NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CZ4071/SC4022 Network Science

Team Project:

Chua Yi Xiang (U2022391F)

Zeng Ruixiao(U2220375D)

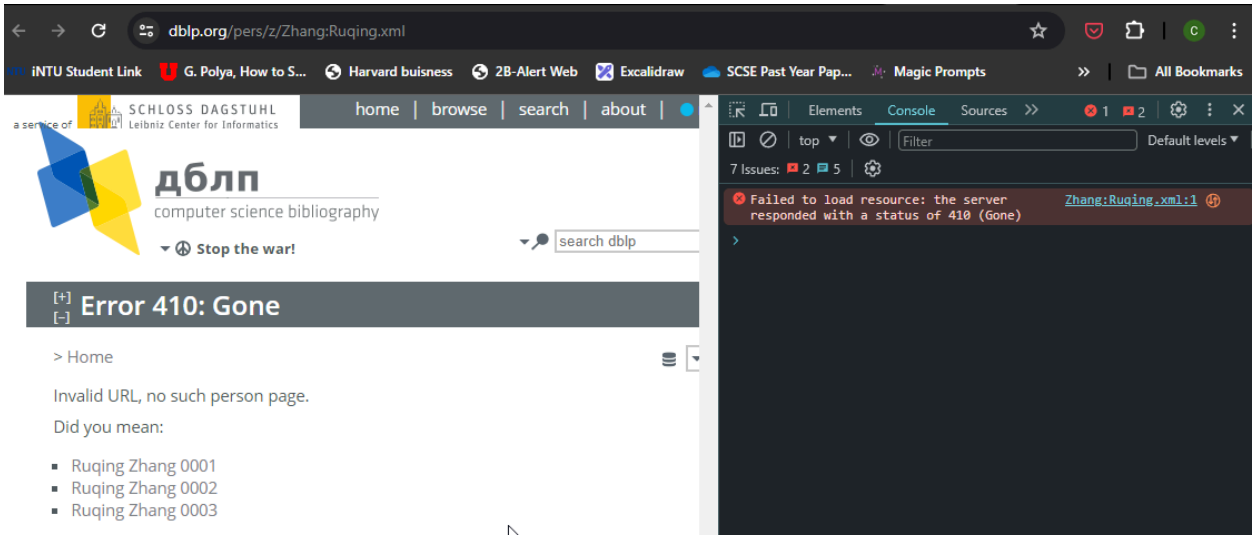
Introduction

This project aims to use network science to analyse patterns in research collaboration among data scientists over time. To achieve this goal, we are given a list of data scientists as an input file, and tasked with the following:

1. Data cleaning
2. Network Construction
3. Network Analysis
4. Network Transformation

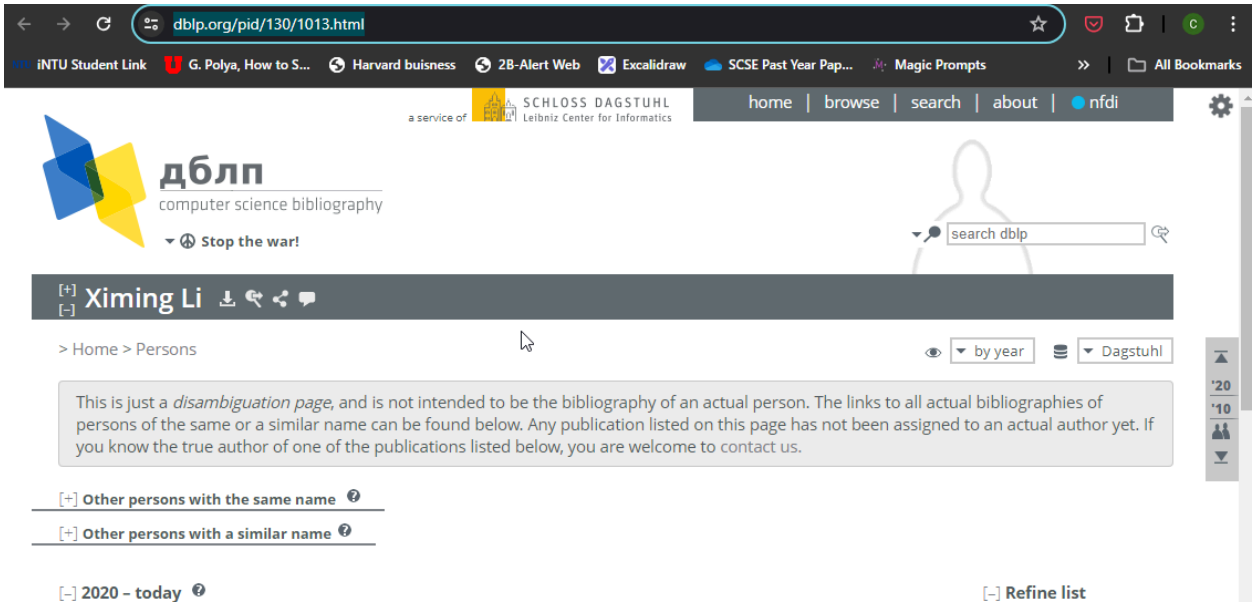
Data Collection and Preprocessing

Before constructing the network, we must ensure the data is clean. There are two main categories of dirty data. The first category refers to dead-end pages where the user no longer exists on the DBLP.



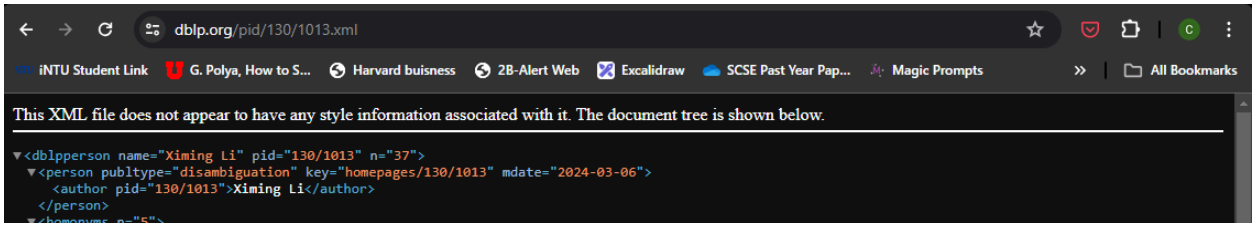
Attempting to access their url will return a status code of 410 (Gone) or 404 (Not Found). Hence, we can retrieve all dblp links from the given list of data scientists, and send a request to each one, removing any rows where the url leads to a dead end.

The second category refers to links that lead to disambiguation pages.



Disambiguation pages are not the bibliographies of an actual person but a collection of bibliographies of authors with similar names. If added, this will affect our network as disambiguation nodes will naturally become hubs due to having unnaturally large numbers of coauthors.

To identify them, DBLP has a feature that allows users to view an XML file of the associated author by replacing the HTML tag at the end of the url with XML. Doing so brings up an XML file.



If the link leads to a disambiguation page, the publtype will be assigned that keyword in the person tag of the XML retrieved. Hence, we can iterate through every XML page of the list of data scientists given, checking the person tag for the disambiguation keyword and removing the row from the list if the keyword exists. We also save each XML file retrieved if the author is valid for later use.

After performing both operations, we finally have a clean dataset and can proceed to build the network graph.

Network Construction

Looking again at the XML file, we notice that each publication made by the author is detailed in a single <r> tag.

```
<r>
 <article key="journals/access/OzansoyZ23" mdate="2023-06-02">
    <author orcid="0000-0002-9375-9571" pid="130/0908">Cagil R. Ozansoy</author>
    <author pid="46/5623">Aladin Zayegh</author>
    <title>Optimal Sampling Requirements for Robust and Fast Vegetation High Impedance Fault Detection.</title>
    <pages>42924-42936</pages>
    <year>2023</year>
    <volume>11</volume>
    <journal>IEEE Access</journal>
    <ee type="oa">https://doi.org/10.1109/ACCESS.2023.3270928</ee>
    <url>db/journals/access/access11.html#OzansoyZ23</url>
  </article>
</r>
```

The <r> tag contains, amongst other things, a list of pid (unique identifier) of all co-authors in the <author> tags, as well as the year in the <year> tag. We now have enough information to construct our network graph.

We use two dictionaries to store the edges and nodes of each year as follows:

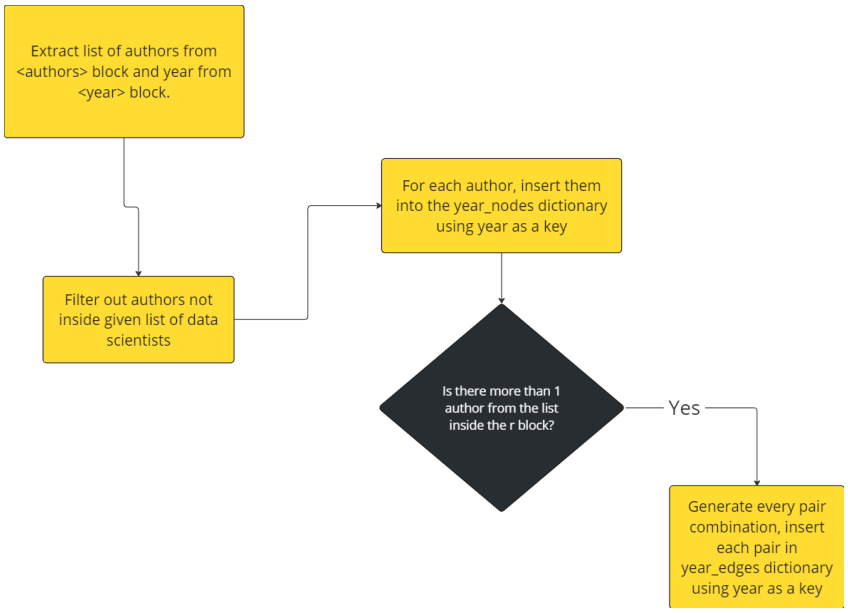
year_edges

{Year xxxx: {(EdgeData Scientist 1, Data Scientist 2), …},
Year yyyy: {(Data Scientist 1, Data Scientist 2), …}}

year_nodes

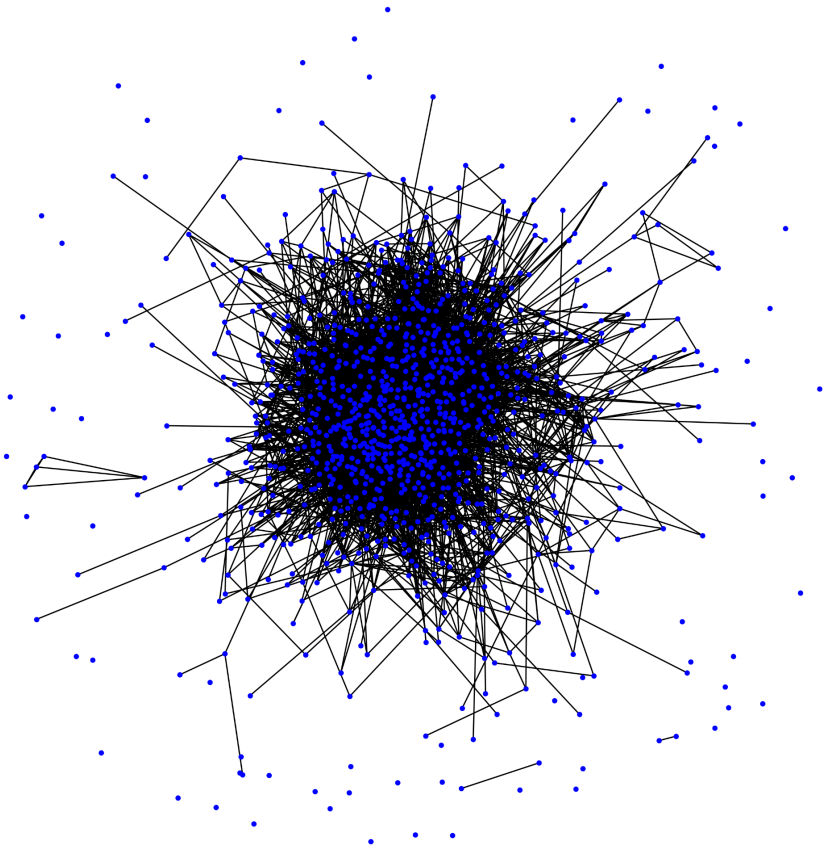
{Year xxxx: {EdgeData Scientist 1, Data Scientist 2, …},
Year yyyy: {Data Scientist 1, Data Scientist 2, …}}

