Network Role Discovery and Node Importance in the Bitcoin OTC Trust Network

Austin Maxwell, Melissa Castellanos
austintmaxwell@lewisu.edu, melissacastellanos@lewisu.edu
DATA-61000-001, Summer 2025
Advanced Data Mining and Prescriptive Analytics
Lewis University

I. Introduction

Bitcoin over-the-counter (OTC) trading is the process of buying and selling large quantities of Bitcoin directly between parties without public visibility [1][2]. It's a favorable trading method among high-net worth individuals and institutional investors who are seeking discretion or avoiding market place disruptions [1]. Unlike typical exchange based trading, Bitcoin OTC trading offers increased liquidity, flexibility, and secrecy [2]. Due to the inherent ambiguity of Bitcoin OTC trading, it's not without risks. One of the biggest challenges in OTC traders encounter is the counterparty risk, where one party fails to deliver after the other has already complied with the agreed-upon obligations. To combat this potential risk, it's critical to identify reliable, trustworthy traders as well as unreliable, untrustworthy traders. Network embedding enables the detection of trusted users and bad actors by identifying meaningful node representations within a given network [3]. Through network role discovery and importance evaluation, it becomes possible to discern irregularities which might indicate untrustworthy traders [3].

This project aims to analyze the Bitcoin OTC trust network to distinguish key players and potentially risky users through the discovery of node roles and their relative importance. Low-dimensional feature vectors are generated using two different network embedding techniques, Node2vec and Struc2vec. Node2vec is mainly utilized for node classification, while Struc2vec aims to determine structural equivalence [4]. By utilizing these two network embedding methods, we get a better understanding of the network's structure and behavior. These embeddings are then clustered to reveal structural roles within the network, which are evaluated using Eigenvector Centrality and PageRank.

The following report outlines the development of the low-dimensional feature vectors, the role discovery process, and the evaluation of each identified role. Section II describes the data utilized for the network embedding. Section III addresses both embedding techniques, the clustering process, and the calculation of centrality and node importance measures. Section IV and V discuss results and conclusions.

II. Data Description

The risks associated with Bitcoin OTC trading are mitigated by the maintenance of a user reputation record [5][6]. The Bitcoin OTC trust weighted signed network dataset describes this record. The dataset contains data on how members of Bitcoin OTC rate each other's trustworthiness on a scale of -10 to 10, with 10 being total trust. Attributes of the dataset are described in Table 1.

TABLE I. Bitcoin OTC Trust Weighted Signed Network

Attribute	Type	Example Value	Description
SOURCE	Numeric (integer)	6	Node id of source, i.e., rater
TARGET	Numeric (integer)	2	Node id of target, i.e. ratee
RATING	Numeric (integer)	4	The source's rating for the target, ranging from -10 to 10 in steps of 1
TIME	Numeric (float)	1289241911.72836	Time of rating, measured as seconds since Epoch

There are a total of 5,881 nodes, or ratings, in the dataset with 35,592 edges. Of the 35,592 edges in the network 89% are positive. A visualization of the network is available in Figure 1.

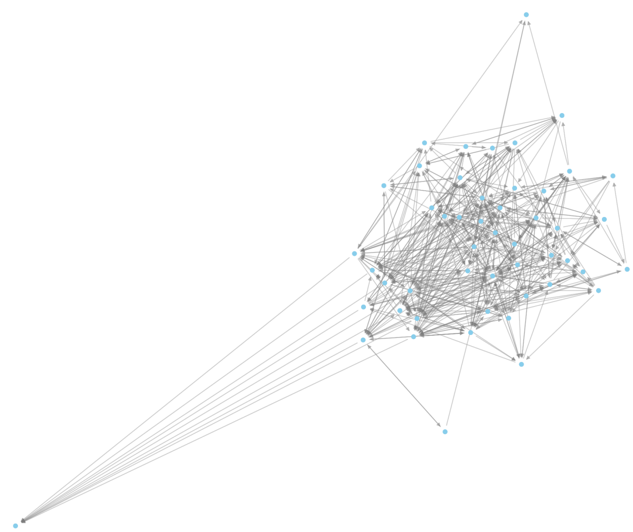

Fig. 1. Bitcoin OTC trust weighted signed network. Top 50 nodes by centrality represented by blue points, where directed edges are shown as grey arrows.

