

Bitcoin Alpha trust weighted signed network

(HW 1 report Task 1)

Bita Naserfarahmand

Student ID: 2021380075

Email: bita.nf@gmail.com

Computational sociology class

Content:

1. Background..... 3

2. Goal..... 4

3. Solutions..... 5

4. Findings..... 14

1)Background:

Homework 1 task 1:

Question:

Perform network analysis on one large-scale dataset:

- **Crawl one real-world dataset or download one public dataset**
- **Size of nodes: > 1000;**
- **Size of edges: > 3000;**
- **Quantify the centrality of nodes including degree, betweenness, closeness**
- **Visualize the network topology**

Apparatus:

Gephi software

Dataset Information:

This is who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. Members of Bitcoin Alpha rate other members in a scale of -10 (total distrust) to +10 (total trust) in steps of 1.

Nodes : 3783

Edges : 24186

Range of edge weight : -10 to +10

Percentage of positive edges : 93%

Here is s a brief look on how the dataset looks like in the excel format before being imported to Gefi:

SOURCE	TARGET	RATING	TIME
7188	1	10	1.41E+09
430	1	10	1.38E+09
3134	1	10	1.37E+09
3026	1	10	1.35E+09
3010	1	10	1.35E+09
804	1	10	1.34E+09
160	1	10	1.39E+09
95	1	9	1.38E+09
377	1	7	1.41E+09
888	1	7	1.37E+09
89	1	7	1.35E+09
1901	1	6	1.41E+09
161	1	6	1.41E+09
256	1	6	1.34E+09
351	1	5	1.42E+09
3129	1	5	1.39E+09
3341	1	5	1.39E+09
649	1	5	1.38E+09
1583	1	5	1.38E+09
87	1	5	1.38E+09
37	1	5	1.37E+09
309	1	5	1.38E+09
821	1	5	1.37E+09
1496	1	5	1.39E+09
637	1	5	1.36E+09
964	1	5	1.35E+09
594	1	5	1.37E+09
2249	1	5	1.36E+09

The dataset has been divided into 4 columns, source, target, rating and time, as can be seen in the above picture.

SOURCE: node id of source, i.e., rater

TARGET: node id of target, i.e., ratee

RATING: the source's rating for the target, ranging from -10 to +10 in steps of 1

TIME: the time of the rating, measured as seconds since Epoch.

2)Goal:

The goal of this project is to import the dataset into Gephi and visualize the data. This dataset is a who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. Since Bitcoin users are anonymous, there is a need to maintain a record of users' reputation to prevent transactions with fraudulent and risky users. So by the end of the project, one should be able to distinguish between a trustworthy and a fraud trader.

The main focus for this project is the big traders (big nodes) and to see whether or not they are trustworthy or not. For this purpose, a clear and easy graph must be created to make the analysis easy.

This report will show some changes being made to the graph step by step to reach a clear graph.

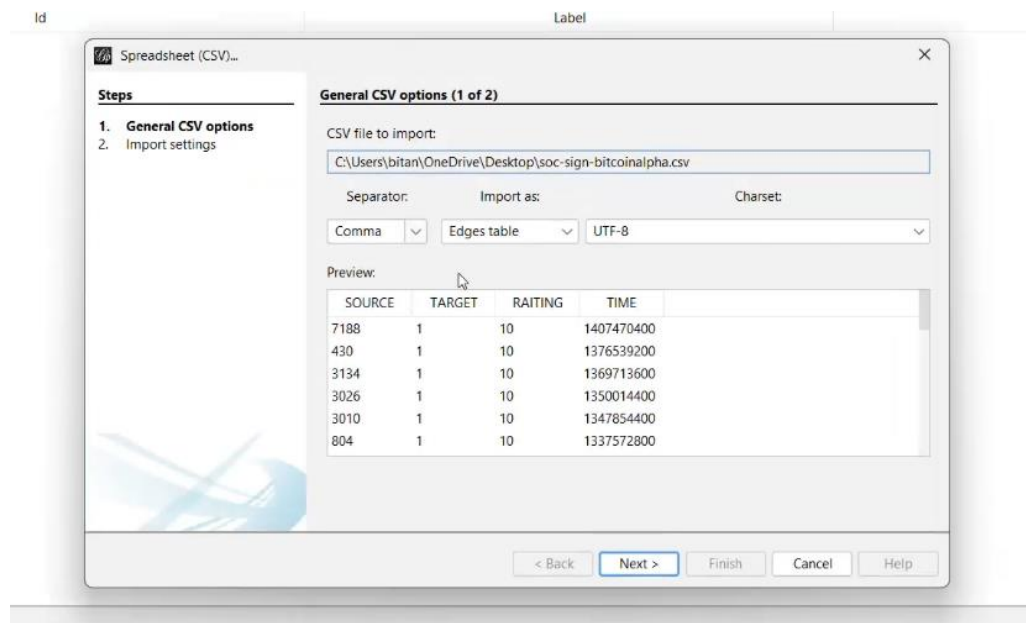
Unlike most datasets, this dataset focuses on edges instead of nodes, so the edges need to be distinguishable.

Since the question asked for closeness and in-betweenness calculations, this Report provides them however, these are attributes related to the nodes which we do not need.

At the end of this report some example analysis will be made on the biggest nodes in the graph to show how one can tell them apart.