For each data scientist in our cleaned list, we extract every publication (<r> tag) made by the data scientist and perform the following checks:



Network Analysis

Question 1: Properties of collaboration network



Number of nodes: 1028
Number of edges: 7327
Average clustering coefficient: 0.3043159815248051
Average degree: 14.254863813229573
Number of connected components: 69
Density: 0.013880101083962582

Only 1028 of the 1220 scientists had valid URLs. The average clustering coefficient value of 0.304 suggests a moderate level of clustering(within any given subgroup of data scientists, about 30.43% of the potential links (collaborations) that could exist between them exist). From this value, we can infer that data scientists generally tend to collaborate within smaller groups or communities.

However, the average degree per scientist is relatively high at 14.25, indicating that, on average, each data scientist collaborates with 14 others. Overall, data scientists are very open to collaborations with other individuals.

One possible reason for this disparity between the clustering coefficient and the high degree is that superstars within the network are part of multiple clusters, leading to many collaborations per scientist. These superstar scientists act as bridges between different clusters or communities.

This theory is further highlighted by the density of the graph, suggesting that only approximately 1.39% of all possible collaborations have occurred between data scientists. The network is relatively sparse, with most collaborations pushed by a select handful of popular data scientists.

Finally, there are 69 connected components, suggesting a wide range of distinct clusters/communities of data scientists who collaborate only with each other.

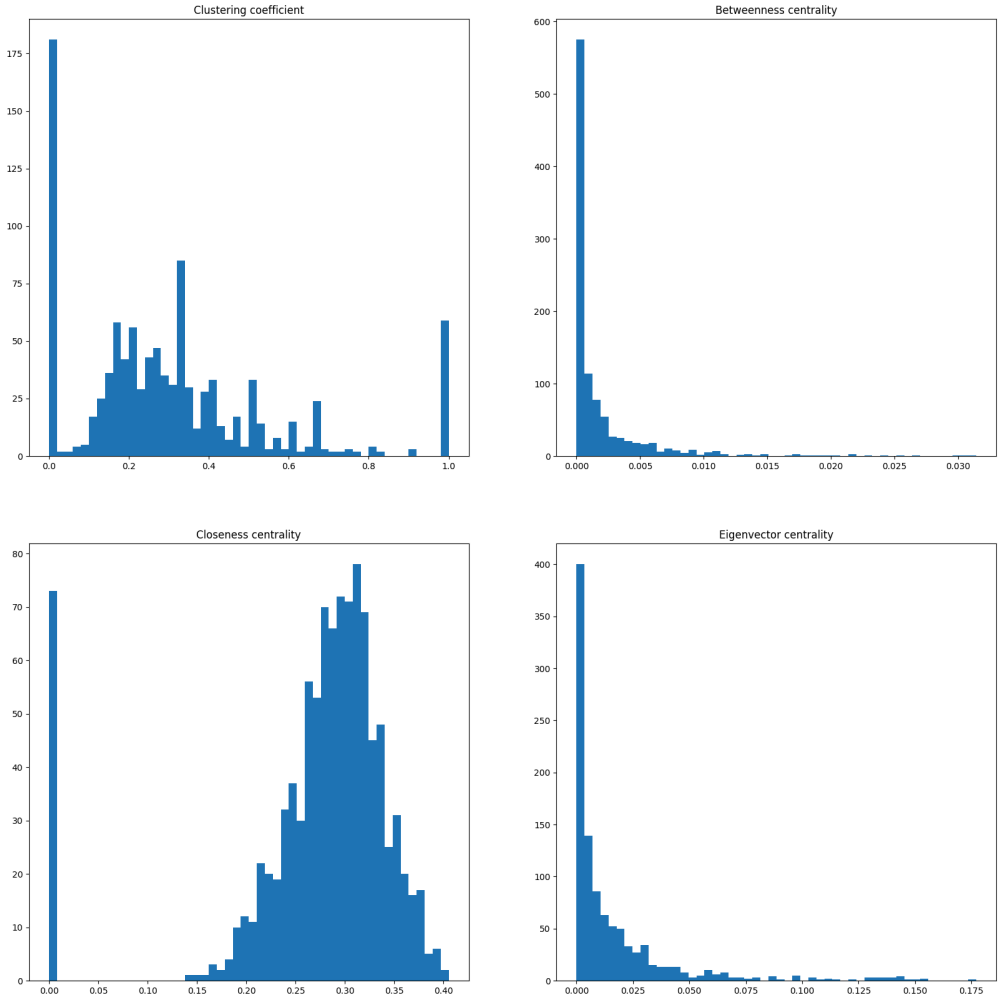
Analysing each connected component individually,

Size of largest connected component	955
Number of edges in largest connected component	7319
Average clustering coefficient of largest connected component	0.3233893497460729
Diameter of largest connected component	10
Average shortest path length of largest connected component	3.269421669026529

The size of the largest connected component and the number of edges in the largest connected component(LLC) tell us that most data scientists are connected in a single large component. Most of the collaborations would occur within this LCC, and all other components will likely be much smaller or isolated nodes.

Despite a large number of nodes, this component's diameter is only 10, suggesting that any two data scientists are separated by at most 10 collaboration hops. Furthermore, the average shortest path length is only approximately 3.27, meaning that on average, any two data scientists are within 3 to 4 collaboration hops. This backs up the theory of super scientists acting as bridges connecting to most other scientists.

Finally, while the LCC's clustering coefficient is slightly higher than the overall network's, it suggests that data scientists within the LCC are more open to collaboration than those outside, which would make sense as most of the remaining components are isolated nodes.

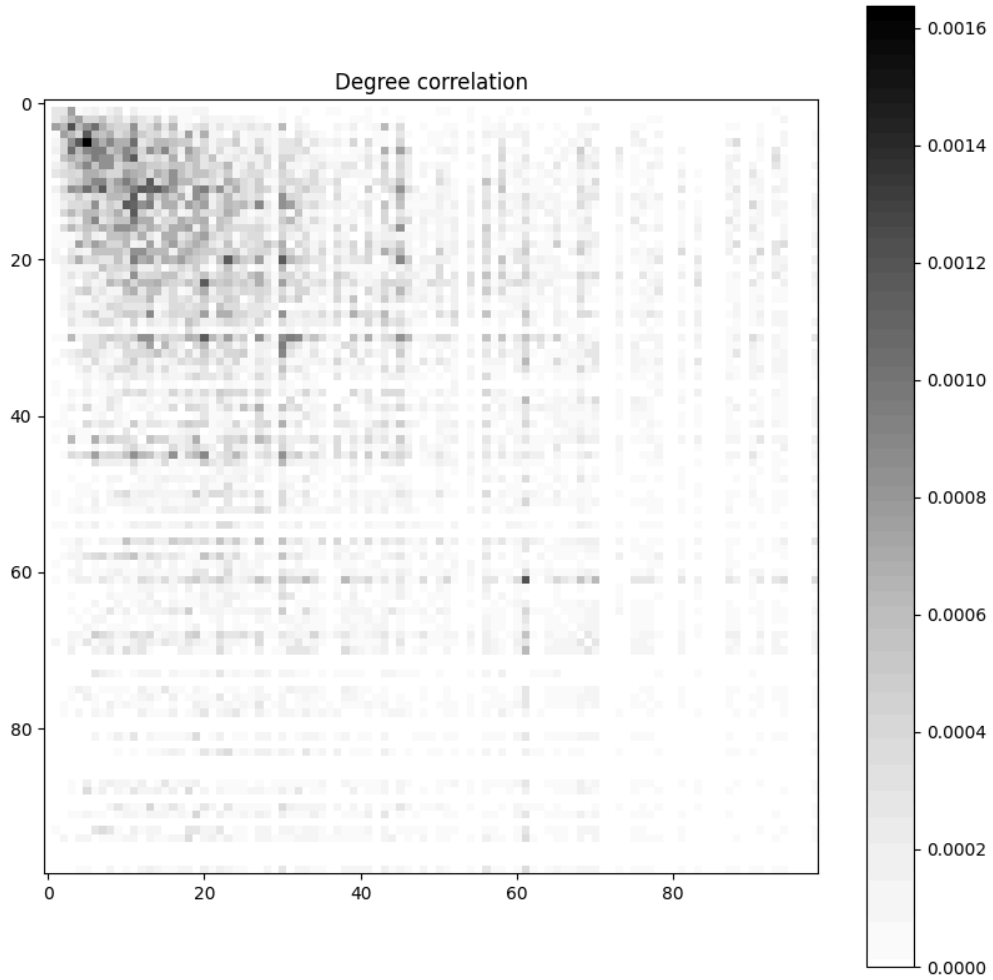


The clustering coefficient histogram has two prominent peaks, one peak close to 0 and another, smaller peak close to 1. This suggests that quite a few data scientists are isolated, and quite a few (to a lesser extent) tight-knit communities exist, most likely those from the same university or research group.