III. Methodology

Roles within a network can be identified through the process of role discovery, which detects nodes with similar functions, connections, or interactions [8][9]. Role discovery leverages network embedding methods to identify the particular role each node plays [9]. This project utilizes Struc2vec and Node2vec, network embedding, to convert each node to a lower-dimensional feature vector. These low-dimensional vectors were clustered using DBSCAN to discover functional roles in the network and potential abnormalities. Using Degree Centrality, Eigenvector Centrality, and Closeness Centrality, each node was then ranked on importance.

A. *Struc2vec*

The Struc2vec network embedding method is a framework utilized to derive vector representations that reflect each node's structural position or role [10]. It's implemented in this project using the Pandas, Numpy, NetworkX, and Scikit-Learn libraries. The first step in the application of Struc2vec was the creation of a directed graph utilizing the "Source", "Target" and "Rating" attributes of the Bitcoin OTC Trust dataset. To capture structural characteristics, the degree signature of each node in the network was computed. This signature summarizes the sorted degrees of a node's neighbors [11]. A cosine similarity matrix was then computed from the degree signatures, indicating which nodes were structurally similar to each other [11]. This was converted into a distance matrix, which was reduced into lower-dimensional embeddings using PCA.

B. *Node2Vec*

For our Node2Vec portion of this study, we began by assembling a suite of open-source Python libraries to support graph construction, random-walk generation, and vector representation learning. The NetworkX package provided efficient utilities for building and manipulating the network structure along with its attributes. Pandas and NumPy were leveraged to ingest tabular data and perform numerical operations. We also imported Python's built-in random module to ensure reproducibility in sampling procedures. Finally, we leveraged the Node2Vec implementation of Grover and Leskovec to generate node embeddings via biased random walks and skip-gram training [7].

The raw dataset comprised an edge list in CSV format, where each record specified a source node, a target node, a rating score, and a timestamp. We loaded this file into a pandas DataFrame, explicitly naming the columns to guard against misalignment. Only the source and target columns were retained for embedding, since the Node2Vec algorithm treats the graph as unweighted by default.

With the edge list prepared, we instantiated an undirected NetworkX Graph by converting the DataFrame directly into graph form. This step automatically registered each unique identifier as a node and each pairwise interaction as an edge, yielding a complete representation of the transactional network.

Next, we initialized the Node2Vec model with hyperparameters: a 64-dimensional embedding space, walk length of 30 steps, and 100 walks initiated per node. The return parameter p and in-out parameter q were both set to 1 to balance exploration and exploitation in the random walks, and four worker processes were allocated to parallelize walk generation.

Model fitting brought in the underlying Word2Vec training routine, configured with a context window of size 10, a minimum node count threshold of 1 (to include all nodes in the vocabulary), and a modest batch size. Upon convergence, the resulting embedding vectors were available through the model's `wv` attribute. We extracted these vectors along with their corresponding node identifiers into a pandas DataFrame, labelling each column with `emb_0` through `emb_63`. This final table of node-to-vector mappings was then exported for subsequent clustering and role-detection analyses which we will discuss next.

C. Clustering & Role Discovery

In the clustering phase, we began by loading the node embedding tables generated in the aforementioned steps, i.e., Struc2Vec & Node2Vec, into memory as pandas DataFrames. For each embedding set, we isolated the vector columns (i.e., all but the identifier column) and converted them into NumPy arrays for compatibility with scikit-learn clustering routines. Before clustering, no further dimensionality reduction was applied, as both embedding methods had already distilled the graph structure into a compact feature space.

For density-based clustering, we first applied DBSCAN, an algorithm that groups together points closely packed in space while marking low-density points as outlier. Key hyperparameters included ε (the maximum neighborhood radius) and min_samples (the minimum number of points required to form a dense region). We tuned these parameters empirically by inspecting the proportion of points labeled as noise and ensuring that the number of core clusters was neither trivially small nor excessively fragmented. Recognizing DBSCAN's sensitivity to global density variations, we also brought in HDBSCAN, a hierarchical extension that identifies clusters across multiple density thresholds without requiring a single ε parameter. HDBSCAN's primary parameter, min_cluster_size was set to balance the granularity of detected clusters against the desire to avoid overfitting to minor irregularities.

