graph.

Our initial approach was to use the partitions made by the Louvain algorithm to find any anomaly in our original graph but even if we marked our communities in the graph the graph was too cluttered (Figure 2) to have an actual visual anomaly associated with it.

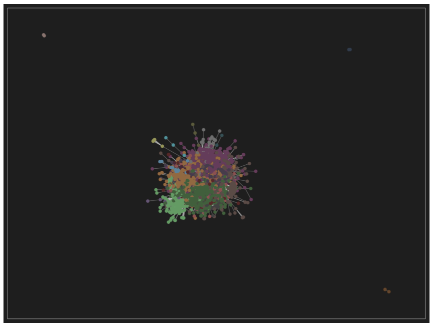


Figure 2: Bitcoin OTC WSN partitioned by the Louvain Algorithm (each partition has been associated with a new color)

So, we then took the statistical route to actually produce a meaningful result. For each community we compared their sizes, edge count, density, diameter, average degree, clustering coefficient, average path length and most importantly the percentage of fraudulent users in that community.

3.5 Remaining Network Analysis and Comparison

For the remaining comparison we used the Networkx library to calculate the Average Centrality, Average Betweenness Centrality, Average Closeness Centrality of the overall graph and the same averages specifically for the fraudulent users.

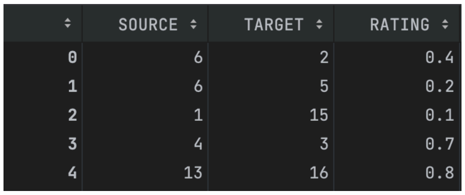
$$\text{Average Centrality of Fraudulent Users} = \frac{1}{N}\sum_{u \in \text{Fraudulent Users}} \text{degree\_centrality}_u \qquad (5)$$

Equation (5) represents how we are specifically calculating the average centrality of each fraudulent user only. This method has been applied for all the remaining metrics as well to make a comparison.

The last metric that we have calculated is the strongly and weakly connected components in the graph. Again since we already have a list of fraudulent users we compare each partition and check the percentage of fraudulent users in each connected component (both strong and weak).

4 EXPERIMENT AND OBSERVATIONS

The approach section has described the scaling of ratings for each edge from [-10, 10] to [-1, 1]. This was required to be done because of the nature of our recursive algorithm for 'fairness' and 'goodness' which converges and always lies between [0, 1] and [-1, 1] respectively.



⇕	SOURCE ⇕	TARGET ⇕	RATING ⇕
0	6	2	0.4
1	6	5	0.2
2	1	15	0.1
3	4	3	0.7
4	13	16	0.8

Figure 3: Bitcoin OTC WSN dataset after scaling the ratings of each edge

For computing the fairness and goodness of each node we employ the equation (1) and (2) (mostly our algorithm was a direct implementation of Figure 1) for 100 iterations (generally it converges within these iterations). Then store these values in a dictionary.

We notice that the mean fairness and goodness is 0.93 and 0.070 for all the users which shows that a vast majority of our users are actually honest users. The mode and median values of the data also show similar values suggesting the same.

We had also thought about using a different metric to find the fraudulent users but we felt that it was too strict and it was capturing very few potentially fraudulent users. Our approach was to look at the histogram of Fairness and Goodness values and we came to the conclusion that we just need to take the top 5 and 10 percentile as the threshold respectively.

The next step is filtering out the fraudulent users. For that we simply apply equations (3) and (4) on each of the users and finally create a list of these users. We catch a total of 434 fraudulent users which is a total of about 7.37% of the total users. The final threshold was 0.95 fairness and 0.084 goodness.

4.1 Fraudulent Users in detected Community

The 'community' API [3] helped us to determine that there are a total of 22 communities in our social network. We then used the list of fraudulent users to determine what percent of these users are in each community.