3)Solutions:

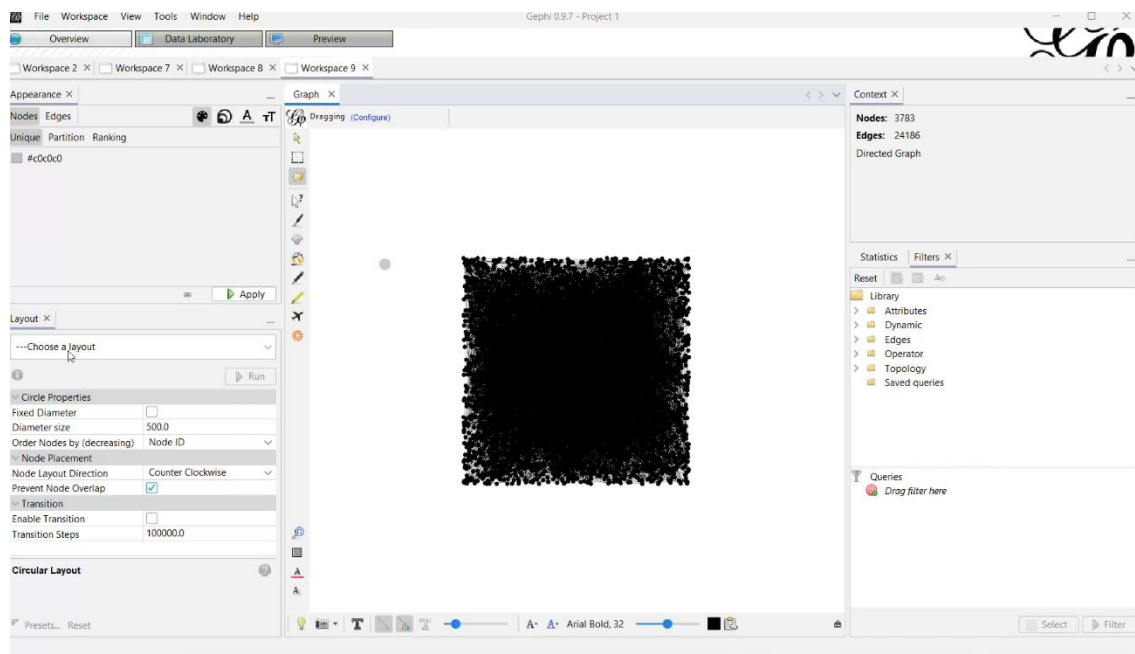
Importing the dataset to Gephi:



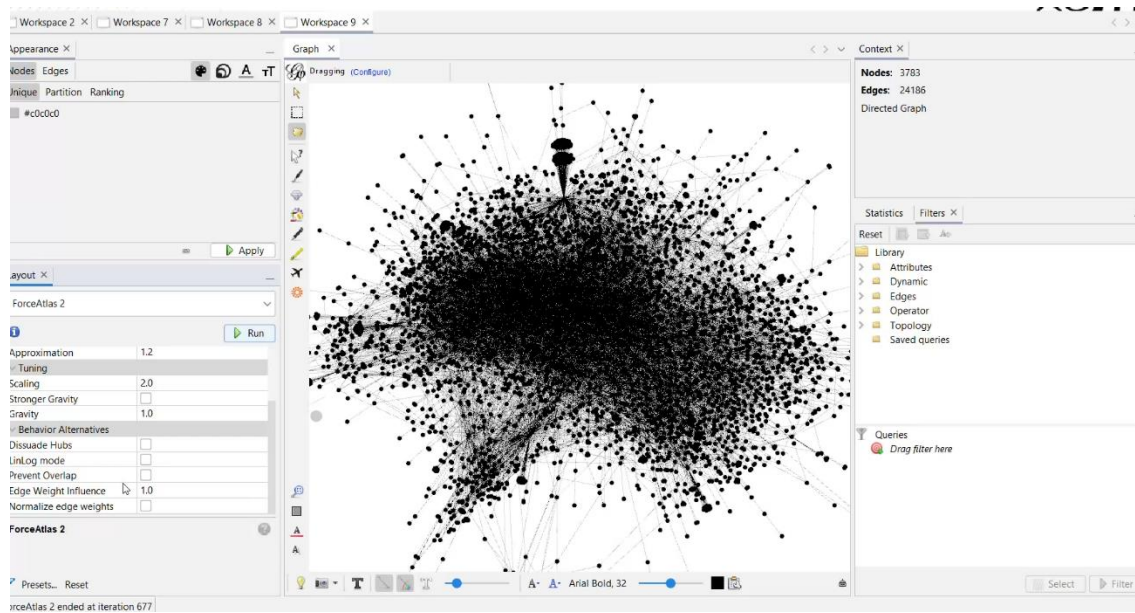
Source	Target	Type	Id	Label	Interval	Weight	Rating	time
7188	1	Directed	72558			1.0	10	1407470400
430	1	Directed	72559			1.0	10	1376539200
3134	1	Directed	72560			1.0	10	1369713600
3026	1	Directed	72561			1.0	10	1350014400
3010	1	Directed	72562			1.0	10	1347854400
804	1	Directed	72563			1.0	10	1337572800
160	1	Directed	72564			1.0	10	1394683200
95	1	Directed	72565			1.0	9	1384578000
377	1	Directed	72566			1.0	7	1414728000
888	1	Directed	72567			1.0	7	1365652800
89	1	Directed	72568			1.0	7	1351742400
1901	1	Directed	72569			1.0	6	1411790400
161	1	Directed	72570			1.0	6	1413345600
256	1	Directed	72571			1.0	6	1342584000
351	1	Directed	72572			1.0	5	1416373200
3329	1	Directed	72573			1.0	5	1389934800
3341	1	Directed	72574			1.0	5	1388984400
649	1	Directed	72575			1.0	5	1384491600
1583	1	Directed	72576			1.0	5	1380945600
87	1	Directed	72577			1.0	5	1381809600
37	1	Directed	72578			1.0	5	1374033600
309	1	Directed	72579			1.0	5	1378353600
821	1	Directed	72580			1.0	5	1368676800
1496	1	Directed	72581			1.0	5	1394168400
637	1	Directed	72582			1.0	5	1364961600
964	1	Directed	72583			1.0	5	1350100800
594	1	Directed	72584			1.0	5	1370836800
2249	1	Directed	72585			1.0	5	1360386000

