Identifying Circular Trading in Transactional Data with Multi-Stage Clustering

Preprocessing and Data Analysis:

In the dataset provided we found that there were self edges and edges with 0 cost transactions. We removed these edges as they do not corroborate well with real world data.

We use Benford Analysis to understand how likely the data captured in the given timeframe is to be naturally occuring or not.

On performing Benford Analysis there was a Mean Absolute deviation of 0.017076 which indicates that there is a non-conformity between the expected and observed probabilities.

A brief overview of workflow can be found here and the final results can be found at the end.

Clustering Stage 1:

Edge-Betweenness Centrality Clustering:

This method of clustering is very slow. To remove each edge the algorithm has to find the paths through each edge and then remove them. So it is not practical to run this on our dataset directly.

HCS Clustering:

Here, first we find the minimum number of edges to remove to disconnect the graph. This again is a time taking process and cannot be used to used to find circular trading in our dataset directly.

Collusion Clustering:

This algorithm is cubic in the number of vertices and again cannot be used to directly on the dataset.

Clustering Stage 2:

As we can see, most of the clustering methods outlined above take a lot of time to run and are not practical. Some of the clustering methods like HCS do not take the direction of edges into consideration and those that do, like sNN and mNN cannot guarantee that the clusters that they output will contain cycles.

The second stage of our clustering aims to tackle these problems. We first convert the graph our directed graph of transactions into an undirected graph using the method described in this paper. This greatly reduces the size of the graph, with the number of nodes falling from more than 6000 to around 1500.

Once we have a smaller graph we can run various clustering algorithms mentioned in stage 1 to find cycles. Since the graph only contains edges that participate in cycles we can also guarantee that there will be cycles in the identified clusters.

We can see in the results that mNN now runs very well for identifying cycles. Algorithms that take longer time like HCS, Collusion Clustering, Edge-betweenness centrality can also be used in real time by running the stock flow graph through knn_filter before using 3-cycle weighting.

sNN works on the original transaction graph, but can result in some large clusters in which finding clusters can be challenging. Taking these large clusters and running them through other clustering before or after running them through 3-cycle weighting.

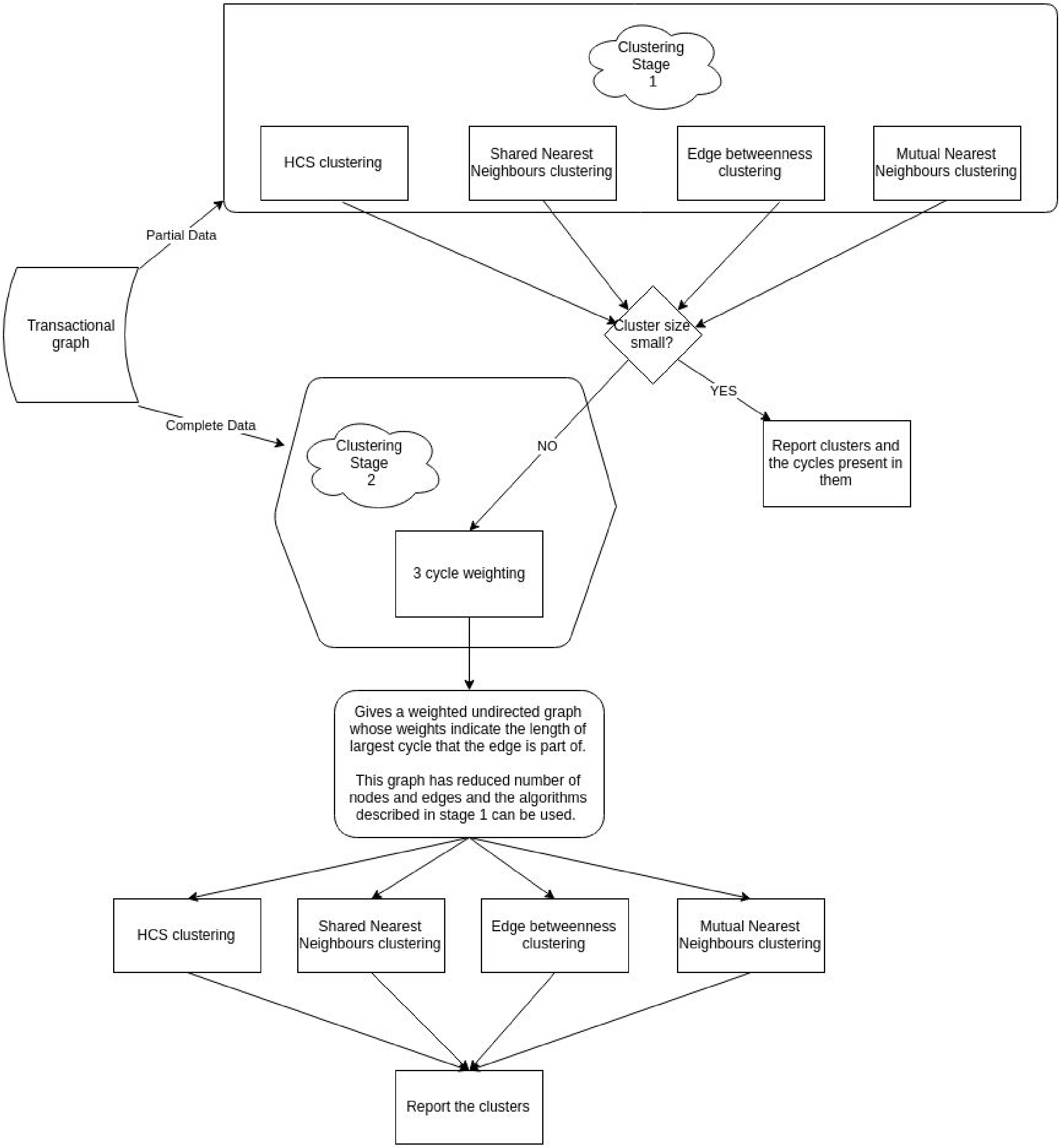
Output and Evaluating the Clusters:

