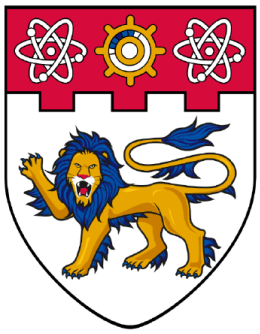


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC4022

Network Science Group Project

Can network science help us to understand research collaboration among data scientists over time and select a subset of them for various tasks?

Members and Roles:

Name, Matriculation No.	Role
Eugenia Poon U2222896D	Data Retrieval, Q1
Xing Kun 2023452E	Q2, Q3
Shourya Kuchhal U2123334A	Q4

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING NANYANG TECHNOLOGICAL UNIVERSITY

Table of Content

1 Data Retrieval

This section describes the end-to-end data acquisition and cleaning workflow designed to extract publication records of data scientists from DBLP, a prominent computer science bibliography website. The pipeline involves three major stages: data crawling, parsing, and data cleaning. The final outputs are standardized, deduplicated datasets of scientists and their respective publications.

1.1 Data Crawling

The crawling process begins with an Excel file, datascientists.xls, which contains DBLP profile URLs for a curated list of data scientists. These links are then programmatically accessed to extract a unique PID (personal identifier) that DBLP assigns to each author page.

```
match = re.search(r'pid/(.*).html', final_url)
```

However, it is important to note that not all entries in the Excel file conform to the standard DBLP PID format. While many DBLP profile URLs redirect to pages containing the PID pattern dblp.org/pid/{pid}.html, a subset of the URLs point to name-based profile pages (e.g., dblp.org/pers/hd/s/Smith:John), from which PIDs cannot be programmatically extracted using a uniform pattern. This limitation introduces a layer of incompleteness in the dataset, as shown in the following code snippet where unsuccessful PID resolutions are logged:

```
    match = re.search(r'pid/(.*).html', final_url)

    if match:
        pid = match.group(1).replace('/', '-')
        pids.append(pid)
        final_urls.append(final_url)
    else:
        pids.append('Error')
        final_urls.append('Error')
        errors_links.append(link)
```

Additionally, some URLs return HTTP 410 errors (indicating the page is no longer available) or exceed the request threshold, returning HTTP 429 ("Too Many Requests"). These are handled with retry logic and exponential backoff:

```
        if response.status_code == 429:
            print("Too many requests. Sleeping for 60 seconds...")
            time.sleep(60)
            continue
        elif response.status_code == 410:
            print(f"{link} is gone (410). Skipping.")
            response = None
            break
        elif response.status_code != 200:
            raise Exception(f"HTTP Error {response.status_code}")
        break  # success
```

Ultimately, only the links with valid PID extractions are retained. This results in a cleaned list of scientists with associated DBLP identifiers, saved for downstream use:

```
cleaned_df = cleaned_df[(cleaned_df['pid'] != 'Error') & (cleaned_df['final_url'] != 'Error')]
cleaned_df = cleaned_df.drop_duplicates(subset='pid', keep='first')
cleaned_df = cleaned_df.drop_duplicates()
cleaned_df.to_csv(scientists_output_file, index=False)
```

1.2 Data Scraping

The scrape_scientist function is designed to programmatically extract structured metadata—such as publication titles, years, DOIs, and authors—from a given scientist's DBLP profile page. Each row in the dataset corresponds to an individual researcher, represented by a unique pid (persistent identifier) and their corresponding DBLP URL (final_url). The function performs an HTTP GET request to access the HTML content of the page and parses it using the BeautifulSoup library.

- **Title Extraction**

The title of each publication is usually embedded within a tag with the class "title". The scraper extracts it using:

```
        title_tag = entry.find('span', class_='title')
        title = title_tag.text.strip() if title_tag else 'N/A'
```

If the tag is absent, "N/A" is recorded, ensuring downstream consistency.