Our focus is the edges in our dataset. Shown above in the data library.

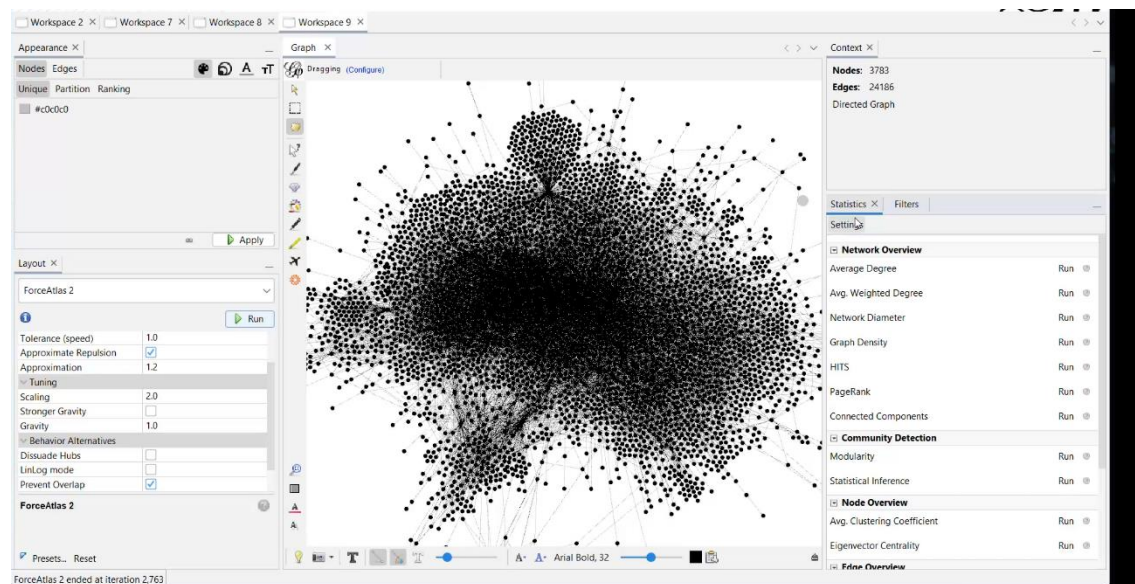
First look at the graph:



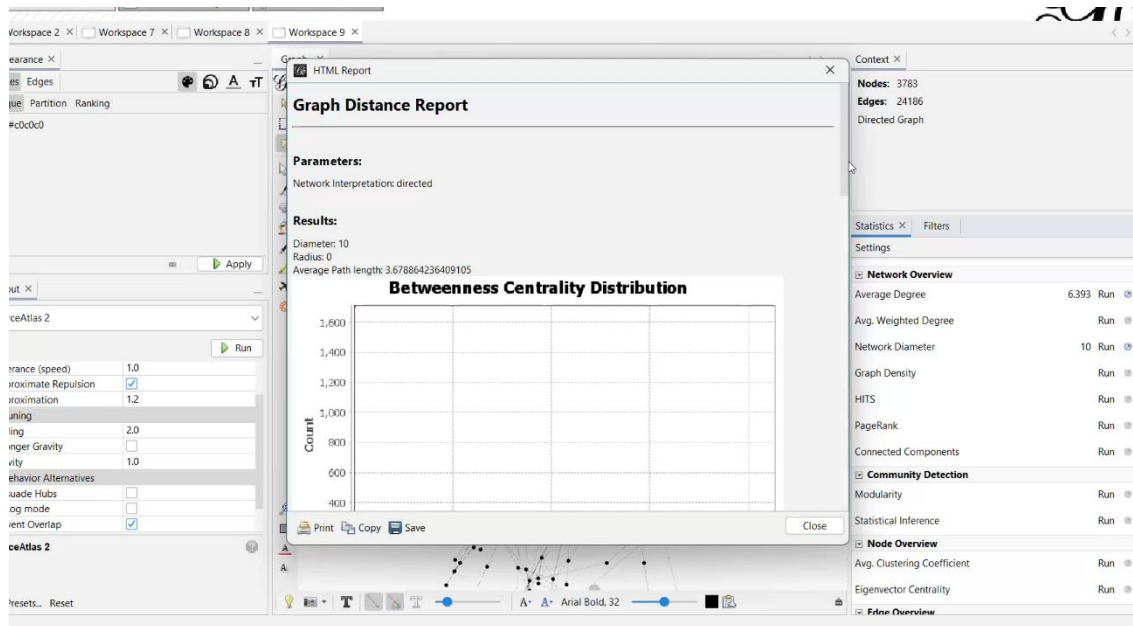
Using ForceAtlas2 layout:



Prevent overlap:



From statistics, run network diameter:



Now looking at the dataset it can be seen that the centrality of nodes including degree, betweenness, closeness has been calculated and added:

Overview

Data Laboratory

Preview

Workspace 2 x Workspace 7 x Workspace 8 x Workspace 9 x

Data Table x

Nodes Edges Configuration Add node Add edge Search/Replace Import Spreadsheet Export table More actions

Filter: Id

Id	Label	Interval	In-Degree	Out-Degree	Degree	Eccentricity	Closeness Centrality	Harmonic Closeness Centrality	Betweenness Centrality
7188	seller	0	1	1	6.0	0.293546	0.312198	0.0	
1	seller	398	490	888	5.0	0.41541	0.479993	2132893.99254	
430	seller	6	9	15	6.0	0.316684	0.337563	10350.766924	
3134	seller	3	4	7	6.0	0.31792	0.337163	86.776374	
3026	seller	1	1	2	6.0	0.293514	0.312147	0.0	
3010	seller	1	1	2	6.0	0.293514	0.312147	0.0	
304	seller	7	7	14	6.0	0.323771	0.34355	4291.803773	
160	seller	9	10	19	6.0	0.308547	0.329335	11852.479276	
35	seller	101	122	223	6.0	0.391782	0.425687	24659.308969	
377	seller	4	4	8	6.0	0.293859	0.312904	10996.973232	
388	seller	4	5	9	6.0	0.309107	0.328076	3231.181936	
39	seller	49	46	95	6.0	0.352725	0.377119	111176.661537	
1901	seller	1	1	2	6.0	0.293514	0.312147	0.0	
161	seller	35	36	71	6.0	0.347556	0.372667	26772.987411	
256	seller	30	30	60	6.0	0.344805	0.368037	51644.960557	
351	seller	14	16	30	6.0	0.329929	0.353558	11703.437516	
3329	seller	1	1	2	6.0	0.293514	0.312147	0.0	
3341	seller	1	1	2	6.0	0.293514	0.312147	0.0	
549	seller	14	13	27	6.0	0.32526	0.346637	3843.617305	
1583	seller	1	1	2	6.0	0.293514	0.312147	0.0	
37	seller	54	59	113	6.0	0.365418	0.393071	89031.676965	
37	seller	46	54	100	6.0	0.374139	0.400747	127213.066427	
309	seller	23	23	46	6.0	0.340946	0.364621	24796.391556	
321	seller	4	4	8	6.0	0.304238	0.323219	1417.391517	
1496	seller	2	2	4	6.0	0.296651	0.315417	0.0	
337	seller	7	7	14	6.0	0.323268	0.345067	2183.924088	
364	seller	6	4	10	6.0	0.315563	0.334935	3458.552999	
394	seller	5	5	10	6.0	0.306804	0.326559	1556.116702	

Add column

Merge columns

Delete column

Clear column

Copy data to other column

Fill column with a value

Duplicate column

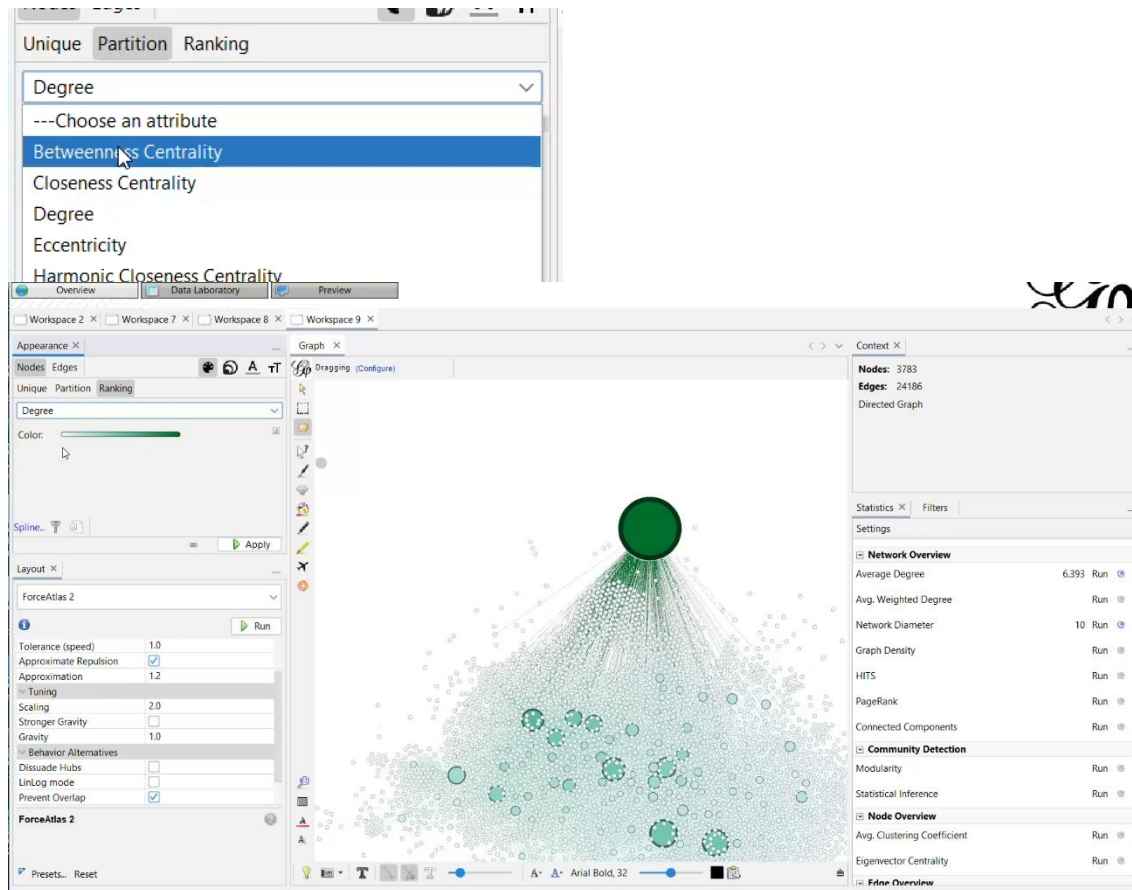
Create a boolean column from regex match