We get the final set of clusters by combining the clusters from all the methods. Then we return clusters of reasonable size (3 <= size <= 10) grouped by their size and then sorted by the relative transactional cost (transactional cost multiplied by the number of edges).

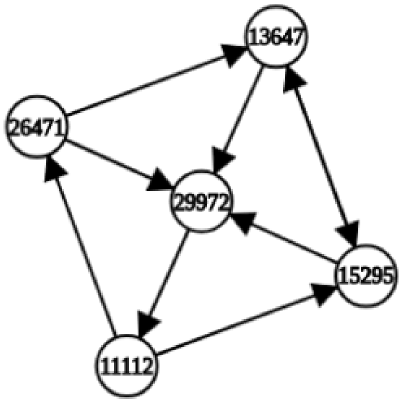
This method of selecting clusters to display was chosen over other methods like Dempster-Shafer theory of evidence because they are computationally very expensive for larger clusters (size > 7).

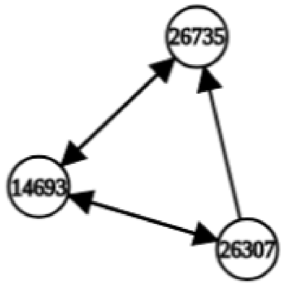
We are multiplying by the number of edges because besides transactional cost, having more edges in a cluster is a good indicator of heavy circular trading.

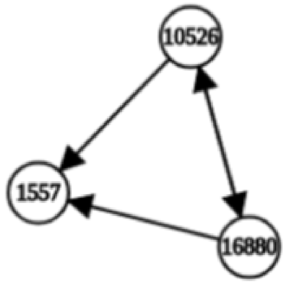
Workflow of the Algorithm

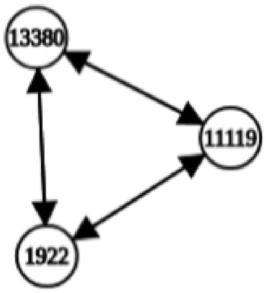

Clustering
Stage
1
HCS clustering
Shared Nearest Neighbours clustering
Edge betweenness clustering
Mutual Nearest Neighbours clustering
Partial Data
Transactional graph
Cluster size small?
YES
Report clusters and the cycles present in them
Complete Data
Clustering
Stage
2
NO
3 cycle weighting
Gives a weighted undirected graph whose weights indicate the length of largest cycle that the edge is part of.
This graph has reduced number of nodes and edges and the algorithms described in stage 1 can be used.
HCS clustering
Shared Nearest Neighbours clustering
Edge betweenness clustering
Mutual Nearest Neighbours clustering
Report the clusters

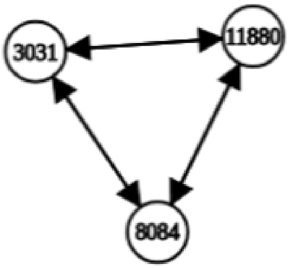
Clusters with sNN:

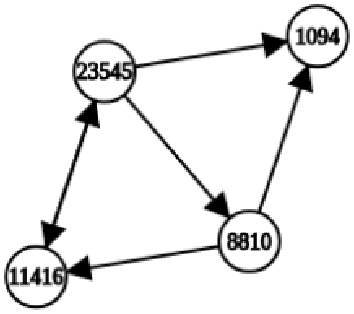




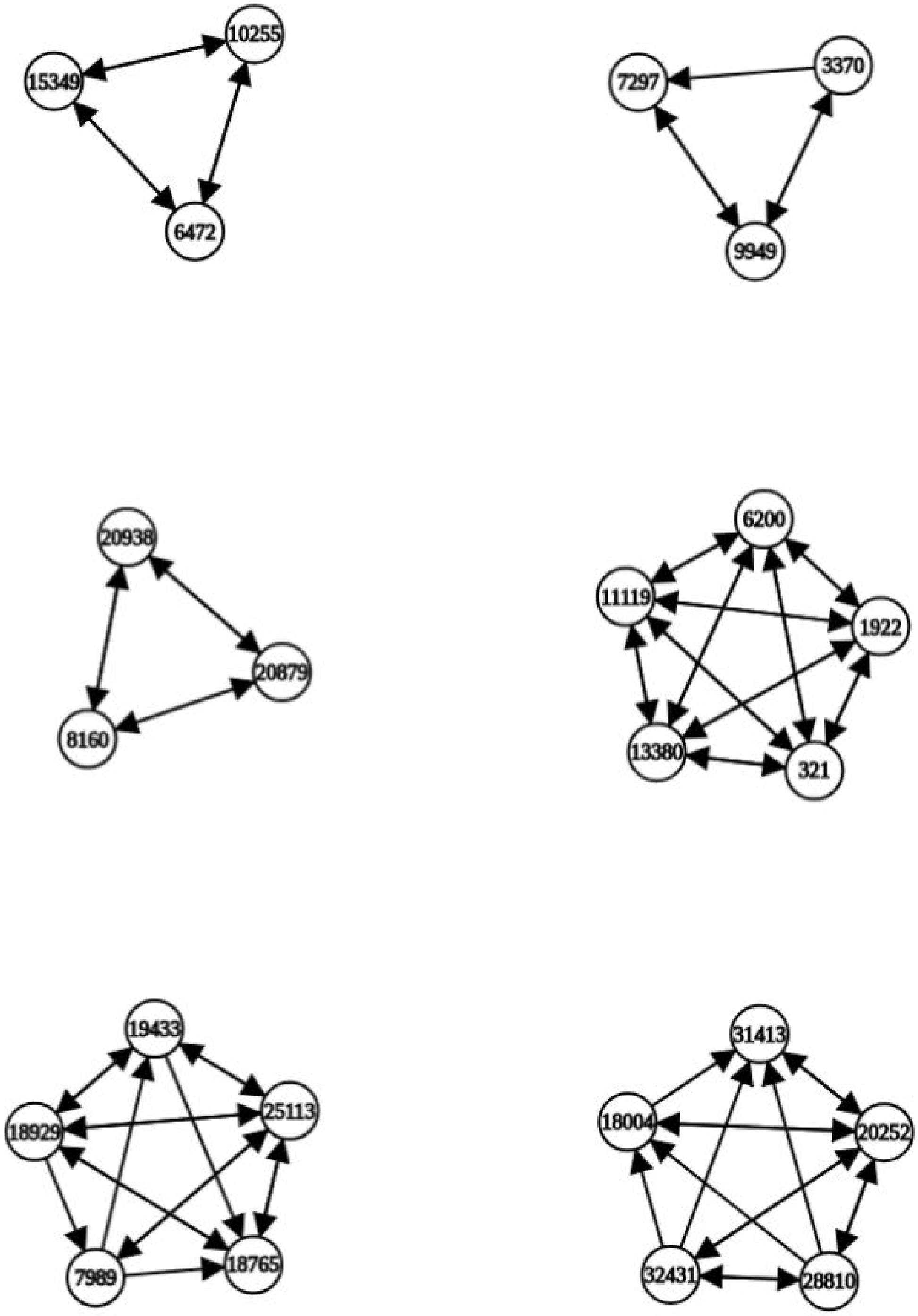




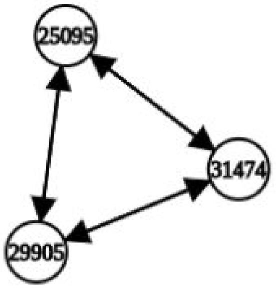


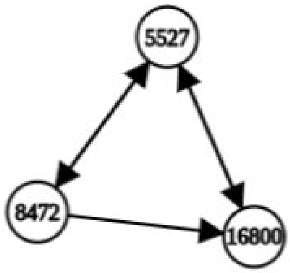


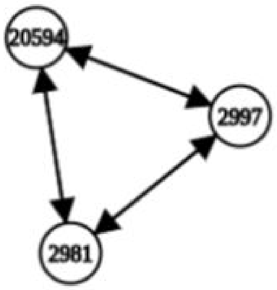
Clusters with mNN on weighted cycle graph

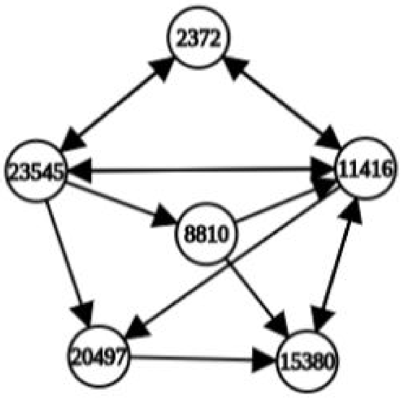


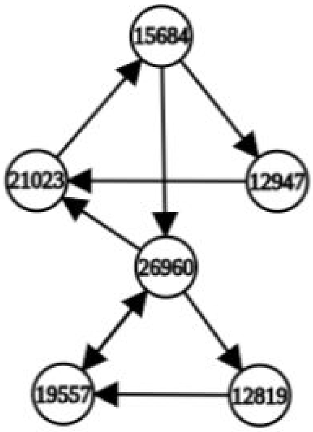