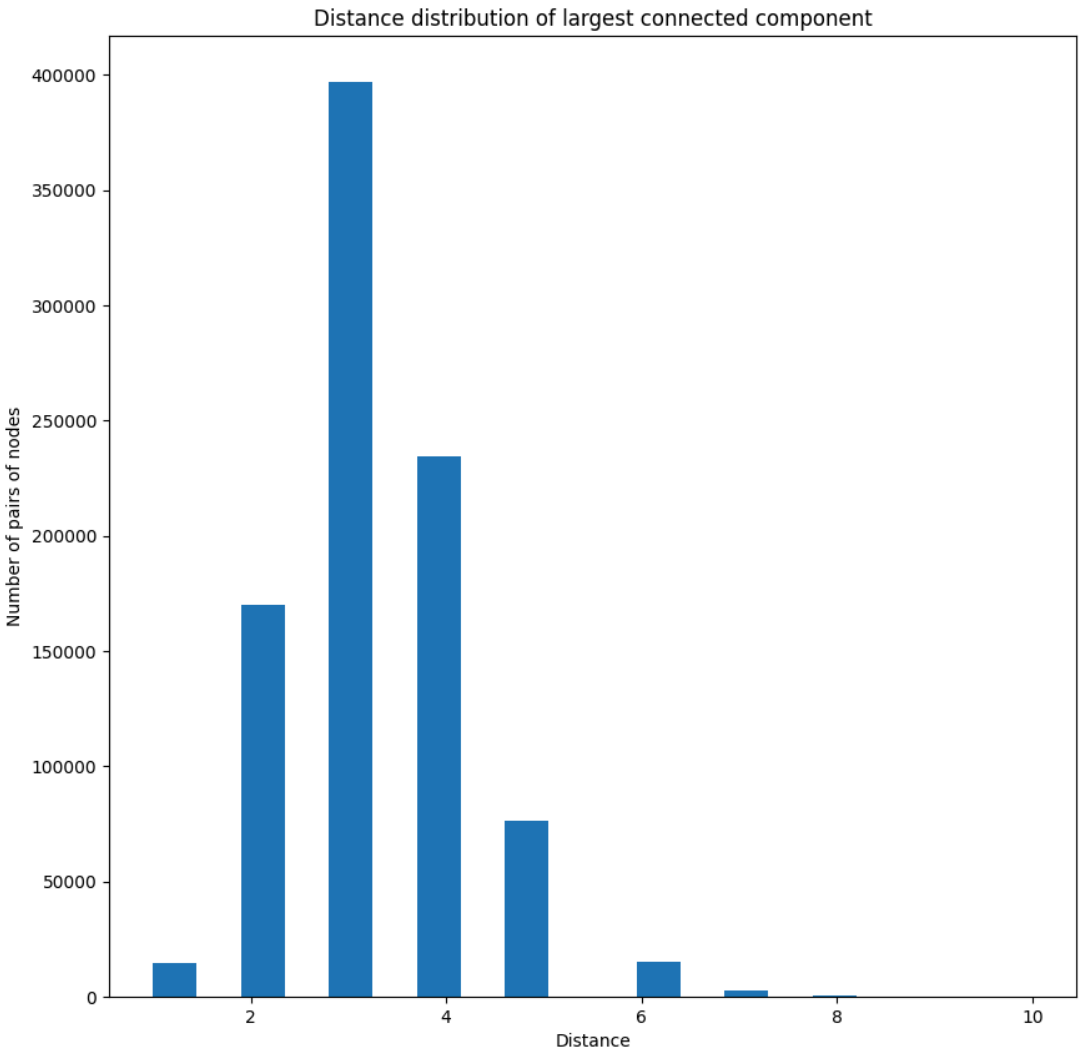
The betweenness centrality histograms show a skewed distribution that skews strongly to the left, which supports the theory of super scientists acting as hubs bridging many parts of the network. Thus, most nodes will never be part of a path between 2 nodes, as most paths are through these superstar nodes.

The closeness graph shows that a relatively large number of nodes have a closeness of 0, meaning they are isolated. At the same time, a handful of nodes have a closeness of over 0.35, indicating that some data scientists are closely connected to many others in the network.

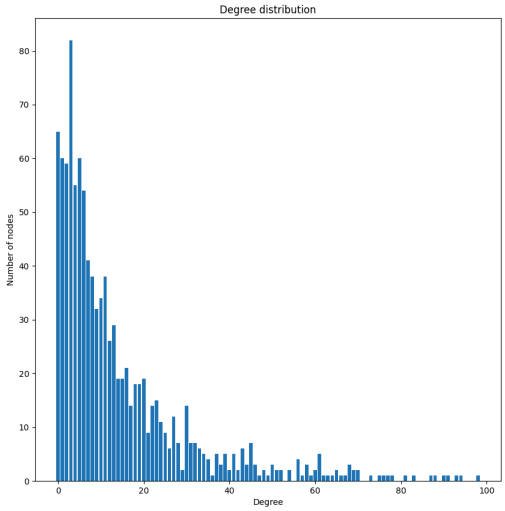
The eigenvector distribution is similar to the betweenness distribution, once again suggesting the existence of superstar data scientists in the tail-off at the right end of the histogram. By now, it is obvious that the network has scale-free properties, where most nodes have few connections while a few nodes (hubs) have very high centrality.

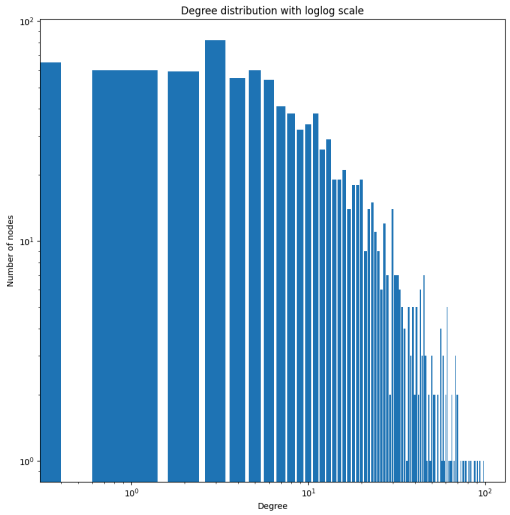


The degree correlation plot suggests that the network is generally an assortative one, as similar-degree nodes are more likely to connect with other similar-degree nodes. Successful data scientists prefer to collaborate with peers who achieve similar levels of success (in terms of the number of collaborations), possibly because they are more likely to contribute equally to the work. It could also be due to reputation, as more reputable data scientists naturally attract other reputable data scientists to add credibility to a project.



Most nodes are only 2-4 hops away, suggesting that the network has a small-world property as mentioned previously.

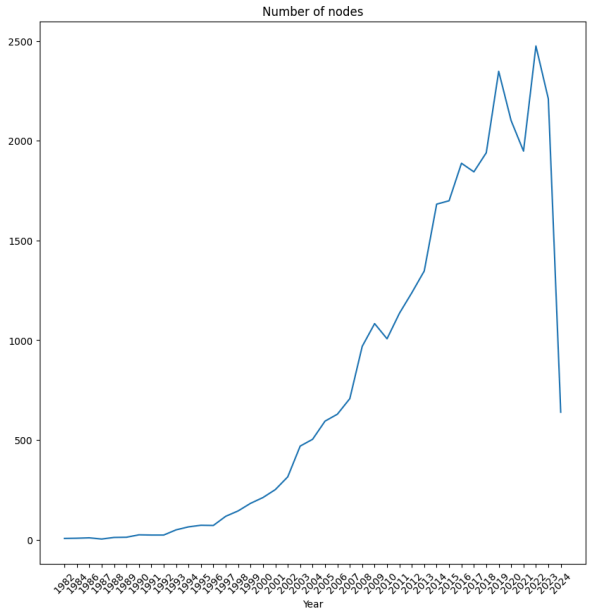


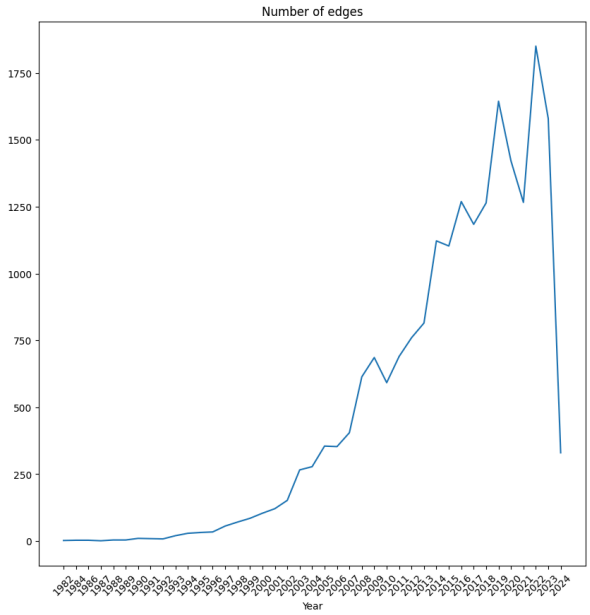


The degree distribution is significantly skewed to the left, with a handful of hubs having a degree close to 100 while most nodes have a degree less than 10. When plotted using a log-log plot, it shows a fairly linear line, hence the network follows a power-law distribution, with a few superstar data scientists and many poorly connected nodes.

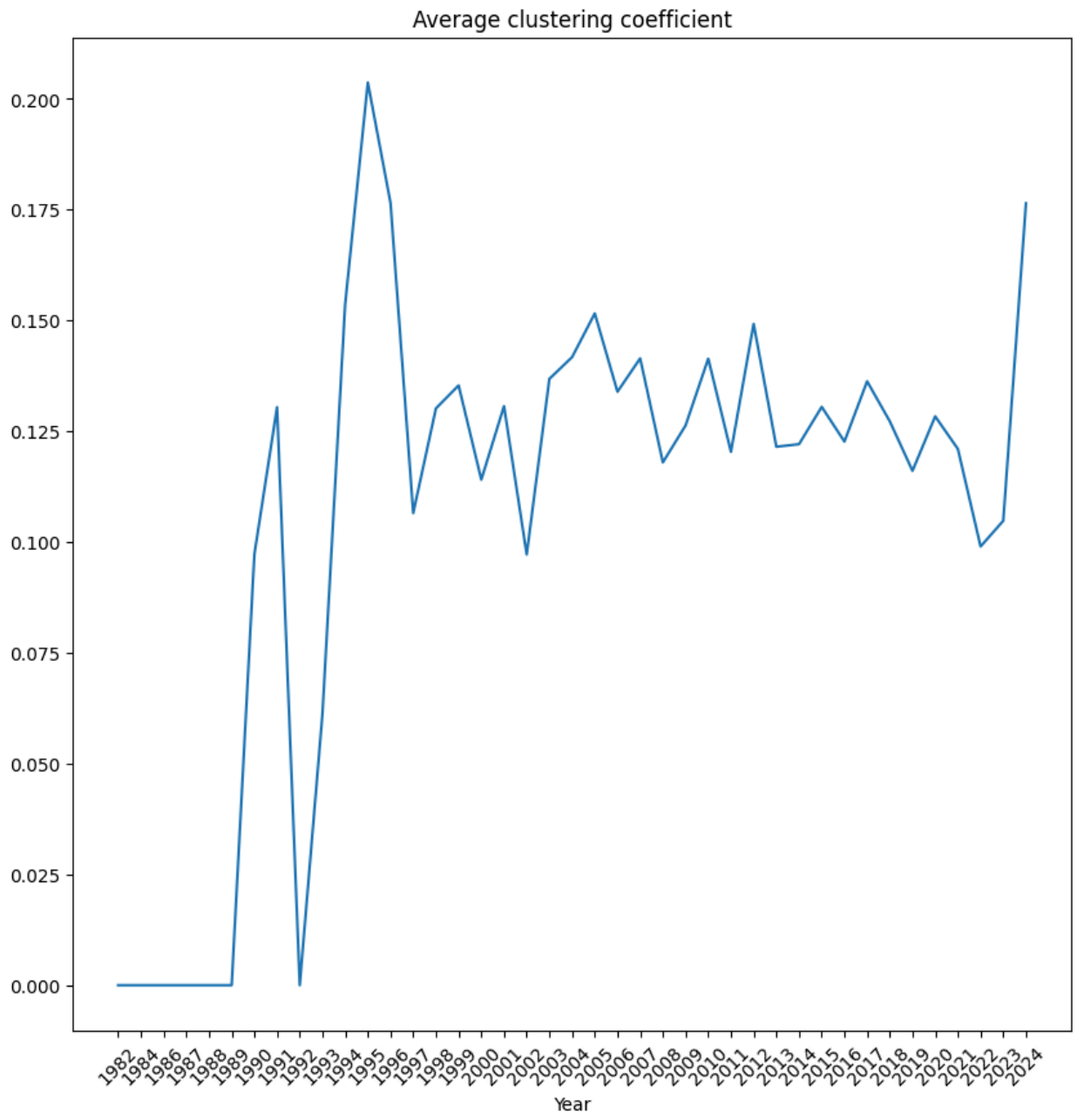
Question 2: Properties change of yearly collaboration network

For each year, we plot the collaborations between data scientists in that particular year.