- **Year Extraction and Validation**

The year is first sought in the tag. If this tag is unavailable, the function falls back to a heuristic method involving regular expression pattern matching:

```
year_matches = re.findall(r'\b(19\d{2}|20\d{2})\b', entry.text)
```

The initial version of the scraper had a flaw where any four-digit number starting with 20 was interpreted as a year, even when it actually represented a *page number* or other numerical identifier (e.g., "2070" from page 2070–2075). To mitigate this, a post-processing filter was added to discard values above 2025. If no valid year is found, the entry is logged for manual review:

```
valid_years = [int(y) for y in year_matches if int(y) <= 2025]
```

- **DOI and ArXiv Link**

To ensure broad coverage of digital identifiers, the scraper searches for hyperlinks whose href attributes match common DOI or ArXiv URL patterns. This technique is more robust than relying on title='DOI' attributes, which may not be consistently present.

```
    doi_tag = entry.find('a', href=re.compile(r'(doi\.org|arxiv\.org)'))
    doi = doi_tag['href'].strip() if doi_tag else 'N/A'
```

- **Author List Aggregation**

Author names are embedded in tags with the itemprop="author" attribute. These are parsed and concatenated into a comma-separated string, ensuring compatibility with tabular or spreadsheet-based downstream processing.

```
author_tags = entry.find_all('span', itemprop='author')
authors = ', '.join([a.text.strip() for a in author_tags]) if author_tags else 'N/A'
```

The scraped publication data is systematically cleaned to ensure accuracy and consistency. This involves removing duplicate entries, both entirely and based on the publication title, checking for

missing values and empty strings across all fields, and identifying rows with incomplete information. The cleaned dataset is saved to a CSV file(papers_cleaned.csv) in the output folder.

2 Network Property

This section of the report analyzes the collaboration network among a group of 1,039 scientists based on their co-authorship on scientific papers. The network is constructed such that each node represents a scientist, and an undirected edge connects two scientists if they have co-authored at least one paper. Edge weights represent the number of co-authored papers.

2.1 Methodology

To analyze the network of data scientists from the datascientists.xls file, the collaboration network is constructed using two datasets: scientists_cleaned.csv and papers_cleaned.csv. To ensure data accuracy and network reliability, the process begins by validating the scientist names in the scientists_cleaned.csv file. In the papers_cleaned.csv dataset, the authors are filtered to match only those found in the validated list of scientists. Finally, a graph is constructed where edges represent co-authorships between valid authors, and the edge weight indicates the number of shared papers. This approach ensures that the resulting collaboration network is both accurate and meaningful.

2.2 Basic Network Properties

The collaboration network comprises 1,039 nodes and 4,720 edges. The average degree is approximately 9.09, meaning that, on average, each scientist has collaborated with about nine others. The network density, calculated as the ratio of actual connections to all possible connections, is 0.0088, indicating a sparse network. Despite the total number of nodes, the network is fragmented into 287 connected components, suggesting that many scientists are part of smaller, disconnected groups or have not collaborated beyond a limited circle.

One key measure of local cohesiveness is the average clustering coefficient, which stands at 0.2684. This relatively high value suggests that scientists often collaborate within close-knit groups where co-authors are likely to also collaborate with one another, forming densely connected triads or cliques.