Clusters with Edge- Betweenness Centrality based Clustering

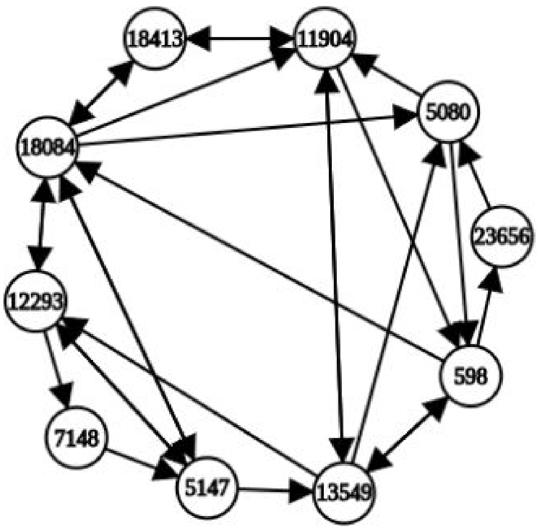




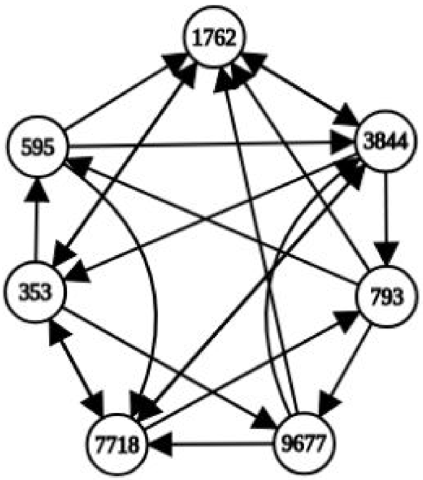


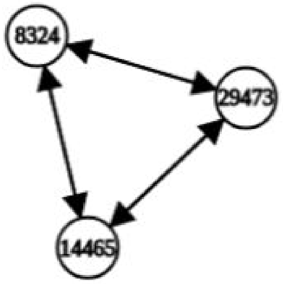


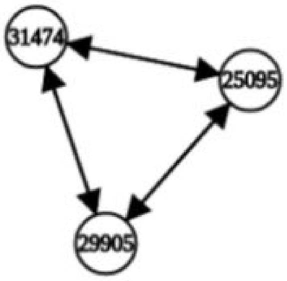


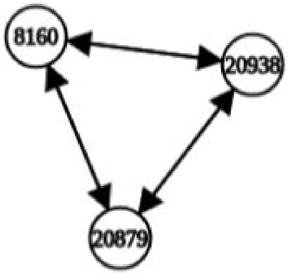


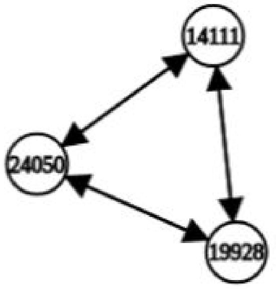
Clusters with HCS on weighted cycle graph of knn filtered graph