Create column with list of regex matching groups

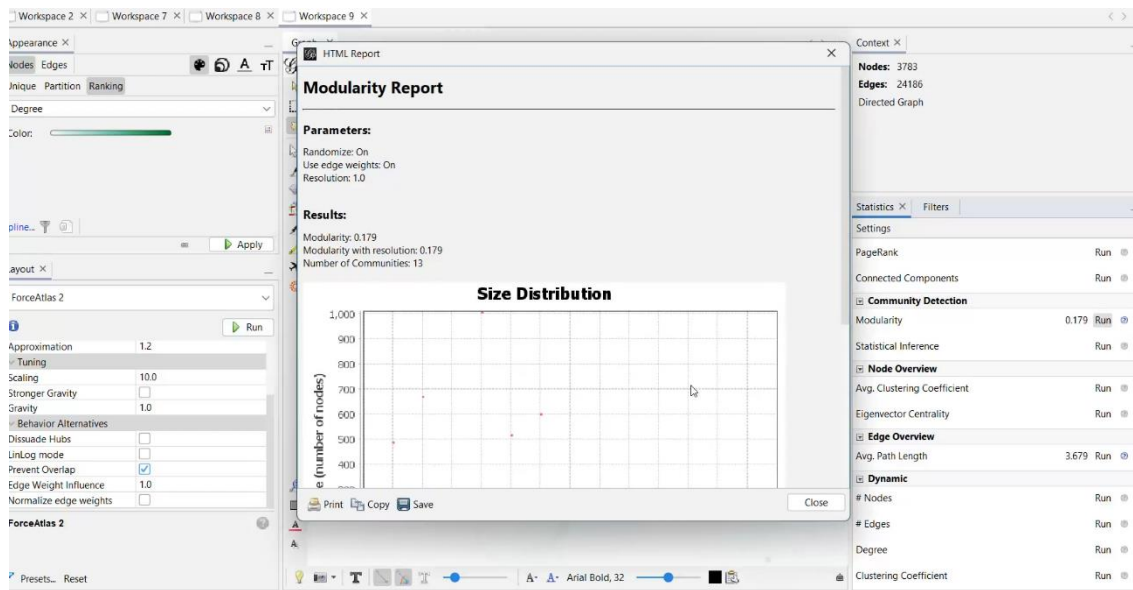
Negate boolean values

Convert column to dynamic

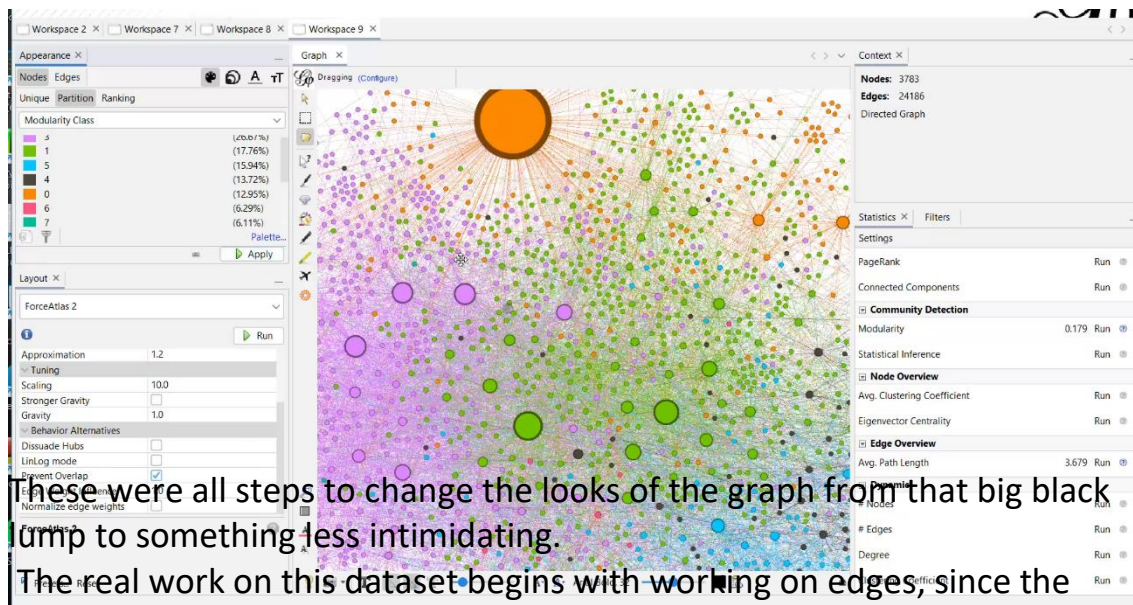
Changing the color of nodes based on betweenness centrality and make the scaling bigger so that the graph will expand (from 2.0 to 10.0):