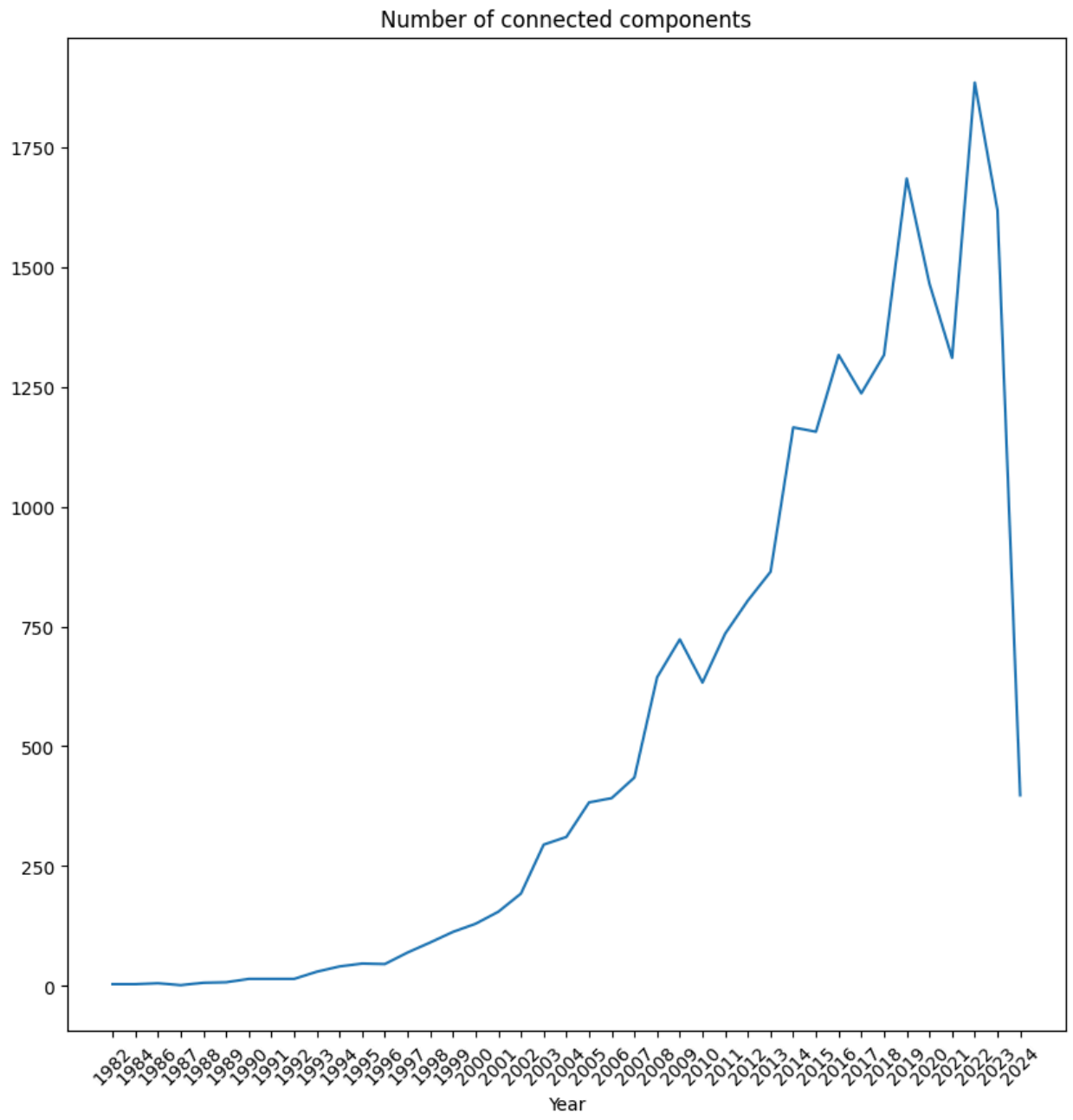




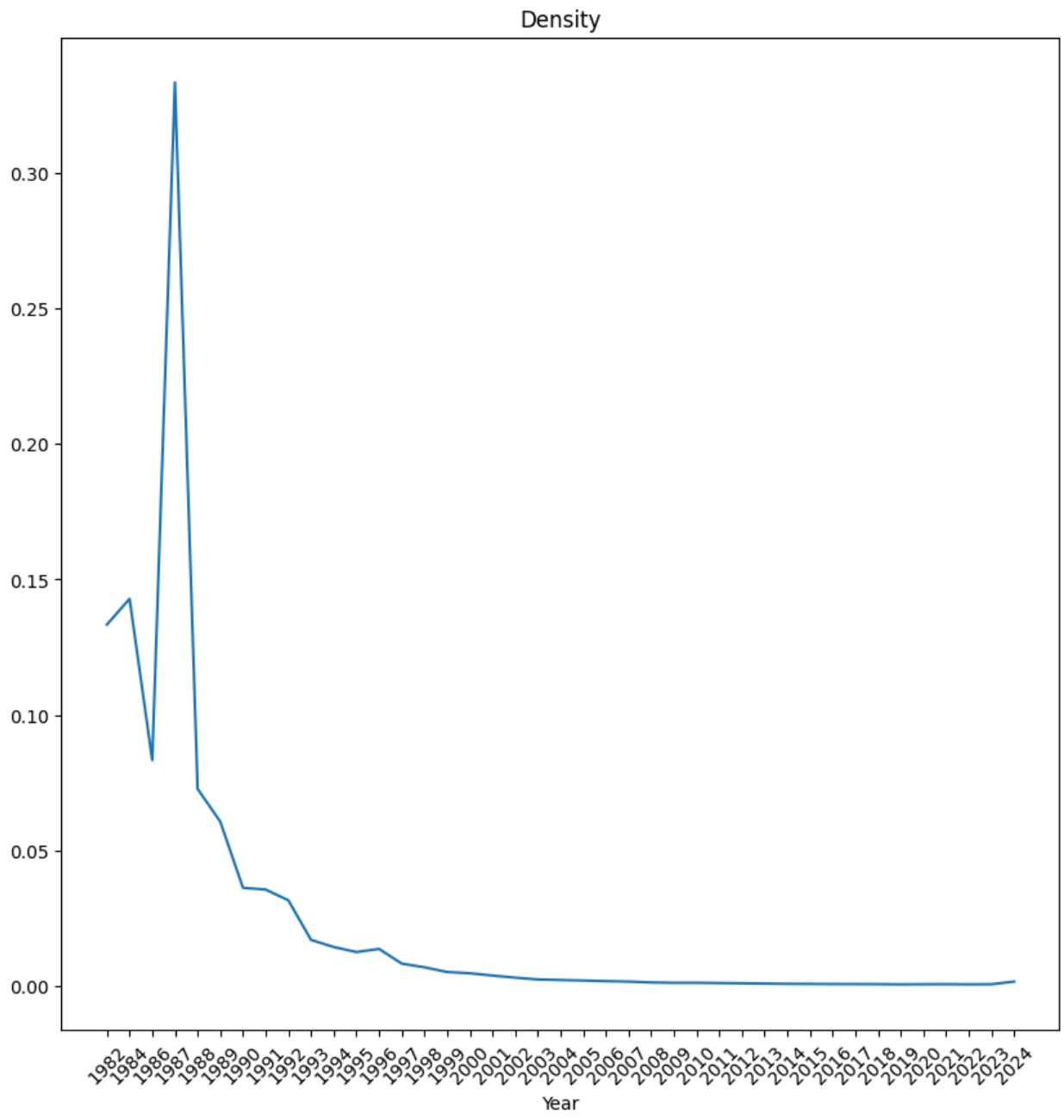
Over the years, the number of data scientists has increased at an increasing rate as data science has become a more popular field, especially in the past decade. Naturally, the number of edges also increases proportionately as more data scientists are available to collaborate with. The discrepancy with 2024 occurs as at the time of this report, we have only entered four months into 2024. This discrepancy will repeat for most graphs and be ignored in future analyses.



The average clustering coefficient increased from 1982 to 2000, before fluctuating wildly at approximately 0.125. There exist 2 prominent peaks in 1991 and 1995, during which only a handful of data scientists were publishing. Hence, any collaborations that occurred would have significantly affected the clustering coefficient, resulting in an unnatural spike. We can also observe the impact of COVID-19 on collaborations between data scientists as there is a fairly sharp drop from 2020 to 2022 before increasing again, likely due to quarantine limiting collaboration opportunities. Finally, the fluctuations can be attributed to superstar data scientists responsible for most of the average clustering coefficient. As such, the absence of any particular superstar for the year would cause fluctuations.

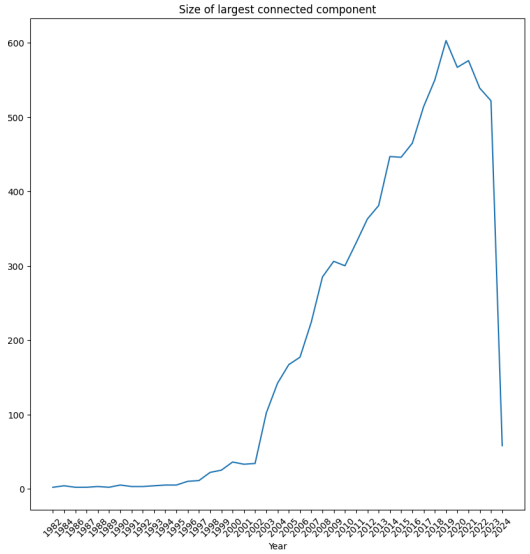


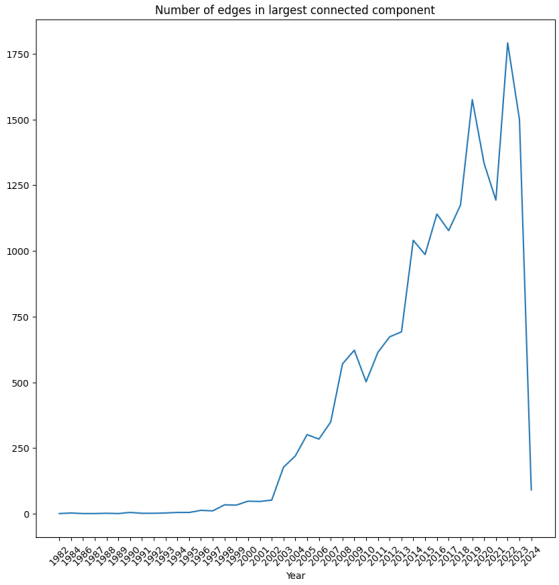
The number of connected components steadily increases from year to year, correlating with the increase in the number of data scientists in general. This indicates an increase in the emergence of new data scientists who most likely recently graduated, are inexperienced, and lack the connections that the older data scientists possess, which limits their collaboration opportunities. Once again, we observe an interesting drop in 2020, when the pandemic broke, disrupting research and lowering the number of publications made until WFH measures became a norm from 2021 onwards.