2.3 Giant Component Analysis

Figure 1: Giant Component

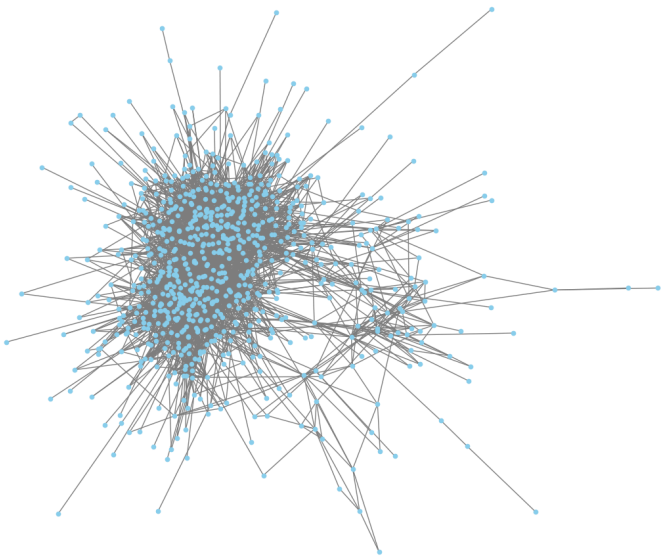


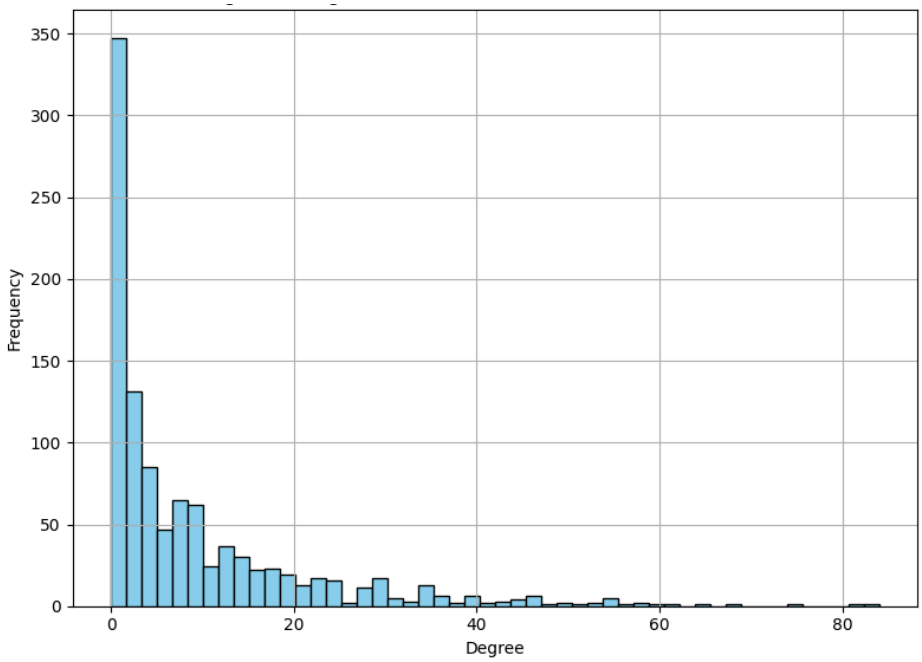
Figure 1 provides a visualisation of the largest connected subgraph, revealing a dense core of interconnected scientists, surrounded by peripheral nodes with fewer connections. This visual reinforces the idea of central hubs, with several key individuals linking otherwise disconnected regions of the network.

From further analysis, we found that the giant component encompassed 747 scientists (approximately 71.9% of the network). This giant component contains 4,712 edges and exhibits properties characteristic of small-world networks. The network diameter within the giant component is 9, which means that the longest shortest path between any two scientists in this subgraph spans nine connections. The average shortest path length is 3.23, indicating that most scientists in this component can be reached from one another in just over three steps.
The large number of isolated nodes, 292, suggests that there are a fair number of scientists who are not involved in direct collaboration, thus forming isolated sub-networks or even singletons.

The giant component's properties suggest that despite a few isolated individuals, a substantial portion of the network is connected, enabling efficient information flow and collaboration within this core subset of the network.