Run community detection in the statistics section to get the modularity report to see different communities which was not a necessary step but it makes the graph more pleasing to the eyes:



This is how the graph looks when the class color is applied to the nodes:

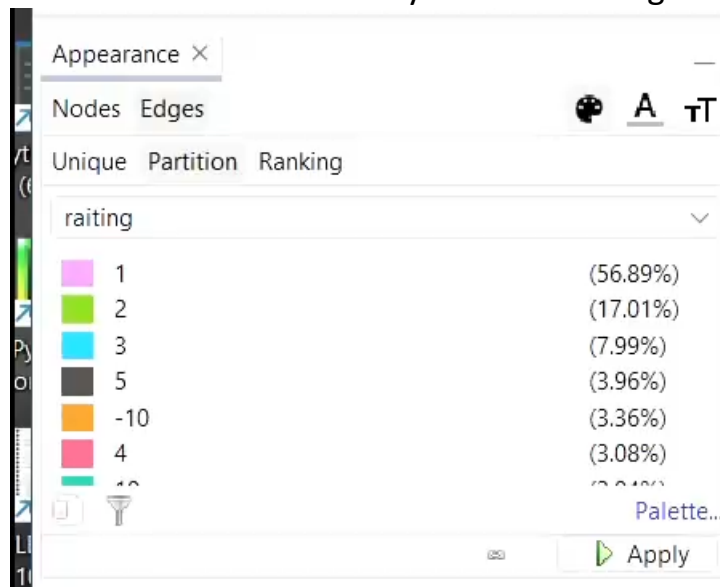


dataset has weighted edges, the best choice was to color code the rating of the edges. Edges have weights from -10 to +10, which is the rating of

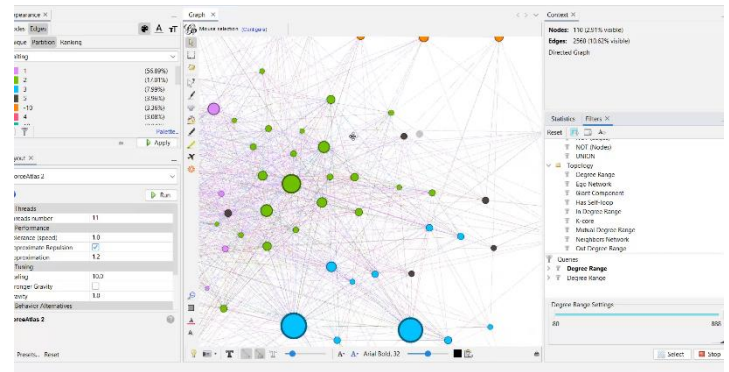
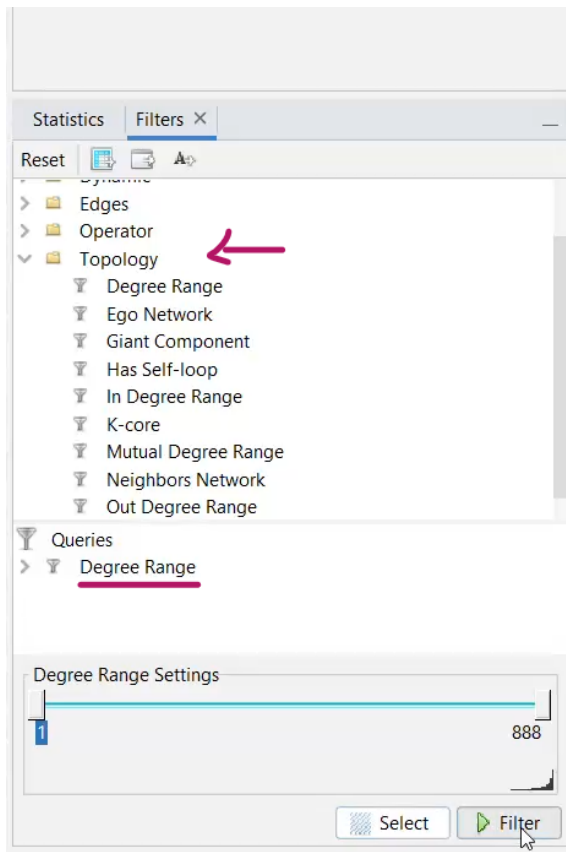
the traders.

For example, from the picture below it can be seen that 56.89% of the edges in the graph have 1 rating, so not many are satisfied with trading with the nodes that have been rated 1. Which is shown by purple in the graph.

Orange means -10 which is the worst rating and green shows 10 which is the highest rating. By looking at the color and the number of the edges going to the nodes, which represent the traders, one can see how many people were satisfied with their service. By choosing the biggest nodes (which shows the ones with the most buyers) we choose the most popular traders and can see if they are fraud or legit.