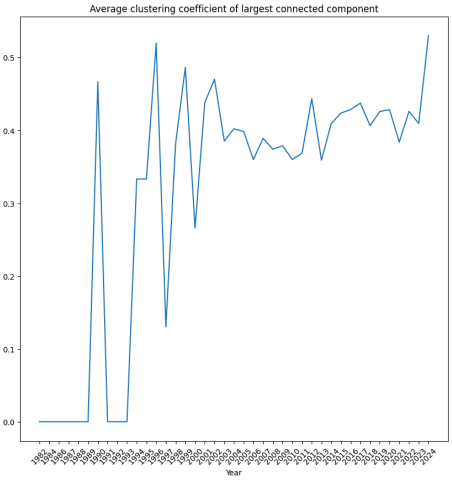


Density initially starts relatively high, likely due to the small size of the network at the beginning. It is much easier to achieve high density with smaller sizes as fewer connections are required to connect every node. However, there has been a sharp decline since the 1990s. This can be explained by the boom in the number of data scientists in our previous graphs, causing rapid growth in the number of nodes without a proportionate increase in the number of edges (collaborations), as once again these new data scientists lack the industry connections that come with experience. Eventually, the density stabilises from the 2000s onwards, still decreasing but at a slower rate. We suspect that technological advancement, especially in the early 2000s, made communication and collaboration much more accessible, which helped the number of collaborations keep pace with the number of new data scientists entering the field.

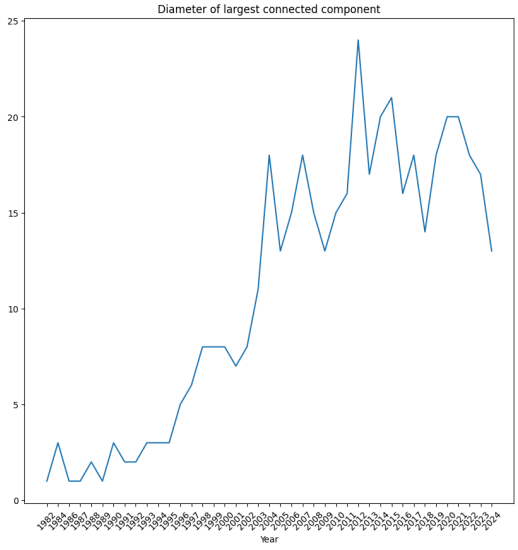
The initial batch of data scientists also starts fostering close-knit communities that promote the integration of new data scientists by connecting them to other established peers for collaborative opportunities.

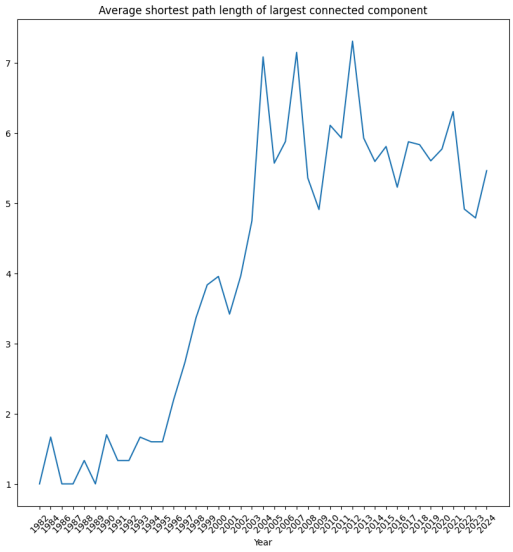




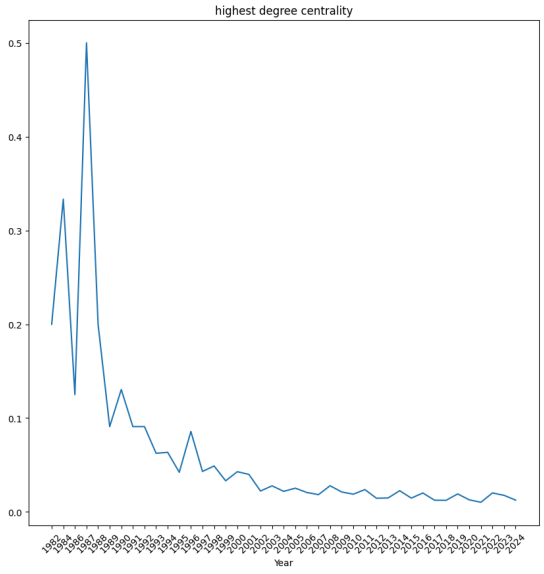


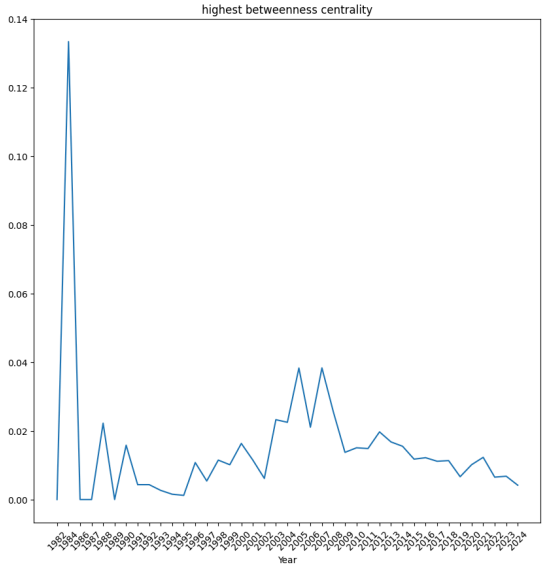
This graph supports the theory that the pioneer generation set up a supporting network structure to help new data scientists. It shows relatively stable growth in the size of LLC until the 2000s when it suddenly spiked immensely and continued for the rest of the years.

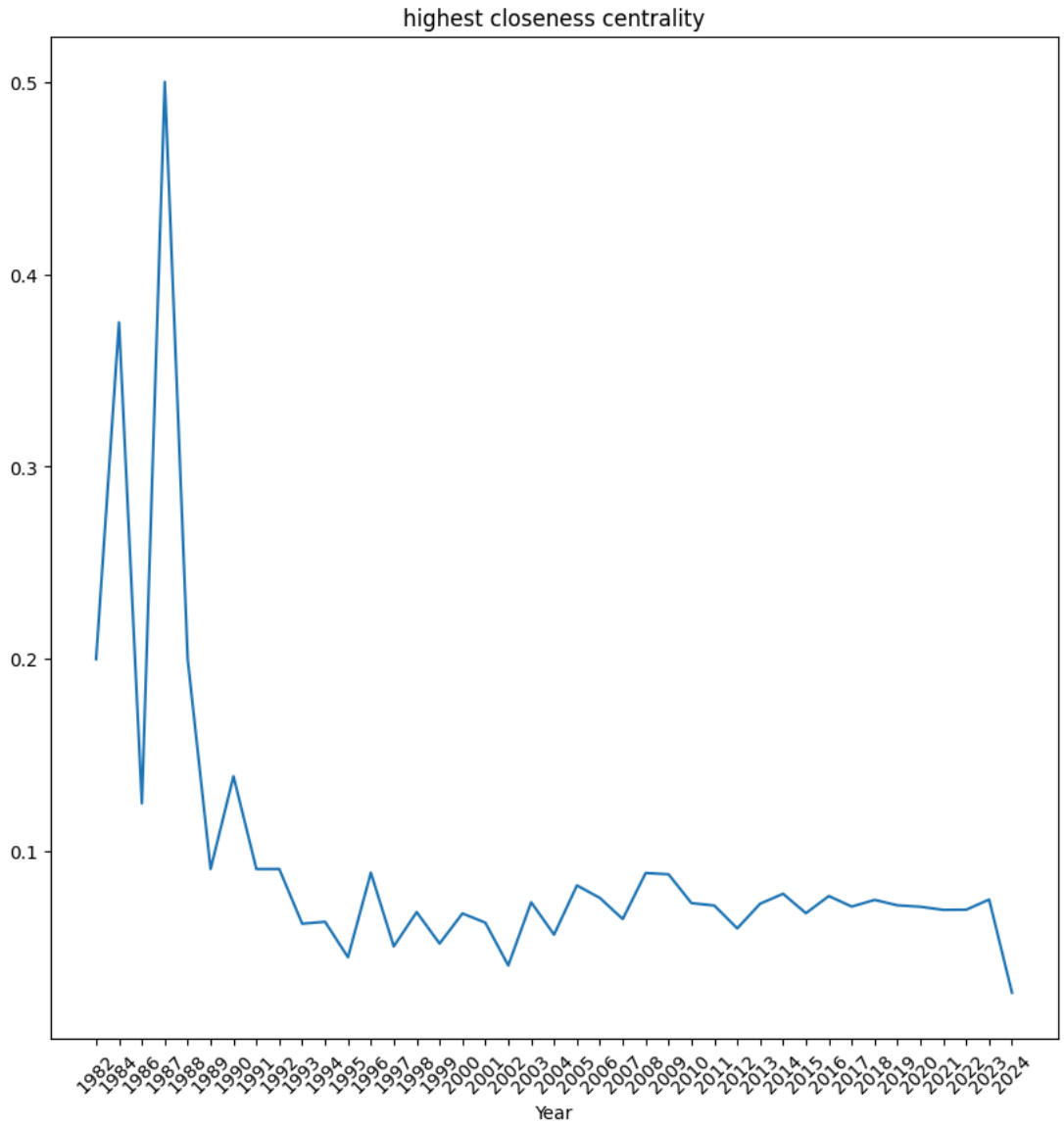




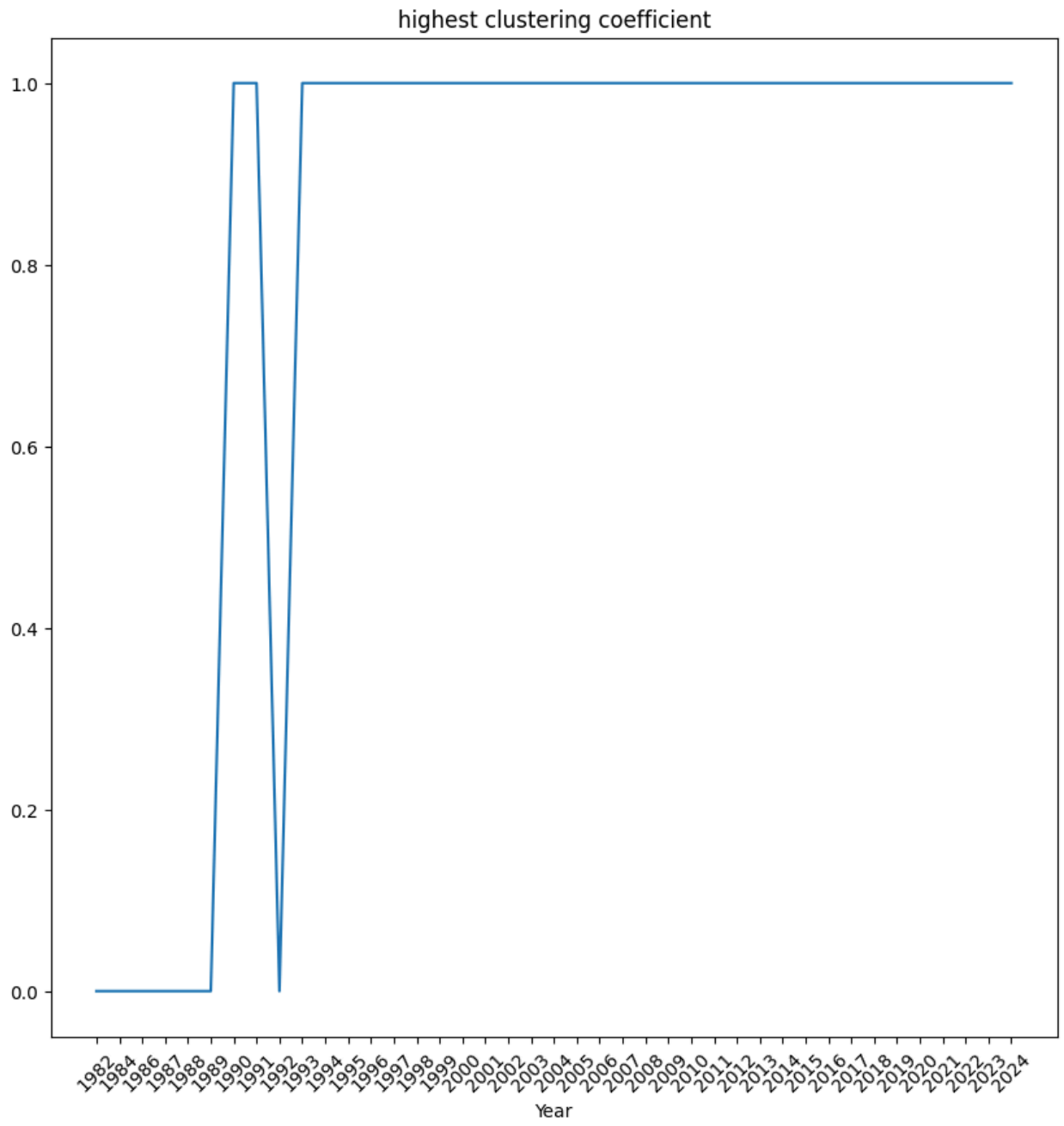
Both diameter and average shortest path length increase steadily in the early years, likely due to the increase in the number of data scientists, until they start fluctuating heavily with many peaks and troughs in the 2000s. We have already deduced that there are superstar data scientists who act as hubs. These peaks could be attributed to some of these hubs taking a break from publications, which can massively increase the average path length between 2 nodes.