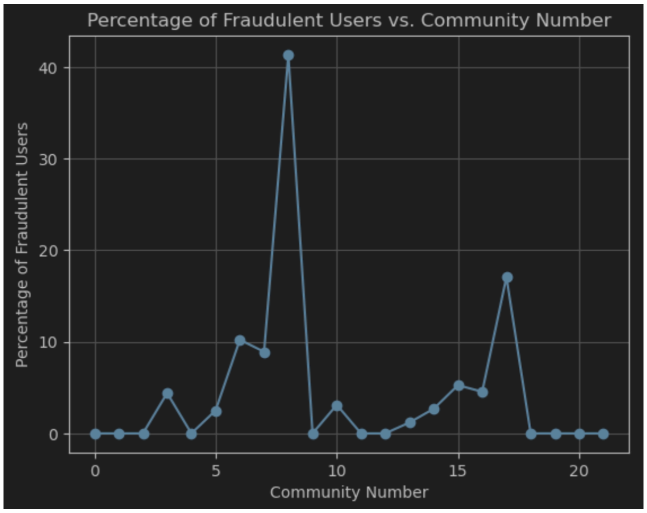


Figure 4: Percentage of Fraudulent Users in each Community

It is quite evident from the graph (Figure 4) that a lot of the communities don’t have any fraudulent users at all. Some observations that we made through this graph:

1. A majority (>50%) of the communities did not contain any fraudulent users.
2. Most communities had less than 20% of its users marked as fraudulent.
3. Community 8 became an outlier and contained around 41.33% fraudulent users.

We then graphed the percentage of fraudulent users with respect to the community sizes to check whether these users exist in particularly larger or smaller graphs.

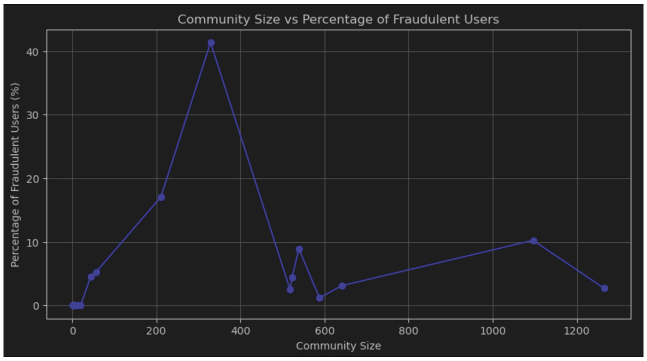


Figure 5: Community Size versus the Percentage of Fraudulent Users in each Community

This graph (Figure 5) suggested to us that most fraudulent users exist in smaller to medium sized communities. Some observations that we made on this graph:

1. Larger communities do not necessarily have a higher percentage of fraudulent users.
2. A medium-sized community (community 8) of around 400 members shows a peak in the percentage of fraudulent users.
3. Most fraudulent users exist in small to medium sized communities.

Our next graph (Figure 6) observed the percentage of fraudulent users in each community versus the clustering coefficient of each community.

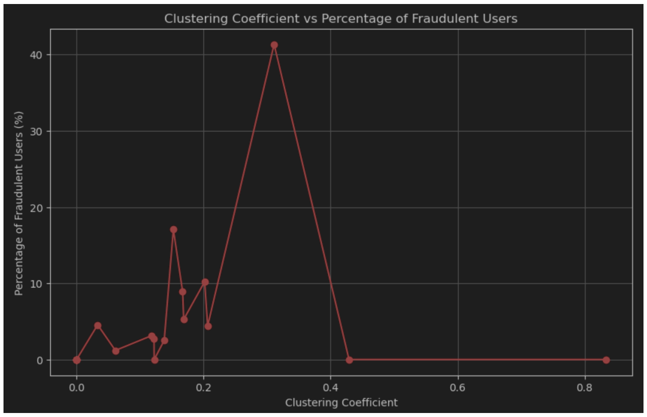


Figure 6: Clustering Coefficient versus the Percentage of Fraudulent Users in each Community

A basic observation we made was that none of the communities which held fraudulent users had a higher clustering coefficient (> 0.4). Some other observations made on the graph:

1. Most communities with fraudulent users have a clustering coefficient between 0.01 to 0.35
2. The peak is around 0.35 which suggests that tighter-knit communities are prone to fraudulent users.
3. Communities with higher clustering coefficients do not contain any fraudulent users this suggests that the more interconnections are made the better people get at identifying fraudulent users.

Nextly, we observed the densities of all the communities and compared it with the percentage of fraudulent users in them.