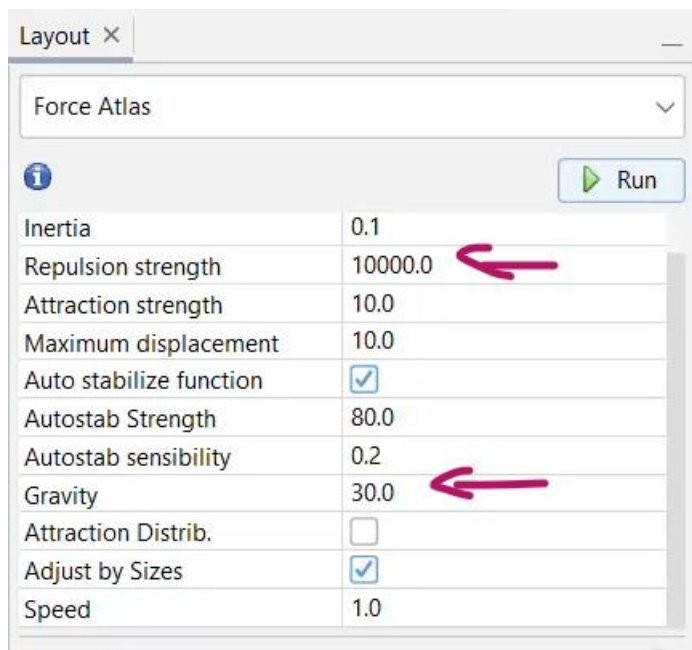


However, the number of edges is too high to see clearly. So in this next step by running topology and choosing degree range, some nodes can be deleted. For example nodes that have 80 edges or less.

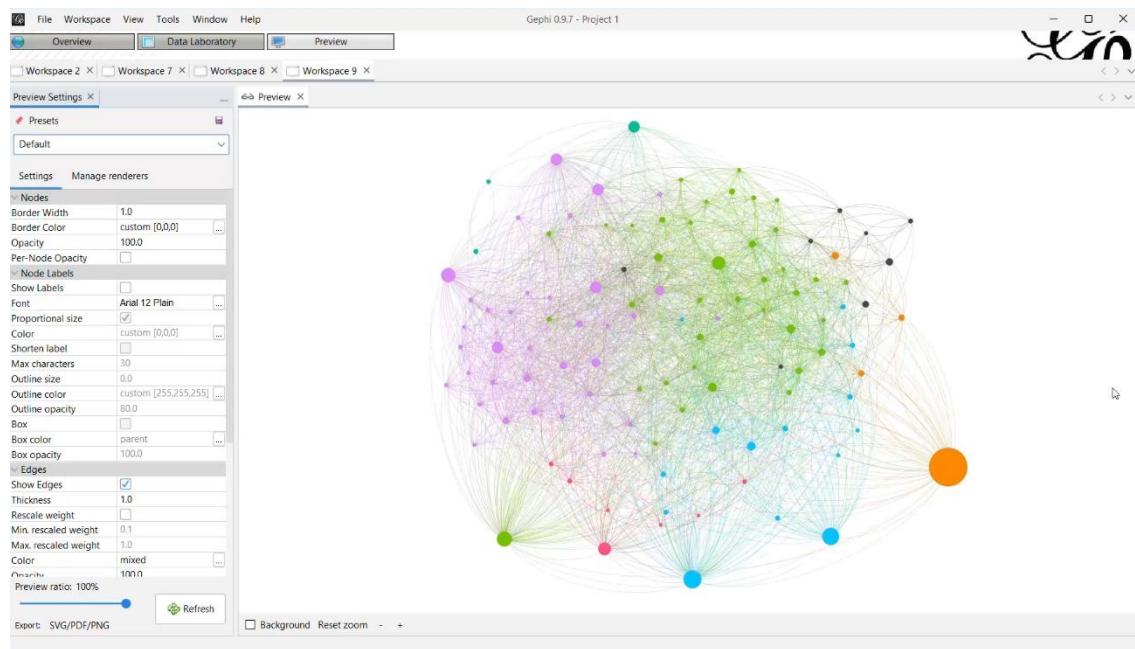


Change to ForceAtlas with characteristics

shown below:



At the end we can look at the preview:



4) Findings:

There were many ways that this graph could be analyzed but in this project the focus was to look at the biggest traders and see if they are a fraud or can be trusted.

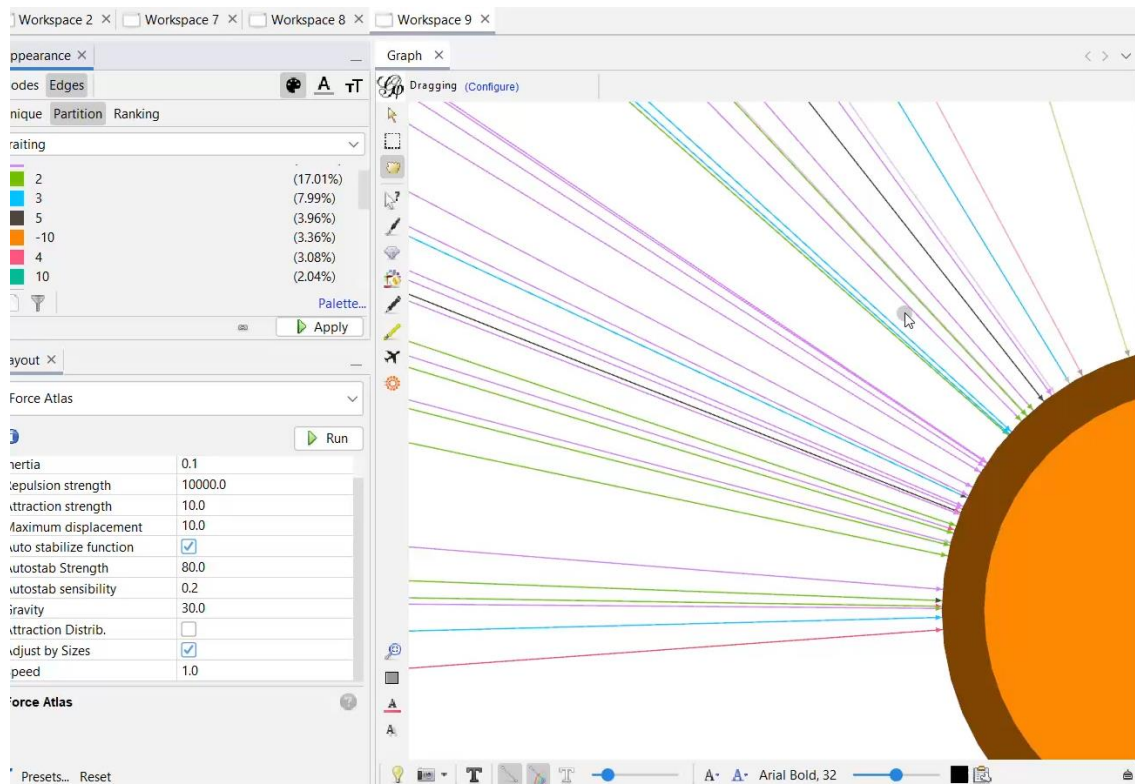
Now that the graph is a bit organized, analyzing the biggest nodes which represent the most popular traders and sellers is possible.