As a complementary approach, we fit Gaussian Mixture Models (GMMs) to each embedding space, treating cluster assignments as latent probability distributions in the continuous feature space. We evaluated models with varying numbers of mixture components using the Bayesian Information Criterion (BIC) to select the optimal cluster count, and then assigned each node to the component for which it had highest posterior probability. Across DBSCAN, HDBSCAN, and GMM, we compared clustering validity using silhouette scores and the Davies-Bouldin Index, selecting the most stable partitioning for subsequent role analysis.

Once cluster labels were assigned, we merged them back with network-level metrics computed earlier, e.g., in-degree, out-degree, total transaction volume, and average transaction volume for each node. By aggregating these metrics at the cluster level (calculating means, medians, and totals), we obtained a concise behavioral profile for each group. These profiles served as the basis for role extraction. For example, clusters exhibiting high average out-degree and volume were flagged as potential distributors, whereas those with high in-degree but low out-degree suggest we are seeing absorbers. Clusters with balanced yet modest transaction metrics would be grouped as intermediaries.

By comparing cluster compositions across Struc2Vec and Node2Vec, we were able to distinguish roles arising from directional transactional patterns versus those emerging from structural similarity in the broader network.

D. Node Evaluation

To quantify the importance of nodes in the Bitcoin OTC Trust network, we calculated three centrality measures, Degree Centrality, Eigenvector Centrality and Closeness Centrality. Degree Centrality measured the number of connections each node had, Eigenvector Centrality quantified the influence of each node in the network, and Closeness Centrality calculated the 'closeness' of each node to all other nodes. All three measures of centrality were normalized and averaged to compute node importance using Multi-Centrality Averaging. Node importance was defined as the sum of all normalized centrality measures, divided by the number of centrality measures calculated (1).

$$\text{Importance} = (\textit{Degree Centrality} + \textit{Eigenvector Centrality} + \textit{Eigenvector Centrality}) / 3 \qquad (1)$$

All 5,881 nodes in the network were then ranked on importance to determine the most influential, and structurally important nodes in the Bitcoin OTC Trust network. Nodes with larger calculated importance values were considered to be key players in the network. While these rankings were done globally, determining the importance of nodes within each cluster could give more insight into the role each node plays in their given cluster. However, to calculate cluster level importance we would have to track nodes and their clustering position through the clustering process.

IV. Results and Discussion

A. Node2Vec

In our analysis of the 64-dimensional Node2Vec embeddings, we applied the DBSCAN algorithm to uncover densely connected communities within the transactional network. As shown in Figure 2, the empirically parameterized DBSCAN identified 12 clusters along with a noise (Cluster -1). Our two largest clusters encompass a vast majority of the total nodes, indicating two large cohorts for tightly interaction participants; the remaining clusters were smaller and more specialized. The noise points, rendered navy in the plot, correspond to nodes whose local density fell below the threshold, potentially reflecting one-off or anomalous interactions.

To visually confirm these groupings, we projected the embeddings into two dimensions using UMAP and colored each point by its DBSCAN label. In Figure 2, the two dominant clusters appear as distinct blobs, while the smaller clusters occupy peripheral regions of the plot. This separation in UMAP space demonstrates that the Node2Vec representation captured community structure amenable to density-based clustering.