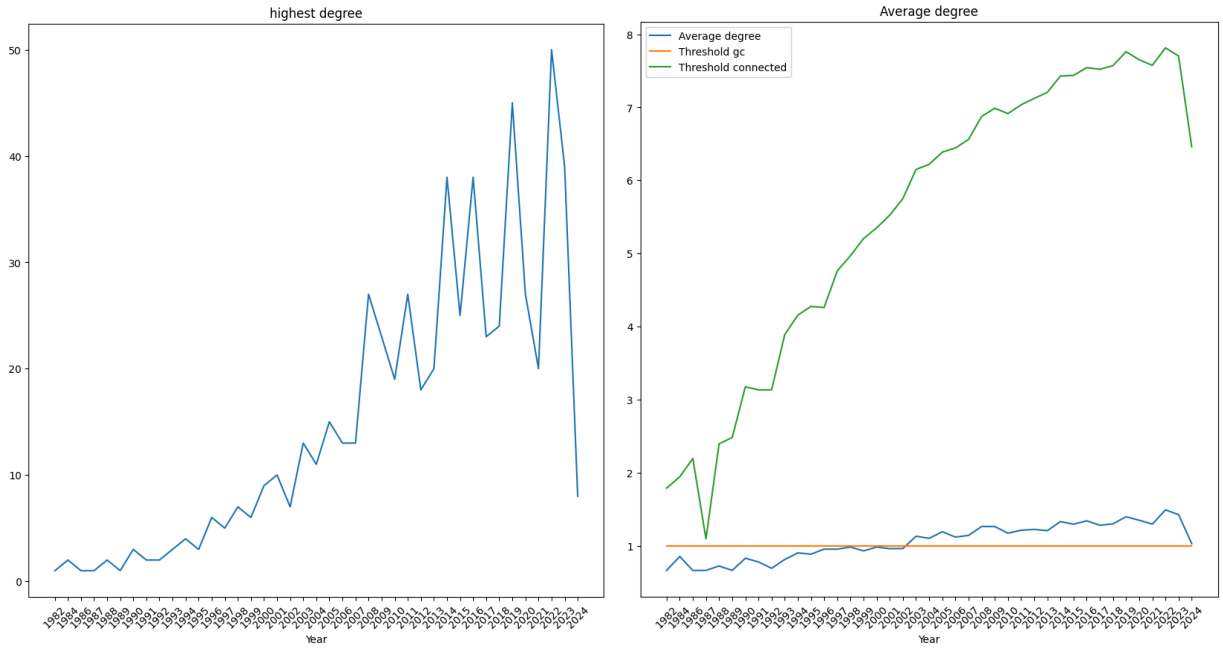




All three centrality graphs exhibit the same pattern, exhibiting a downward trend following an early peak. This suggests that while the early years were dominated by a few superstar scientists, the data science field has grown, causing collaborations to become more distributed amongst a larger group of data scientists. Initial hubs have been bypassed in later years by new connections formed amongst other data scientists, causing a decrease in highest betweenness centrality.

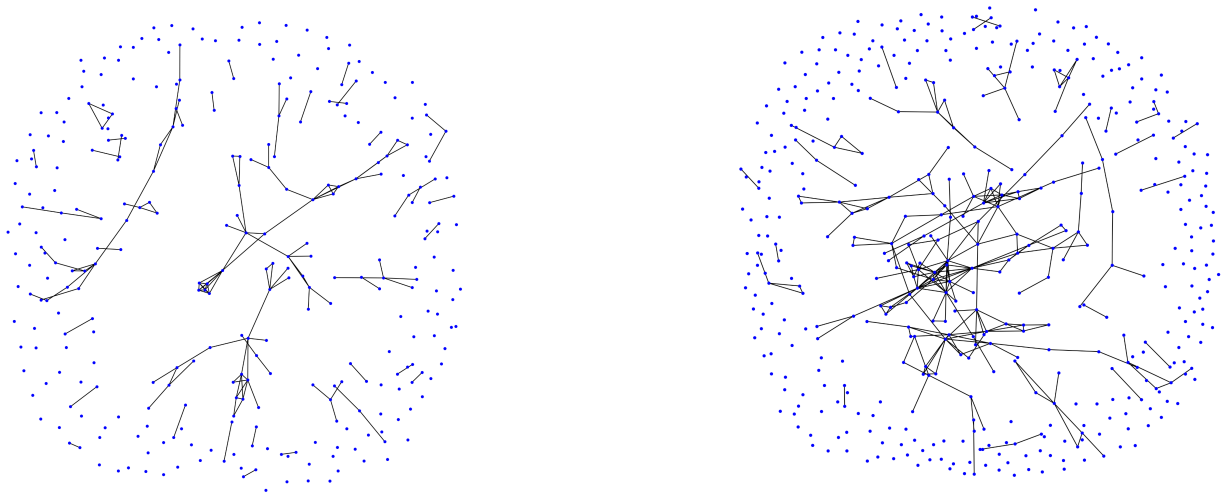


However, the stable and extremely high clustering coefficient indicates that well-established communities (research groups, schools) continue to collaborate tightly.



Finally, we observe the highest and average degree of the graph for each year. The highest degree graph shows the number of collaborations of the most connected individual data scientists. The overall increasing trend suggests that superstar data scientists garner more collaborations with each passing year, and any troughs in the graph can be explained by their lack of publications for that particular year.

The orange line represents the threshold for generating a giant connected component, while the green line represents the threshold for generating a connected graph. The graph shows that the average degree exceeded the threshold needed to generate a giant connected graph in 2003.

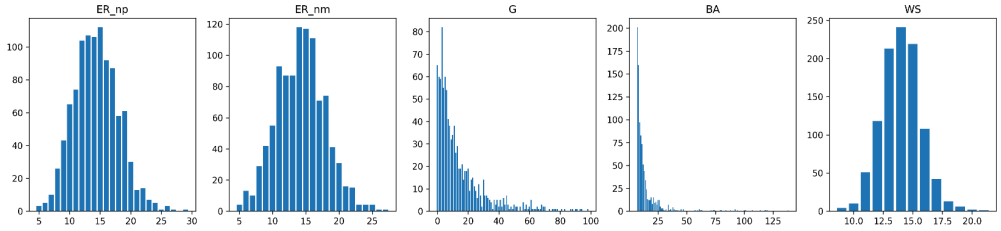


Graph of 2002(left) and 2003(right)

Question 3: Random graph comparison

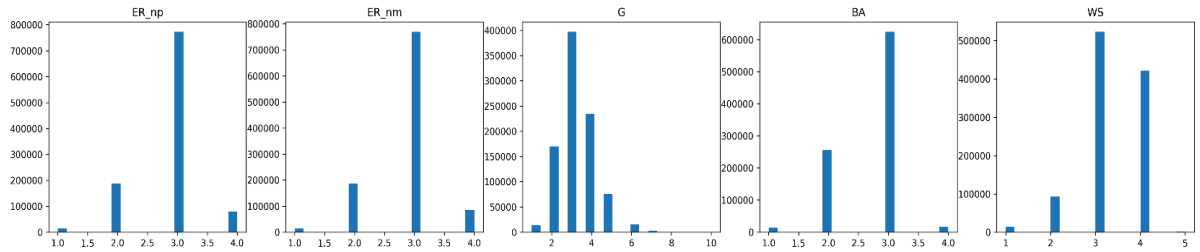
We have decided to randomly generate four graphs, each using a different random generation method, and compare them with our original graph, G.

ER_np	Erdős-Rényi graph (Probability-based)	Each node has a P probability of forming an edge with another node
ER_nm	Erdős-Rényi graph (Fixed number of edges)	The random graph will have M edges out of all possible edges
BA	Barabási-Albert (BA) graph	New nodes are more likely to connect to highly connected nodes
WS	Watts-Strogatz small-world graph	High clustering coefficient with short average path lengths



The figure above represents the degree distribution for each graph.