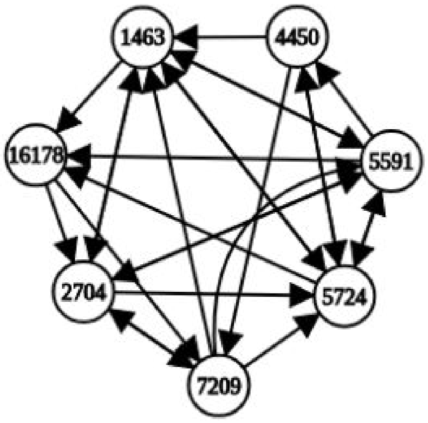




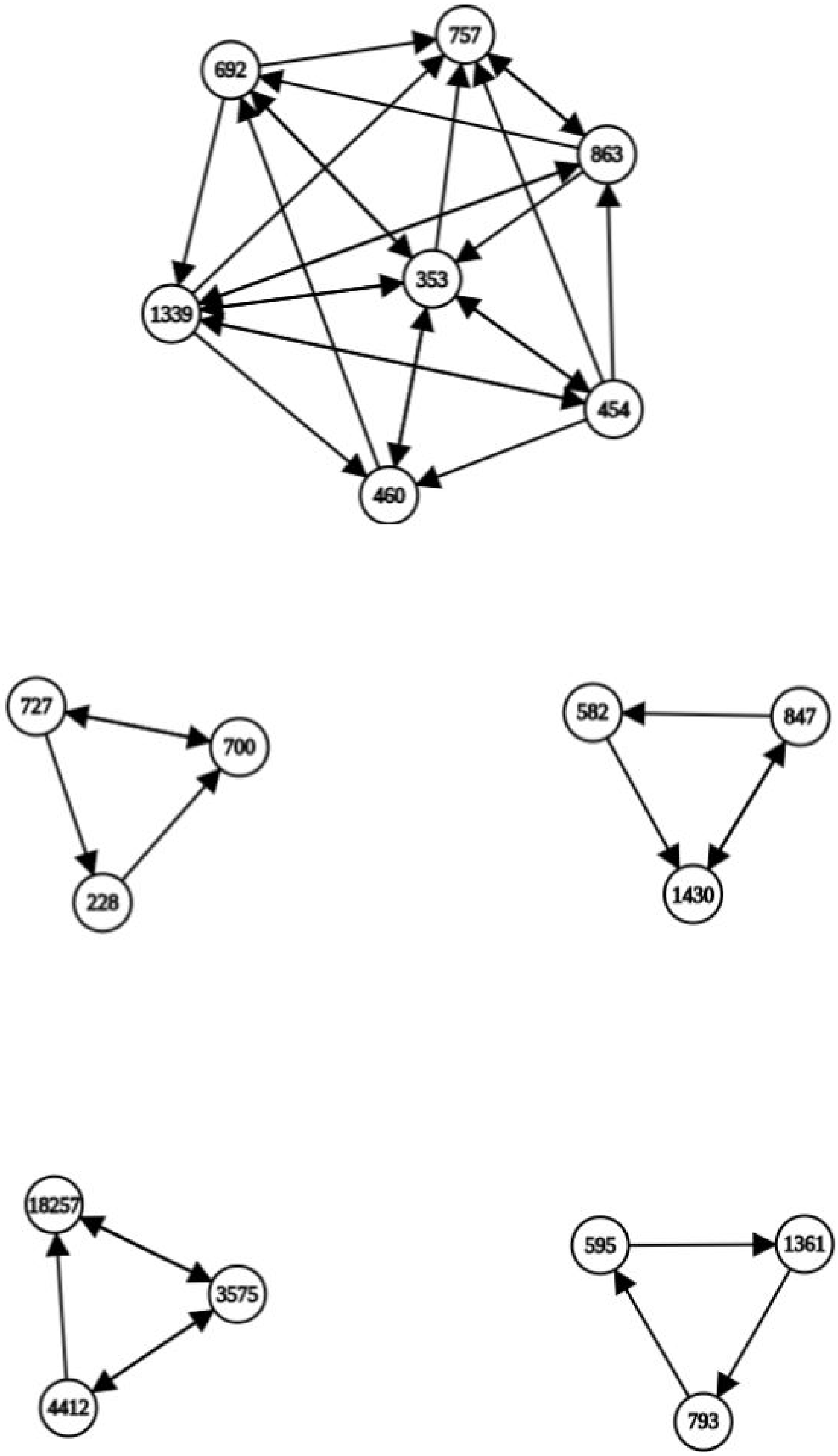




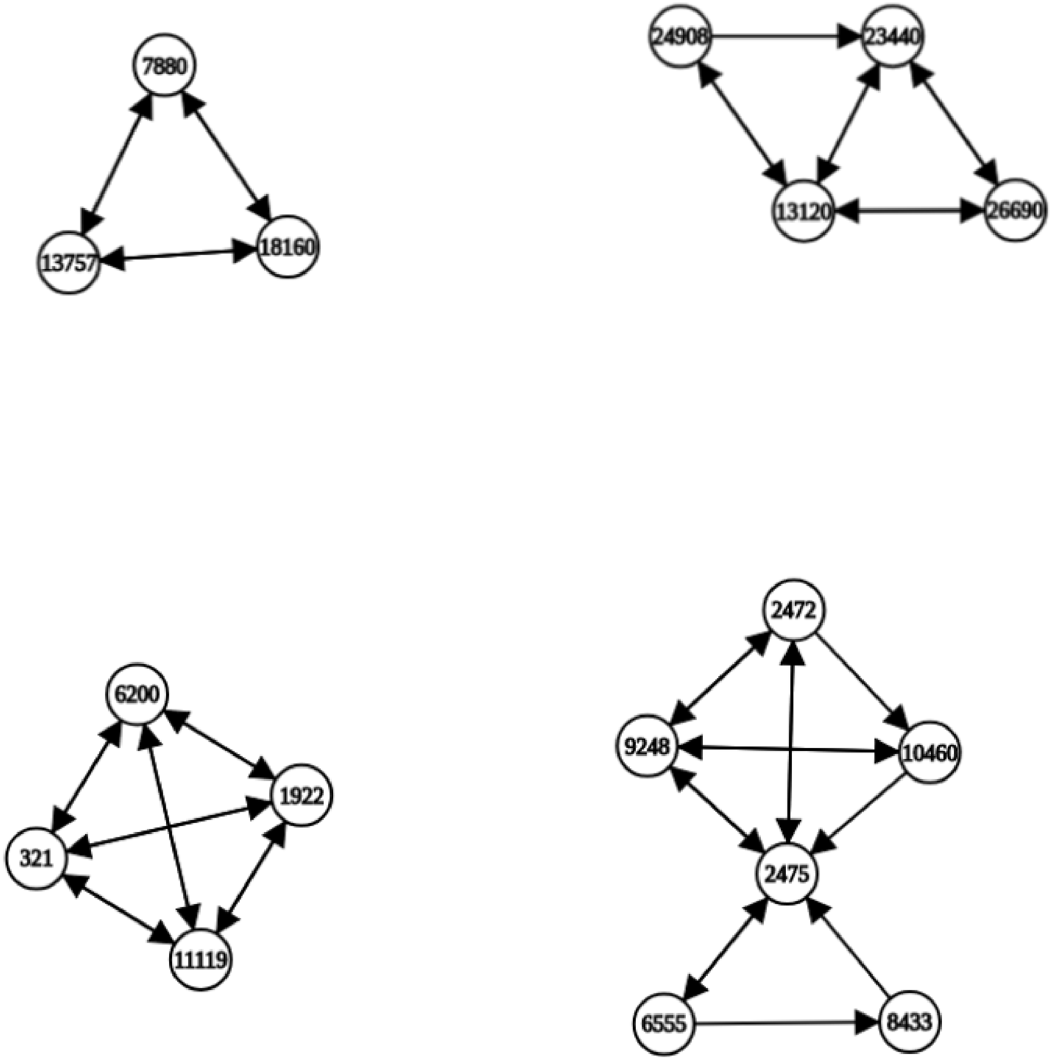




Clusters with HCS on weighted cycle graph of partial data



Clusters with Collusion Clustering on weighted cycle graph


7880
13757
18160
24908
23440
13120
26690
6200
1922
321
11119
2472
9248
10460
2475
6555
8433