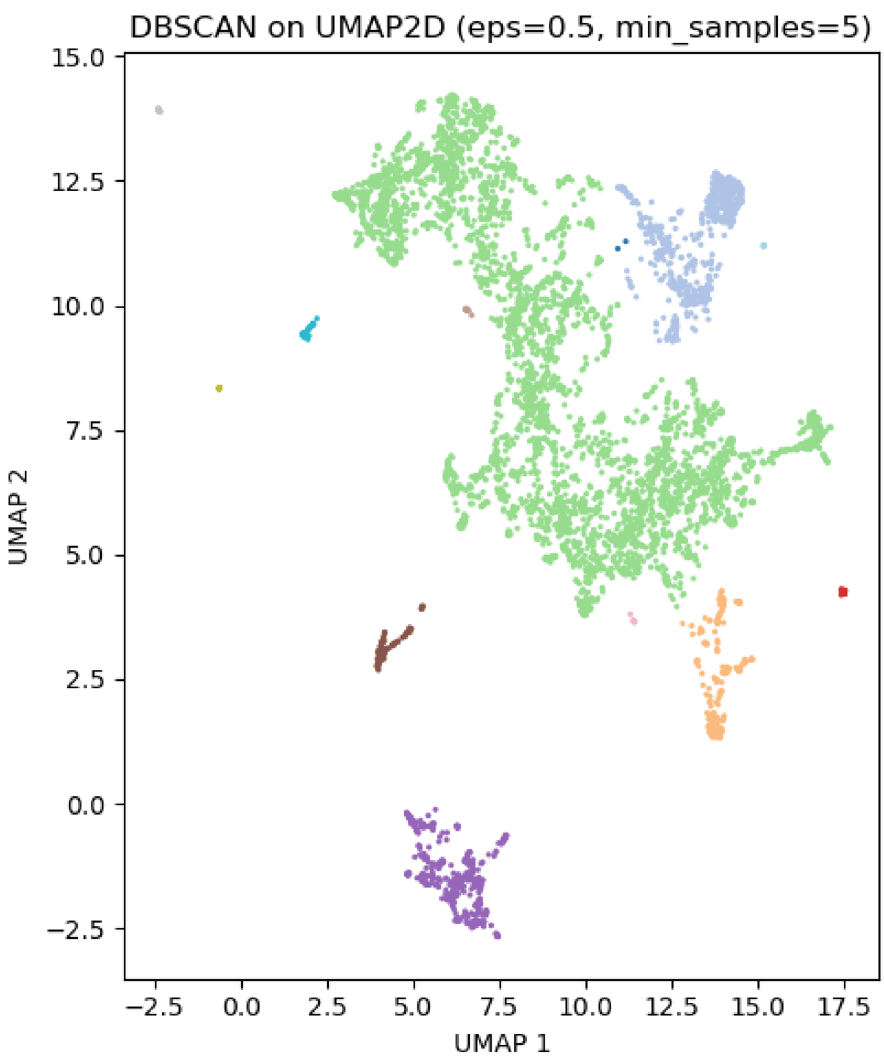

Fig. 2. Caption

Role discovery was performed by merging the DBSCAN cluster assignments with node-level metrics. This revealed a spectrum of user communities, from a large core of everyday traders to several smaller, more specialized cohorts. The dominant cluster captures casual participants who engage in modestly positive trust exchanges, while mid-sized groups reflect well-liked regulars and optimistic raters. A separated distrusted cluster highlights users with net-negative interactions, and at the other end of the spectrum, small micro-clusters represent highly trusted or elite actors. Table II below consolidates each cluster's size and average in- and out-trust metrics, serving as the basis for high-level role interpretation.

TABLE II. NODE2VEC ROLE DISCOVERY

Cluster Color	Size	avg_out	avg_in	Interpretation
Light Green	4,200	1.52	0.89	**Everyday traders.** This is the vast majority of nodes. They give modestly positive trust and receive modestly positive trust. Think casual users who both rate and get rated a bit above zero.
Light Blue	656	1.20	1.22	**Well-liked regulars.** A mid-sized community with slightly higher incoming than outgoing trust—reliable counter-parties who tend to earn a bit more trust than they give.
Purple	456	1.35	1.10	**Optimistic raters.** They give out more trust than they get back, perhaps generous raters giving their first rating or connectors linking groups.
Orange	292	-0.92	-4.17	**Distrusted cluster.** Net negative both in giving and receiving: these are the accounts that get flagged and flag others. Possible scammers or users with poor reputations.
Brown	125	0.80	1.34	**Trust-recipients.** They receive substantially more trust than they give, perhaps new users who haven't had many opportunities yet to rate others.
Red	49	2.02	1.75	**Trusted core.** A small clique of highly active, highly trusted users. They give very high ratings and get very high ratings back.
Teal	37	1.81	1.77	**Elite traders.** Similar to cluster 3 but even tighter—power users with excellent reputations on both sides.
Grey	18	2.52	2.18	**Super-trusted givers.** They almost always assign top marks (avg_out >2) and earn top marks in return. Very small, very elite.
Yellow	15	2.12	1.87	**High-trust hubs.** Another tiny, highly positive cluster, likely "hubs" in the network who both give and get excellent scores.
Green	13	1.64	1.17	**Moderate elites.** A micro-cluster of users with above-average outgoing trust but more modest incoming.
Pink	12	2.36	1.56	**Generous, selective.** They give very high trust but receive a bit less in return.
Cyan	6	0.17	1.00	**Quiet but trusted.** Very few ratings given but what little they do they do right and they're trusted by others.
Navy (noise)	2	1.23	1.55	**Outliers.** Only two nodes didn't fit any dense region; they have small volumes of mostly positive interactions.