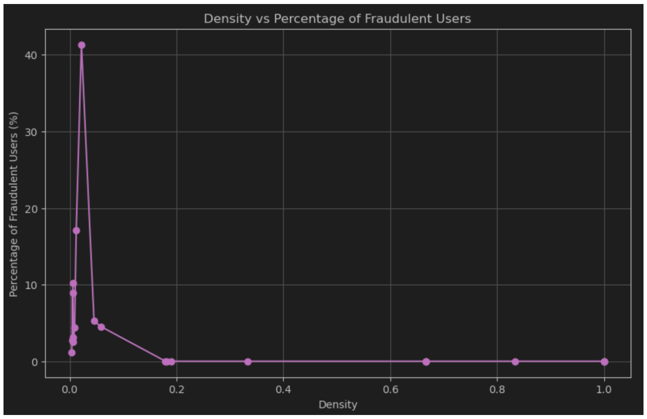


Figure 7: Community Size versus the Percentage of Fraudulent Users in each Community

We can easily observe a sharp graph (Figure 7). Most communities with any fraudulent users lie in graphs with very less density. This shows that fraudulent users have higher penetration in less dense communities.

The next graph we plotted was of the average degree of each community and the percentage of fraudulent users in these communities.

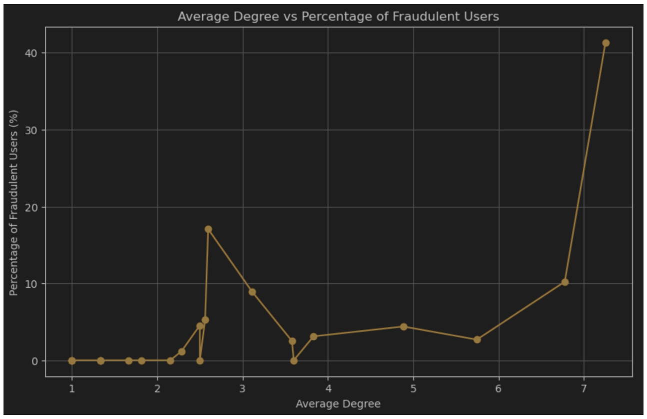


Figure 8: Average Community Degree versus the Percentage of Fraudulent Users in each Community

We see that with gradual increase in the communities average degree we can find a subtle increase in the percentage of fraudulent users. It blows up after an average degree of 7. One community which has an average degree of 3.5 has a percentage of fraudulent users of about 18%. After that we are able to see that there is a subtle increase in fraudulent users after a quick drop.

Community 8 containing 41.33% fraudulent users is a good indicator that fraudulent users **do form** communities. Certain community size ranges can facilitate fraudulent activities. The peak at a clustering coefficient of around 0.35 suggests that communities where members are more interconnected (higher clustering) also have a higher likelihood of fraudulent activities.The sharp peak at very low densities (close to 0.1) with high fraudulent percentages suggests that sparser communities can have concentrated fraud activities. This might be due to easier coordination in less dense networks. The significant increase in the percentage of fraudulent users at an average degree of 7 indicates that communities where each member connects with many others tend to have higher fraud rates. This might reflect that more connected individuals can influence or engage in fraudulent activities more effectively.

4.2 Clustering Coefficient Analysis

We calculated the average clustering of the whole network and compared it to the clustering coefficient of a subgraph which contained only the fraudulent users.

The average clustering coefficient of all the users is 0.178 and the average clustering coefficient of the fraudulent users is 0.378.

This indicates that fraudulent users are more tightly knit than the average user. High clustering among fraudulent users suggests that they tend to form close-knit groups, which could facilitate coordinated fraudulent activities.

4.3 Centrality Measures

We calculated various centrality measures of the whole and the centrality measure of the fraudulent users by using equation (5).

The following are our observations:

1. Average Centrality of All Nodes: 0.00124
2. Average Centrality of Fraudulent Users: 0.00246
3. Average Betweenness Centrality of All Nodes: 0.00044
4. Average Betweenness Centrality of Fraudulent Users: 0.00067
5. Average Closeness Centrality of All Nodes: 0.284
6. Average Closeness Centrality of Fraudulent Users: 0.304

These metrics suggested to us that fraudulent users have higher centrality in the network compared to the average. They are more connected and can reach other nodes more quickly, which may enhance their ability to spread influence or information within the network.