In this section 4 node will be analyzed:

Node1:

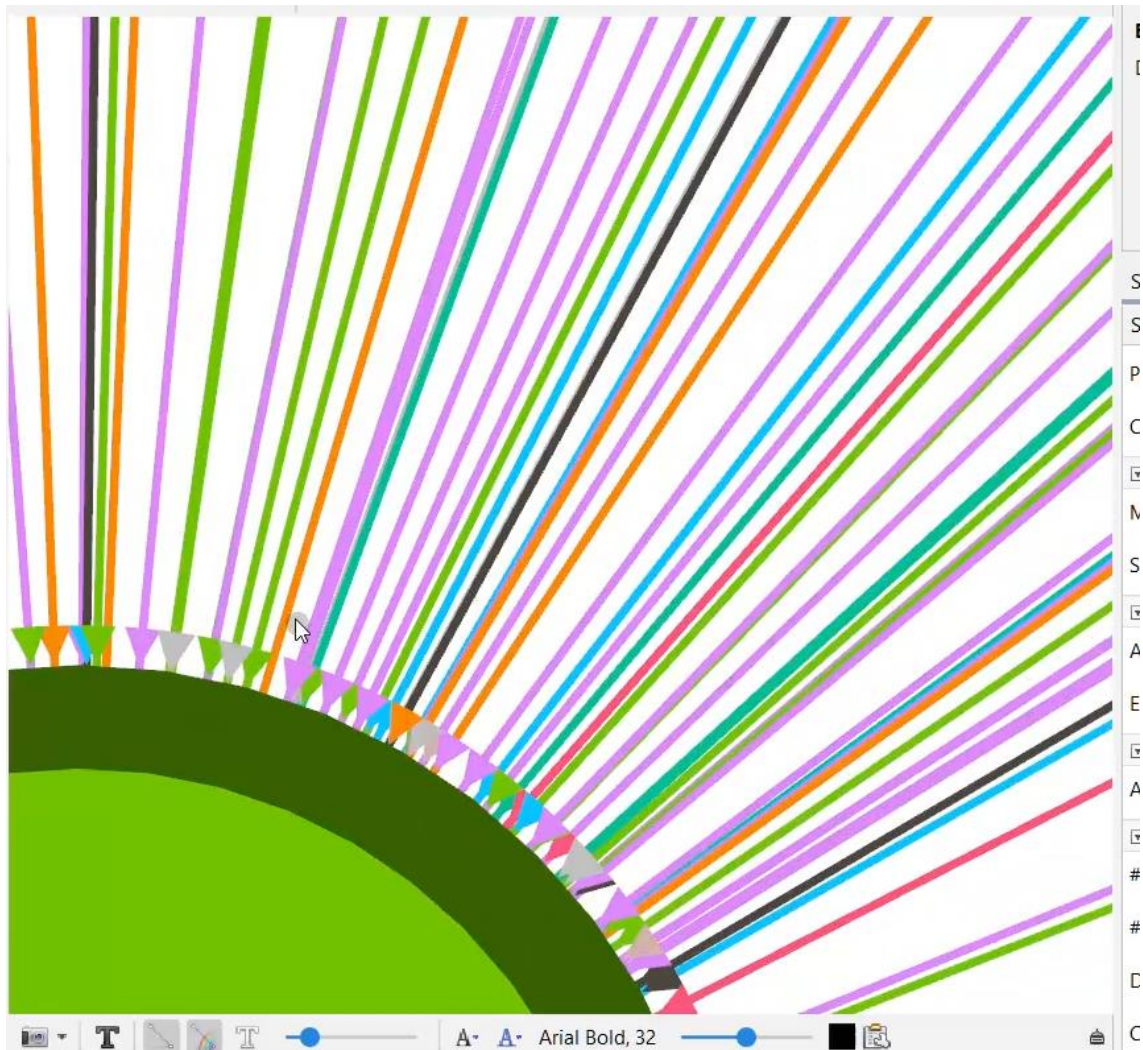
The largest node was dragged further so that we could take a closer look at its edges.

Not many orange edges (-10) going in this node. most buyers are somewhat satisfied with their trade. So, this node does not represent a fraud trader.



Node2:

mostly gave the rating 1. Most people are happy about this one but there is still orange which shows complete dissatisfaction but mostly positive so this node is most likely not a fraud.



Node3:

This node has the most orange edges, so a lot of people gave -10. This trader is not to be trusted as is most likely a fraud.