B. *Struc2vec*

Applying DBSCAN to the 10-dimensional Struc2Vec embeddings yielded a set of structurally equivalent communities alongside a small noise component. DBSCAN, once more configured empirically, identified 23 clusters without any points labeled as noise. This indicates that every node found a structurally similar community under the chosen parameters. The three largest clusters encompass over one half of all nodes, reflecting broad structural roles: everyday participants (balanced, modest trust exchanges). Mid-sized clusters capture connectors (linking disparate regions) and well-liked regulars (slightly more incoming than outgoing trust), while a long tail of smaller clusters highlights niche structural positions. These likely point to strong negatives to elite givers, each defined by distinct avg_out and avg_in profiles.

To visually validate these communities, we projected the Struc2Vec embeddings into two dimensions via uMAP and overload DBSCAN labels in Figure 3. The results show well-separated blobs for the largest clusters, clear offshoots for mid-sized groups, and fine-grained separation among smaller clusters. This separation confirms that Struc2Vec effectively captured structural equivalence: nodes occupying similar network positions cluster together even without direct interactions.

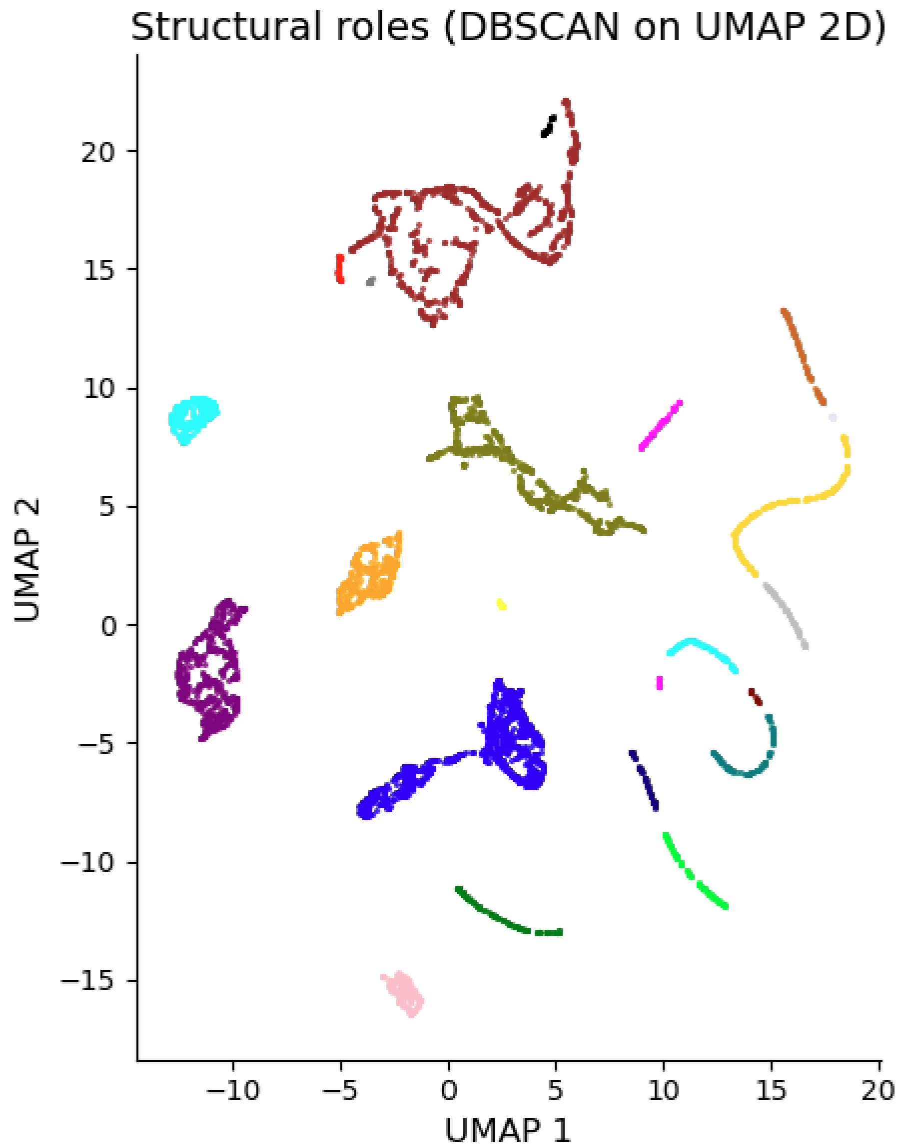

Fig. 3. Caption.

For role discovery, we once again merged each node's cluster assignment with node-level metrics. This process revealed a larger spectrum of structural roles, from a large core of everyday participants to several smaller, more specialized cohorts. The dominant cluster captures casual users with balanced modest trust exchanges, while other mid-sized groups correspond to generous givers and active raters. Similar to with Node2Vec, we also see a distinct negative-trust cluster highlighting nodes with predominantly unreciprocated or adverse interactions, and at the opposite extreme, several micro-clusters represent elite or highly trusted actors. Table III below consolidates all the cluster- and node-level information extrapolated from our visual.

TABLE III. STRUC2VEC ROLE DISCOVERY

Cluster Color	Size	avg_out	avg_in	Interpretation