2.4 Degree Distribution

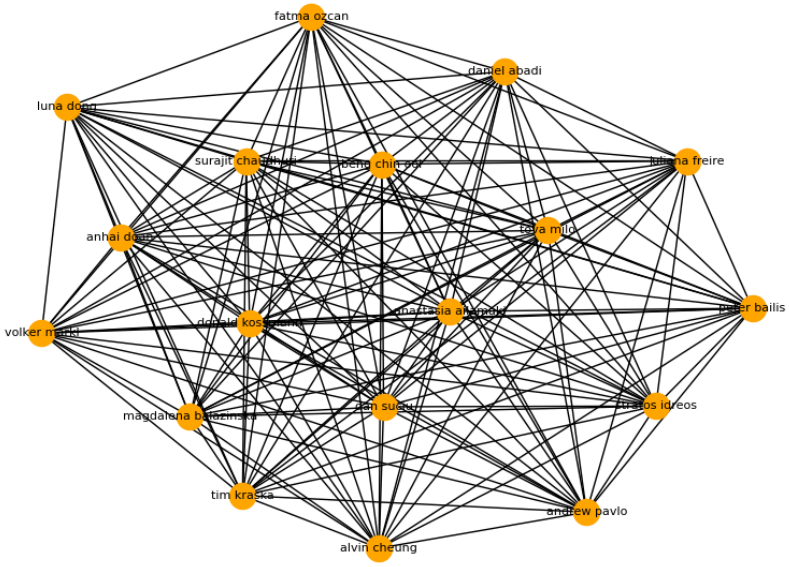
Figure 2: Degree Distribution



The degree distribution of the network shows the number of collaborations each scientist has. As seen in Figure 2, most scientists have fewer than 50 collaborators, while only a small number have degrees as high as 80–90. This skewed distribution indicates a typical pattern seen in real-world social networks: a majority of nodes have low degree, while a few central nodes act as hubs with very high connectivity.

2.5 Clique Structure

Figure 3: Largest Clique



The largest clique in the network consists of the most tightly-knit group of scientists, where every member is directly connected to every other member. In the context of collaboration networks, this group of 18 represents the set of scientists who have a highly cohesive relationship with each other, working together on a set of shared projects.

Beng Chin Ooi who has one of the highest betweenness centrality of 0.0238, in the largest clique indicates that this scientist is at the core of one of the most cohesive collaboration subgroups. This could suggest a leading role in a specific area of research or a pivotal position within this tightly interconnected group.

2.6 Correlation Between Degree Centrality and Degree

Figure 4: Degree Centrality vs. Degree

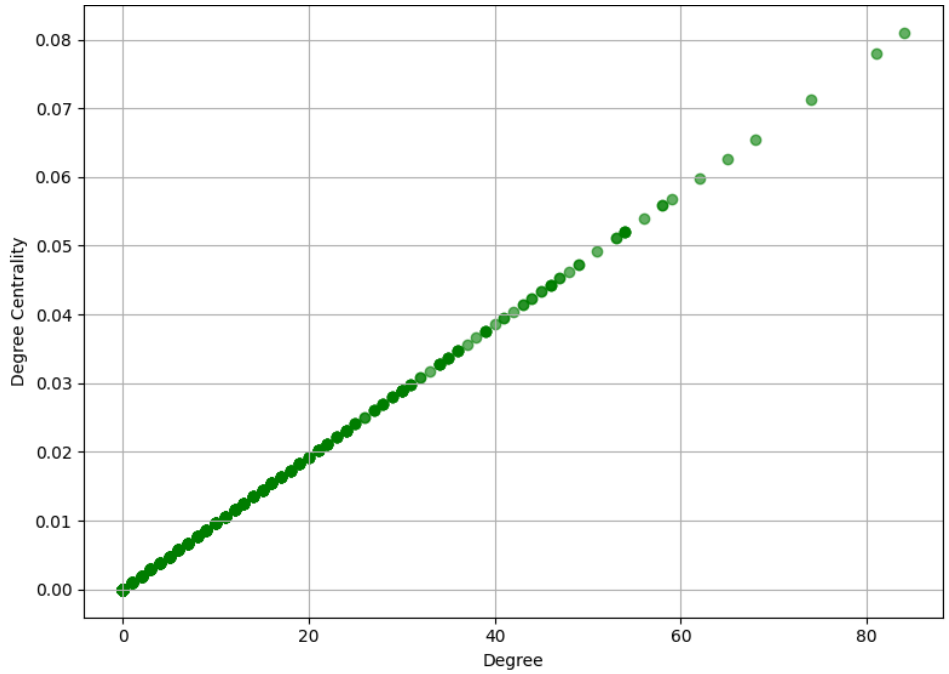
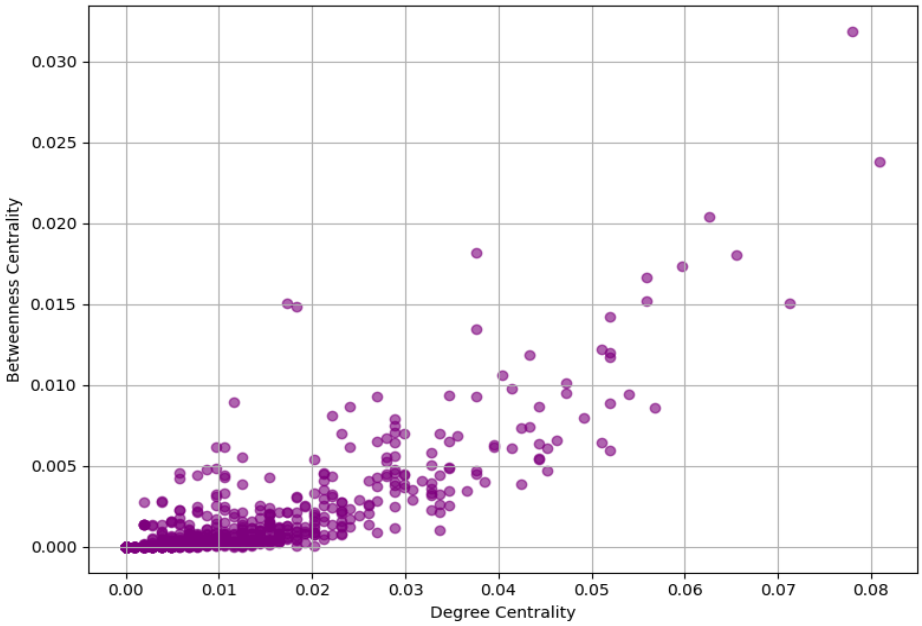