4.4 Strongly and Weakly Connected Components

The Networkx library [4] helped us to find the various strongly and weakly connected components. A vast majority of our strongly connected components had total nodes as 1. And almost all of these singleton strongly connected components are of those of fraudulent users.

Percentage of Fraudulent Users in Components:

1. Strongly Connected Components: Very high concentration (mostly 100%) (Figure 9)
2. Weakly Connected Components: 7.39%

This stark contrast between strongly and weakly connected components could indicate that while fraud clusters are highly interconnected, they only form loose connections with the broader network, maintaining their activities confined within their groups.

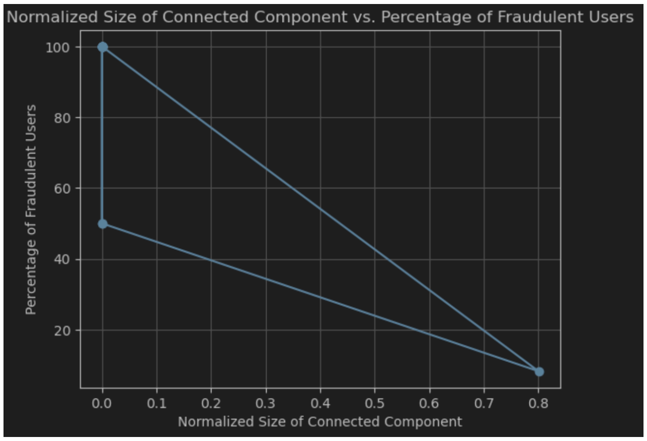


Figure 9: Percentage of Fraudulent Users vs Size Normalized Size of Connected Component

The graph (Figure 9) illustrates a notable **inverse** relationship between the normalized size of connected components in a network and the percentage of fraudulent users within those components. Specifically, as the size of the connected components increases (from 0.0 to 0.8), the percentage of fraudulent users within these components decreases significantly—from 100% in the smallest components to around 10% in the largest components. This trend could suggest that smaller connected components tend to have higher concentrations of fraudulent users, potentially indicating isolated groups of malicious actors within the network. Conversely, larger components, which are likely more diverse and involve more users, have a lower proportion of fraudulent users, possibly due to increased oversight and the presence of more legitimate interactions diluting the impact of fraudulent activities.

5 **CONCLUSION**

This research provides insights into the formation of communities among fraudulent users within the Bitcoin OTC trust weighted signed network. Our comprehensive study demonstrates that fraudulent users not only form distinct communities but also exhibit significantly different network properties compared to the general user base. We observed that these communities are smaller, more tightly knit, and feature higher measures of centrality, indicating that fraudulent users are strategically positioned within the network to exert influence and propagate their activities efficiently.

The detection of such patterns is crucial for developing more effective fraud detection systems. By understanding the characteristics and behaviors of these communities, platforms can tailor their monitoring strategies to be more proactive and targeted, potentially intercepting fraudulent activities before they inflict widespread damage. This research also underscores the importance of applying network science techniques to enhance the trustworthiness of digital trading platforms, where reputation and reliability are of the utmost priority.

Future research could extend this work by exploring temporal dynamics to understand how fraudulent communities evolve over time. Such studies could reveal whether certain network changes precede the emergence of fraud, offering predictive insights that could be used to preemptively tighten network security measures. Additionally, incorporating machine learning models to automatically detect and predict fraudulent behavior based on network changes could further enhance the robustness of fraud detection systems. Finally, expanding the analysis to include multiple trading platforms could validate the generalizability of the findings and help in the development of universal strategies for fraud prevention across various digital environments.

REFERENCES

[1] Kumar, S. *et al.* (2016) 'Edge weight prediction in weighted signed networks', *2016 IEEE 16th International Conference on Data Mining (ICDM)* [Preprint]. doi:10.1109/icdm.2016.0033.

[2] Kumar, S. *et al.* (2018) 'REV2', *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* [Preprint]. doi:10.1145/3159652.3159729.

[3] *Community detection for NetworkX's documentation¶ Community detection for NetworkX's documentation - Community detection for NetworkX 2 documentation*. Available at: https://python-louvain.readthedocs.io/en/latest/index.html

[4] *Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008*

[5] *Bitcoin OTC Trust weighted signed Network SNAP*. Available at: https://snap.stanford.edu/data/soc-sign-bitcoin-otc.html