Blue	1,163	1.16	0.99	**Everyday participants.** The largest group engages in modestly positive trust both ways.
Brown	1,069	2.08	1.07	**Optimistic givers.** High outgoing trust with moderate incoming, likely generous connectors.
Olive	796	1.92	1.06	**Frequent raters.** Regularly active in rating others, receiving similar modest trust.
Purple	705	1.72	0.98	**Connectors.** Above‑average outgoing trust, linking different parts of the network.
Orange	450	1.39	1.09	**Well‑liked regulars.** Steady, balanced trust exchanges, slightly more incoming than outgoing.
Cyan	291	1.32	0.76	**Cautious givers.** Moderate outgoing trust but lower incoming, perhaps new or selective.
Pink	227	1.45	1.10	**Balanced participants.** Evenly matched in giving and receiving, though less active.
Gold	200	0.03	-1.88	**Distrusted sinks.** Almost no outgoing trust and strongly negative incoming.
Teal	133	0.00	0.48	**Silent receivers.** Do not rate others but receive some modest trust.
Green	131	0.00	1.48	**Collectors.** Never give trust but are trusted by others.
Lime	105	0.10	-0.54	**Occasional doubters.** Very little giving, slightly negative incoming trust.
Chocolate	105	0.00	-0.09	**Minor negative.** No outgoing ratings and mildly negative incoming trust.
Aqua	94	0.11	-0.19	**Minimal negatives.** Sparse interactions with slightly negative feedback.
Silver	77	0.00	-5.02	**Strong negatives.** No outgoing ratings and severely negative incoming trust.
Magenta	74	0.00	0.83	**Passive receivers.** Do not rate others and receive moderate trust.
Navy	65	0.00	-2.01	**Occasional sinks.** No outgoing trust and moderately negative incoming.
Red	51	2.06	1.68	**Structural core.** High outgoing and incoming trust, i.e., central, highly engaged actors.
Black	38	2.45	0.85	**Elite givers.** Very high outgoing trust but moderate incoming, possible key opinion leaders.
Fuchsia	27	0.00	1.02	**Quiet receivers.** Do not rate others but earn some trust.
Yellow	26	0.00	0.00	**Nearly silent.** Almost no activity in either direction.
Grey	21	2.19	1.69	**Generous,** selective. High outgoing trust but slightly less returned.
Maroon	19	0.00	-1.65	**Small negative sinks.** No giving and negative incoming, i.e., distrusted nodes.
Lavender	14	0.00	-1.10	**Tiny negatives.** Very small group with no outgoing and negative incoming trust.