Figure 4 illustrates the relationship between degree centrality and the degree of nodes in the network. Degree centrality measures the relative importance of a node in terms of its direct connections, and it should, in most cases, correlate well with the degree (the number of direct neighbors a node has).

The correlation between degree and degree centrality suggests that the more connections a scientist has, the more central they are in the network..This relationship is typical in social networks, where highly connected individuals often play significant roles in the flow of information and resources within the network.

2.7 Correlation Between Degree Centrality and Betweenness Centrality

Figure 5: Degree Centrality vs Betweenness Centrality

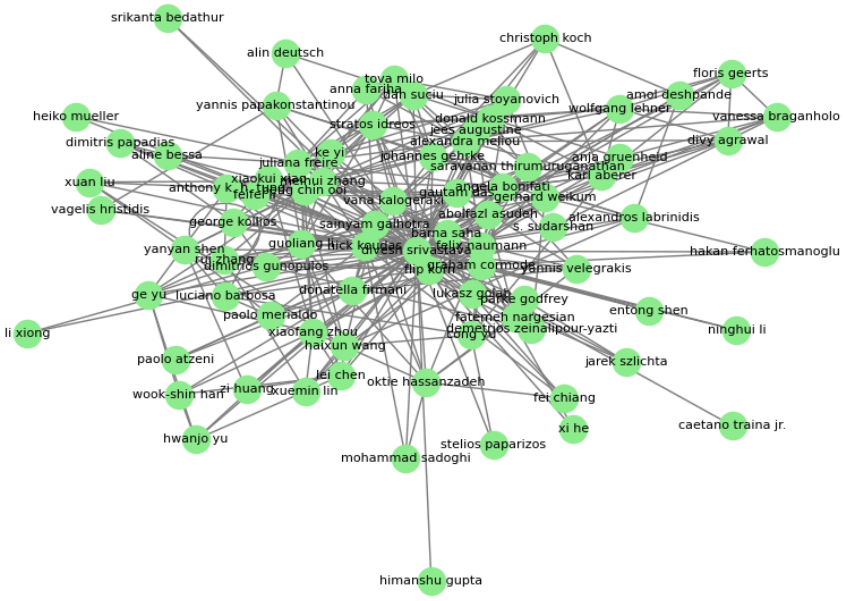


In Figure 5, the scatter plot compares degree centrality with betweenness centrality. Betweenness centrality measures how often a node appears on the shortest path between any two other nodes, indicating the degree to which a node serves as an intermediary in the network.