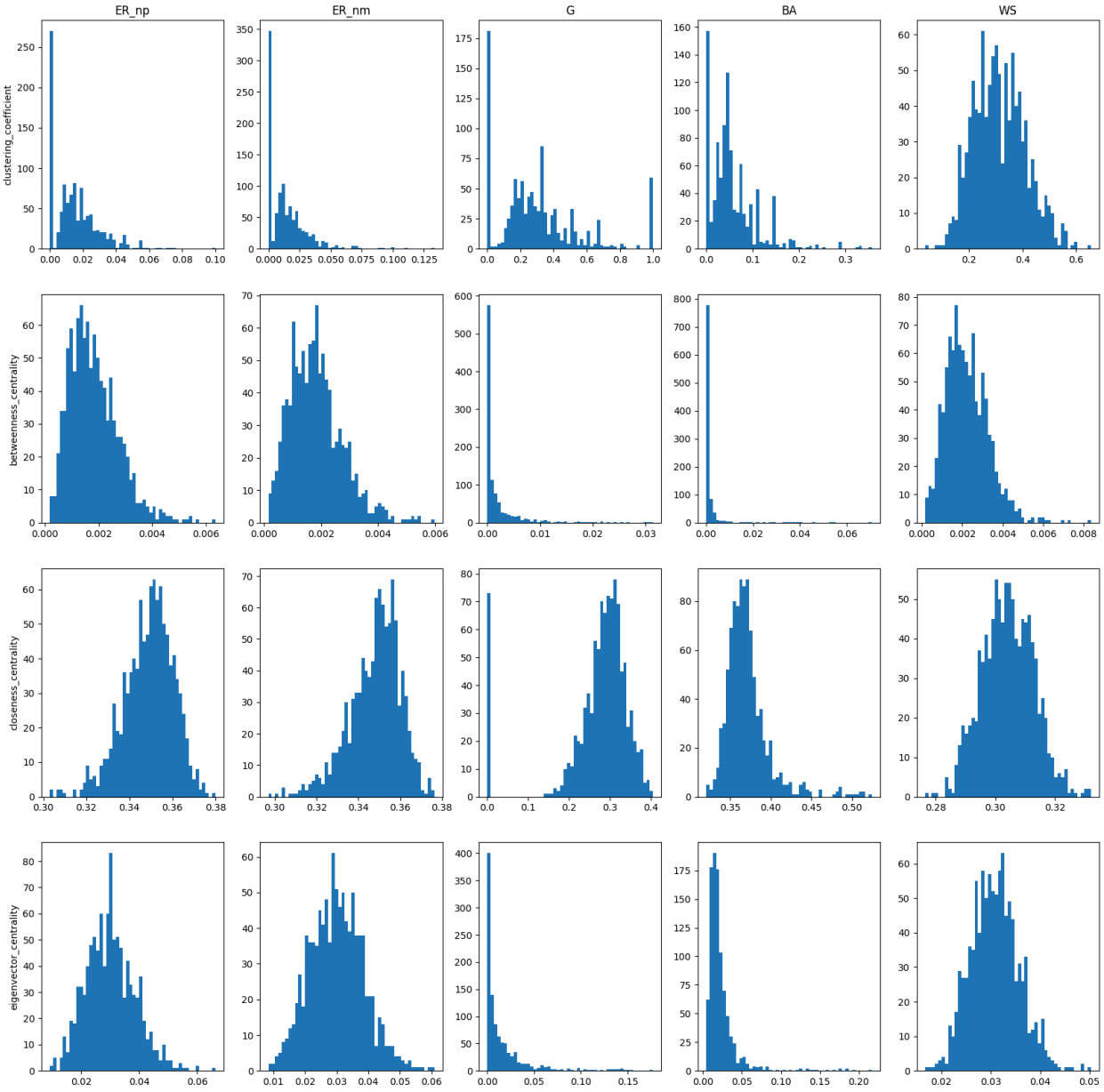
G	As mentioned previously, follows a scale-free, power-law distribution
ER_np	Approximate binomial distribution, symmetric around the average degree due to randomly connecting nodes with a fixed probability
ER_nm	Similar to ER_np as its edges are also assigned probabilistically (exactly m edges amongst n nodes), binomial distribution
BA	The most similar to the original graph as it also follows a power-law distribution, more strongly skewed to the left but the general shape is identical
WS	The histogram is the least similar to the original graph. It is perfectly uniform, with a small range centring around the average degree.



This figure represents the distance distribution for each graph's LCC.

G	As mentioned previously, due to the superstar data scientists, there exist hub nodes that significantly reduce the overall path length between many pairs of nodes
ER_np	Due to the random generation of edges with p probability, the distance between nodes for most will peak at the average path, with few pairs further or closer away.
ER_nm	Similar to ER_np as its edges are also assigned probabilistically (exactly m edges amongst n nodes), and in this case has identical graph shapes as ER_np
BA	Also has hubs to shorten distances between nodes. However, the scale-free nature of the network means that some nodes can be very isolated and much further away, although the average distance would still be significantly lower
WS	Due to its small-world property, its LCC will have a high clustering coefficient and naturally short average path lengths. It exhibits a fairly similar graph shape to BA.

For average distance, most of the graphs exhibit similar average distances, all peaking at a distance of 3. However, the original graph has the broadest range of possible distances, with the furthest distance between 2 nodes at 7, while the next highest is 4.



Clustering Coefficient

G	As mentioned previously, the presence of superstars
ER_np	Least similar to the original graph. Extremely low clustering as edges are placed randomly: Chances of multiple nodes clustering together are lower than P

ER_nm	Identical to ER_np for the same reasons
BA	New nodes tend to form edges with other high-edge nodes, which lowers the clustering coefficient as those nodes are less likely to form edges with neighbour nodes.
WS	It has a high clustering coefficient due to the regular lattice structure in its initial construction. However, unlike the original graph, it has very few unclustered nodes and very few nodes on the other end of the graph. For this reason, it can be considered the most different from the original graph.

Betweenness Centrality

G	As mentioned previously, the presence of superstar scientists means that most routes between nodes go through them instead of others, leading to shallow values for most nodes.
ER_np	Because the edges are randomly spread around, there is no strong presence of central nodes that dominate all paths within the network and the betweenness is distributed relatively evenly
ER_nm	Identical to ER_np for the same reasons
BA	Very Similar to the original graph, its scale-free nature means that there exists a few hubs that lie on the majority of the paths between nodes
WS	Seems to be similar to ER graphs, possibly due to a low rewiring probability (0.24)

Closeness Centrality

G	Peaks at a lower value due to its scale-free property. Has a peak at value 0 as well due to the presence of isolated data scientists that have never collaborated with anyone
ER_np	Randomly distributed edges mean that the closeness of most nodes would be similar, and this is reflected in the graph as the distribution is narrowly centred
ER_nm	Identical to ER_np for the same reasons
BA	The distribution peaks at lower values due to its scale-free network, in which a few hub nodes have significantly higher closeness.
WS	They are relatively uniform due to the small-world property of such graphs, which means that most nodes will be relatively close to each other.

Eigenvector Centrality

G	The distribution is strongly skewed to the left, indicating a hierarchy of node influence. Once again, this is due to superstar data scientists acting as influential, central nodes.
ER_np	The eigenvector values are spread out fairly evenly, as randomly placed edges would not often create nodes with high centrality.
ER_nm	Identical to ER_np for the same reasons
BA	It is identical to the original graph, where the distribution is strongly skewed to the left. The preference for nodes to connect to other nodes with high degrees leads to a graph with a few nodes having substantially higher centrality scores than the rest
WS	Like the ER graphs, it has a uniformly distributed eigenvector with very few nodes on both ends. This is achieved through the initial lattice structure and the induced randomness later on during the graph's construction.

Question 4: Graph compression

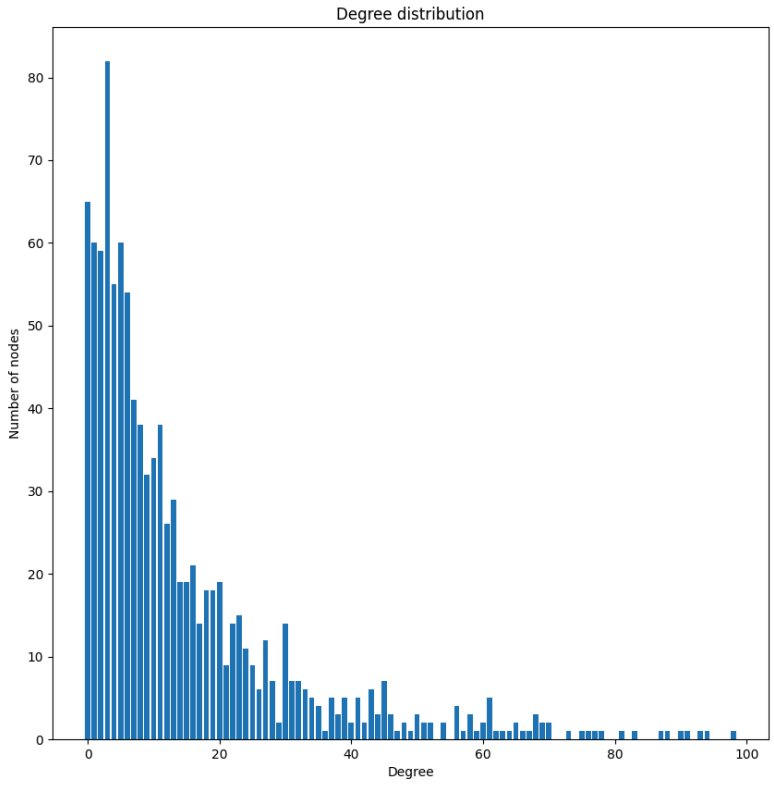
In this part, we should transform the original graph into a robust graph so that low-degree nodes will not be influenced by high-degree nodes. The transformed graph has the characteristics below:

a. Smaller giant components result in a more significant number of isolates.
b. The degree of all nodes should not exceed a collaboration cutoff k_max.
c. Keep the diversity of individuals (country, expertise and institutions) in the transformed graph.

To preserve the diversity of individuals, we should avoid deleting nodes as much as possible so that the information can be preserved well.

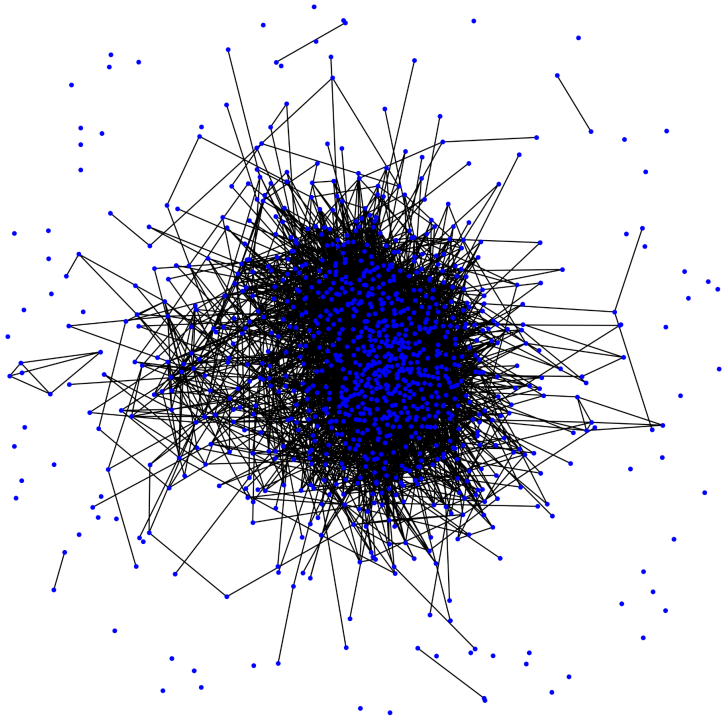
We will split this problem into four steps:

Step 1: k_max setting

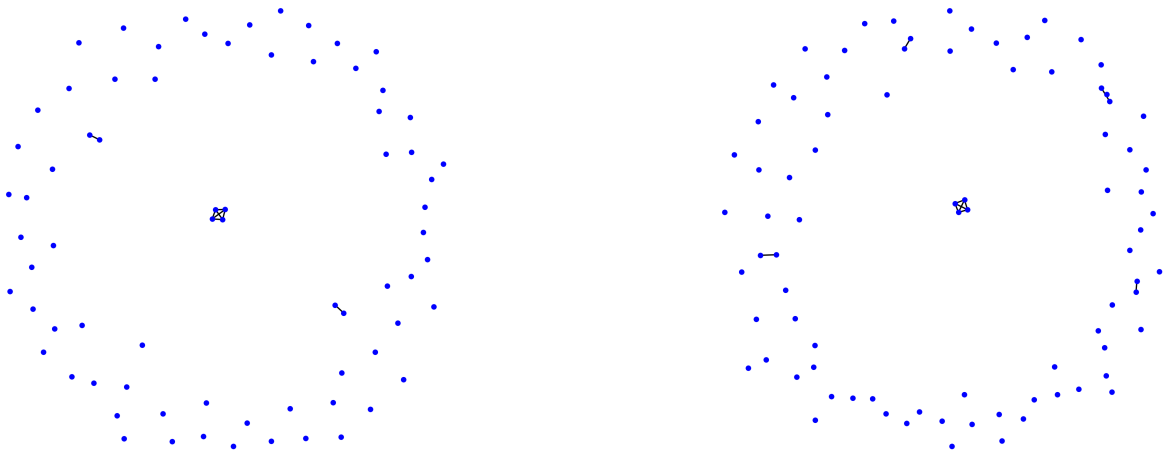


According to the degree distribution, k_max=45 is a good cutoff as the number of degrees has a local minimum of around 45 and the right side only contains a few fraction of nodes.