V. CONCLUSIONS

In this study, we set out to uncover the functional roles and relative importance of participants in the Bitcoin OTC trust network through a combination of network embedding, density-based clustering, and centrality analysis. Beginning with the raw signed, weighted edge list, we generated two complementary representations: Node2Vec embeddings to capture community-flow pattern and Struc2Vec embeddings to capture structural equivalence. Each embedding space was then clustered using DBSCAN, and resulting clusters were interpreted by merging in node-level metrics such as average outgoing trust, average incoming trust, and transaction volumes. Finally, we evaluated node importance via a multi-centrality averaging of degree, eigenvector, and closeness centralities to spotlight the most influential actors.

Our results demonstrate that Node2Vec and Struc2Vec yield distinct yet mutually reinforcing insights into network behavior. The Node2Vec clusters highlighted communities of everyday traders, well-liked regulars, optimistic raters, and a delineated distrusted cohort. This revealed how transactional interactions define roles in a flow-based context. In contrast, the Struc2Vec clusters uncovered a broader spectrum of structural roles, from backbone participants to elite givers and strong negatives, showing how positional similarity in the broader topology corresponds to nuanced trust-exchange behaviors. Together, these role profiles, summarized in Tables II and III, provide a multifaceted understanding of trust dynamics that neither approach could deliver alone.

Beyond role discovery, our node importance analysis identified the key players whose positions and connectivity bestow them with strategic influence. By integrating centrality measures, we not only confirmed the prominence of elite hubs detected via clustering but also quantified their standing relative to the wider population. These findings have practical implications for risk management in OTC trading: network operators can use embedding-based role detection to recommend or flag counter-parties, and regulators might focus scrutiny on clusters exhibiting negative or unreciprocated trust.

In sum, this work illustrates the power of combining multiple network embedding techniques with density-based clustering and centrality evaluation to map the complex landscape of trust in peer-to-peer financial networks. Future research could extend this framework by incorporating temporal dynamics to track role evolution, experimenting with alternative clustering or embedding algorithms, and integrating external metadata (e.g., transaction volumes in USD) to further refine role interpretations. Such enhancements would deepen our ability to detect emerging risks and bolster confidence in decentralized trading environments.

REFERENCES

[1] "What is Bitcoin OTC Trading," Bitcoin.com, 2025. https://www.bitcoin.com/get-started/what-is-bitcoin-otc-trading/ (accessed Jun. 26, 2025).

[2] "What is OTC crypto trading, and how does it work?," Cointelegraph, Mar. 2024. https://cointelegraph.com/explained/crypto-otc-trading (accessed Jun. 26, 2025)

[3] Wang, Y., Yao, Y., Tong, H., Xu, F., and Lu, J. "A Brief Review of Network Embedding," Big Data Mining and Analytics, vol. 2, no. 1, pp. 35–47, Mar. 2019, doi: https://doi.org/10.26599/bdma.2018.9020029

[4] K. Huang, C. Xiao, L. M. Glass, M. Zitnik, and J. Sun, "SkipGNN: predicting molecular interactions with skip-graph networks," Scientific Reports, vol. 10, no. 1, Dec. 2020, doi: https://doi.org/10.1038/s41598-020-77766-9.

[5] S. Kumar, F. Spezzano, V.S. Subrahmanian, C. Faloutsos. Edge Weight Prediction in Weighted Signed Networks. IEEE International Conference on Data Mining (ICDM), 2016.

[6] S. Kumar, B. Hooi, D. Makhija, M. Kumar, V.S. Subrahmanian, C. Faloutsos. REV2: Fraudulent User Prediction in Rating Platforms. 11th ACM International Conference on Web Searchand Data Mining (WSDM), 2018.

[7] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 855-864

[8] Network Embedding for Role Discovery: Concepts, Tools, and Applications, 2022. https://markheimann.github.io/tutorials/NetworkRoleDiscovery (accessed Jul. 02, 2025).

[9] P. Szczurek, "Role Discovery in Networks," presented at: DATA-61000-00 Lecture, Lewis University, Romeville, IL, Jun. 2025.

[10] L. Ribeiro, P. Saverese, and D. Figueiredo, "struc2vec: Learning Node Representations from Structural Identity," doi: https://doi.org/10.1145/3097983.3098061.

[11] Figueiredo, Daniel & Ribeiro, Leonardo F. R. & Saverese, Pedro. (2017). struc2vec: Learning Node Representations from Structural Identity. 10.48550/arXiv.1704.03165.