As shown, the majority of nodes are concentrated in the bottom-left of the scatter plot, indicating that most scientists in the network have both low degree centrality and low betweenness centrality. This suggests that most scientists are not critical intermediaries or bridges within the network. There is also a positive correlation which implies that nodes with higher degrees also often lie on the shortest paths to other nodes. Their role as intermediaries is crucial for maintaining the flow of information and collaboration across different communities.

2.8 Ego Network Analysis

Figure 6: Ego Network Visualisation of Divesh Srivastava



Divesh Srivastava who has the highest betweenness centrality (0.0319), is the central node in this network, consisting of 82 nodes. This includes Divesh Srivastava and the nodes directly connected to it. This network is a localized subset of the larger collaboration network, where the focus is on the immediate connections and the direct relationships that Divesh shares with other scientists in the network.

This ego network signifies a substantial local network of connections. The size of the ego network highlights the extensive reach of Srivastava within the broader collaboration network, with many direct connections. This is further supported by his publication history, with the large number of papers he authored between 1990-2025 as recorded on his DBLP profile.

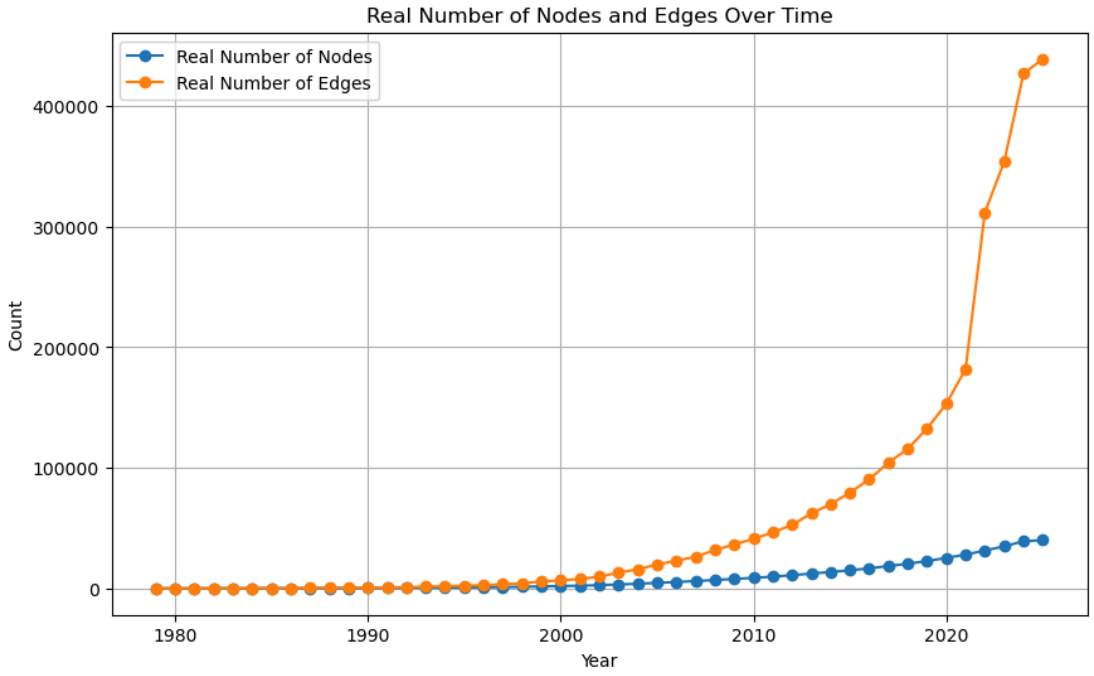
3 Network Evolution Over Time

This section will analyse the real collaboration network over time from 1979 to 2025 (present). The data used in this section contains all the scientists who had published papers on https://dblp.uni-trier.de/, including but not limited to scientists provided in the csv. This is essential as many scientists are not included in the given list and will affect the network analysis for every year.