As the graph is assortative, we can reserve the edge based on the degree of the neighbour linked with that edge. Meanwhile, they should not link together to avoid mutual affection between high-degree nodes. After cutting edges, the graph shows below:



Number of edges after edge cut: 6127
Number of connected components: 77



Small components in the graph change from left to right.

The degree distribution is listed in this dictionary. We can find all degrees below 45.

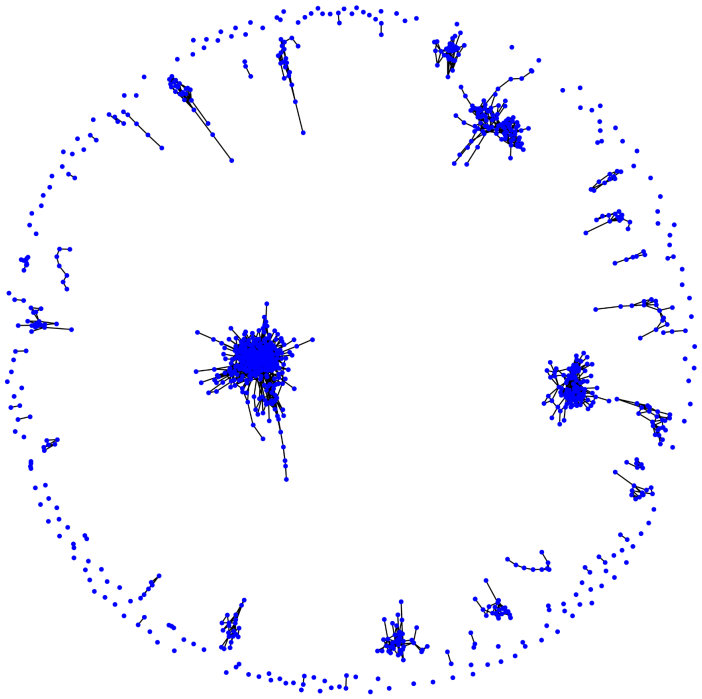
{0: 71,1: 61,2: 71,3: 80,4: 66,5: 65,6: 44,7: 43,8: 32,
9: 36,10: 34,11: 41,12: 17,13: 24,14: 22,15: 28,16: 17,
17: 13,18: 17,19: 18,20: 16,21: 13,22: 15,23: 11,24: 6,
25: 8, 26: 13,27: 8,28: 13,29: 12, 30: 7, 31: 11, 32: 11,
33: 11, 34: 9, 35: 7, 36: 4, 37: 4, 38: 11, 39: 9, 40: 5,
41: 2, 42: 5, 43: 8, 44: 4, 45: 5}

Step 2: Cut edges based on country

In a collaboration network, if a person has more than two countries, it is highly probable that it is a cut node, so we can randomly choose one of its countries to belong to.

Collaboration has a strong nation-split property: A cooperator from another country should have a small influence on a local research network. So, we can just remove all edges of transnational cooperation.

After removing edges, the graph is shown below:



Number of edges after cut: 2575 Number of connected components after cut: 219

It is divided into many smaller connected components. Based on the nodes in each component, we can divide them into three groups.

Small: less than 11;
Medium: between 11 and 50;
Large: more than 50

Step 3: Process medium components

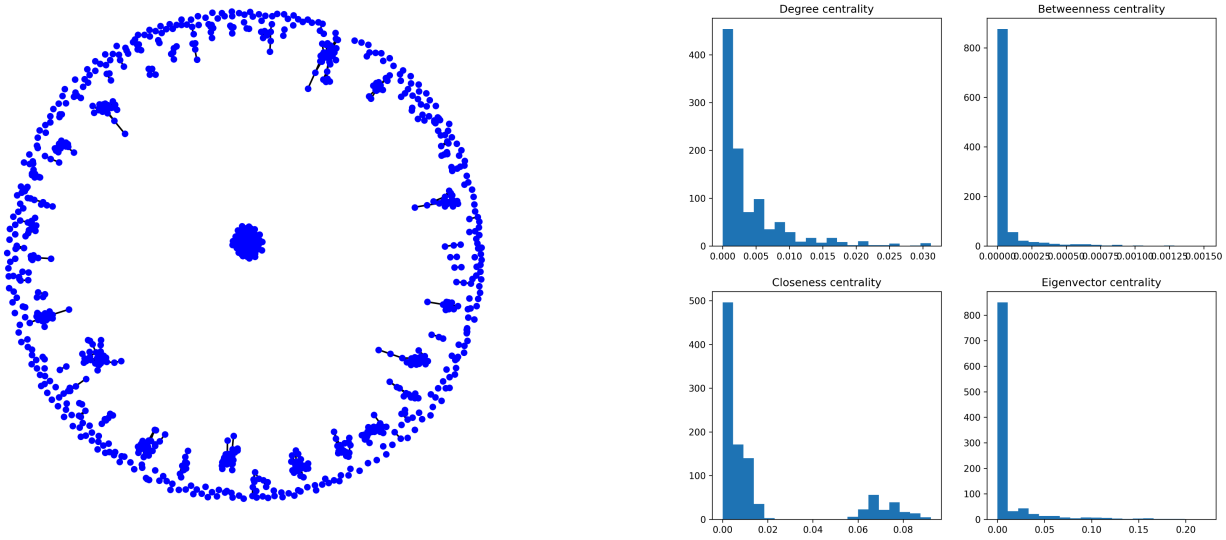
We analyse each medium component and remove the node with both high closeness and betweenness centrality. This can remove a centralised node and reduce centrality.
In our experiment, we only remove one node. The number of edges after removal is 2568 and After removal, the number of connected components is 221, which is an increase of two from the result of step 2.

Step 4: Process large components

In this part, we use different community detection algorithms to find the partition of each large component. We used the Girvan-Newman method, Louvain method and K-core decomposition method.

a. Girvan-Newman method

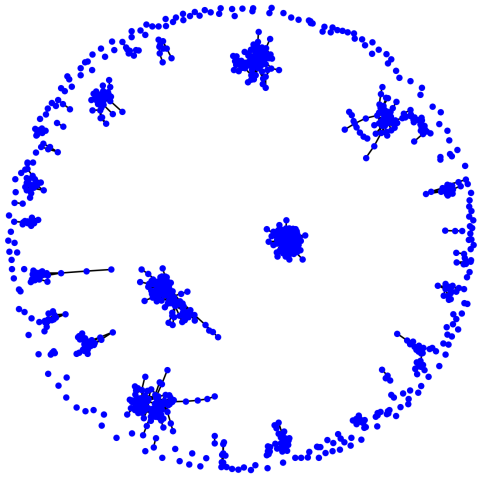
The GN method calculates the betweenness coefficient of each edge and removes the edges with the highest score to split the graph into two subgraphs. We can partition the graph many times. But when processing the GN method, we found it split many isolated nodes, so we added a rewire for isolated nodes: if they have higher connected links with a community in the original graph, then the average degree of the community times 0.9 is a tolerant discount.

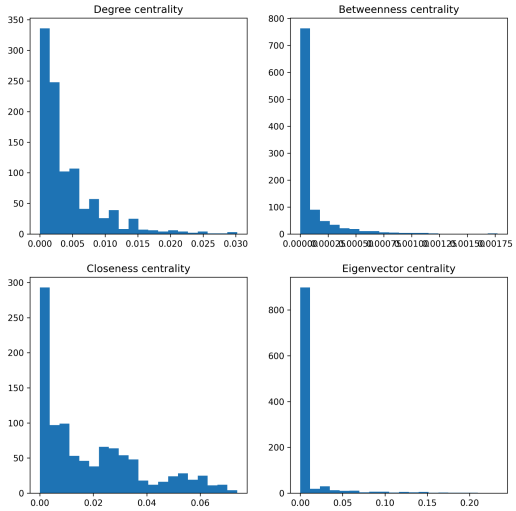


The left-hand side shows the graph and the right side shows the centrality properties of the graph. More properties can be found in ResultGraphCompressionGirvan_Newman.
We found it decreased the centrality compared to the original graph.
The most significant component is with 182 nodes, and the number of isolated nodes is 263.
Total number of connected components is 355.

b. Louvain method

The Louvain method detects the community by optimisation of modularity.

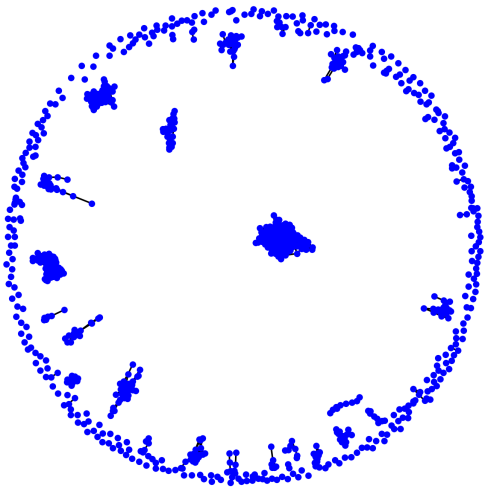


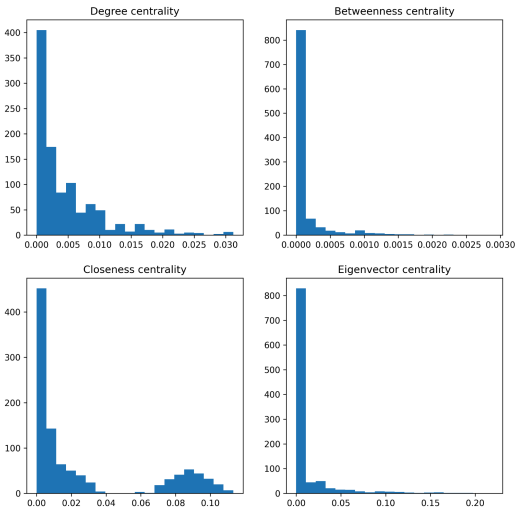


The left-hand side shows the graph, and the right side shows the centrality properties of the graph. And more properties can be found in Result\GraphCompression\Louvain.
We found it decreased the centrality compared to the original graph.
The most significant component is with 144 nodes, and the number of isolated nodes is 166.
Total number of connected components is 226.
Compared to the GN method, it reserves more cooperation links for nodes far from the centre.

c. K-core decomposition method

The k-core decomposition method is used to find the maximal subgraph in which all nodes have a degree of at least K.





The left-hand side shows the graph, and the right side shows the centrality properties of the graph. More properties can be found in ResultGraphCompressionK-core.
We found it decreased the centrality compared to the original graph.
The most significant component is with 250 nodes, and the number of isolated nodes is 297.
Total number of connected components is 352.
It may only find a central core in the graph so that a large community is left.

Conclusion

After transformation through cutting high-degree node edges(a local feature of the graph), removing transnational cooperation(an external feature of nodes), removing high-centralized nodes (an internal feature of nodes), and detecting community(a global feature of the graph), we obtained a less centralised graph with smaller large connected component and more isolated nodes. At the same time, it limits the maximum degree and reserves the diversity of the original graph.