3.1 Nodes and Edges

One of the basic characteristics of a network is the number of nodes and edges. The following graph shows how the respective property evolved over time.

Figure 7: Real Number of Nodes and Edges Over Time



It can be seen that from 1980 to 2000, there is a minimum increase in both the number of nodes and number of edges. From 2000 to 2025, the number of nodes increased drastically from 3 to more than 39,974. The number of edges increased even faster, from 3 to around 439,062.

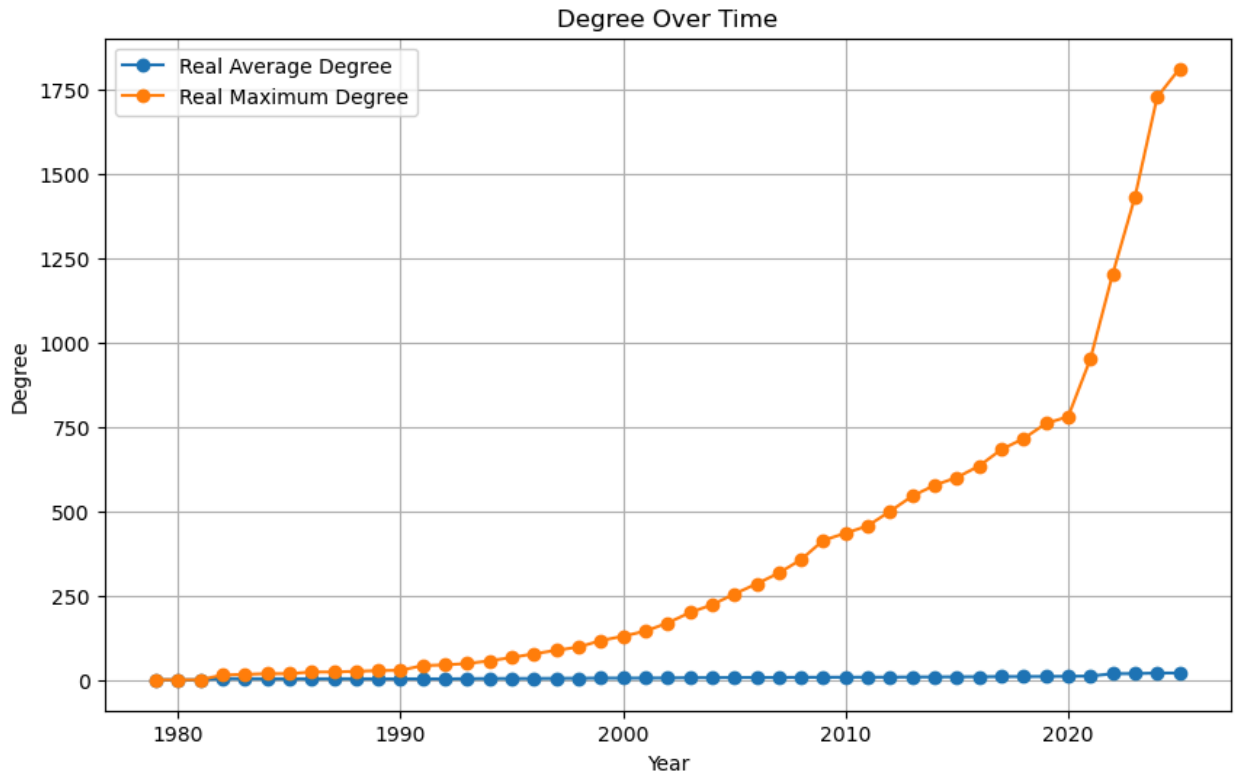
The increase in number of nodes might be due to the rise of many scientists after 2000 and the emphasis of education. The faster increase in the number of edges might be because each

scientist can collaborate with all other scientists, which results in polynomial increase. In a complete graph, the number of edges is the square of the number of nodes.

3.2 Degree

The following graph shows the average and maximum degree over time. From the graph, the average degree increased slowly from 2 in 1980 to 22 in 2025 and the maximum degree increased from 2 in 1980 to 1811 in 2025.

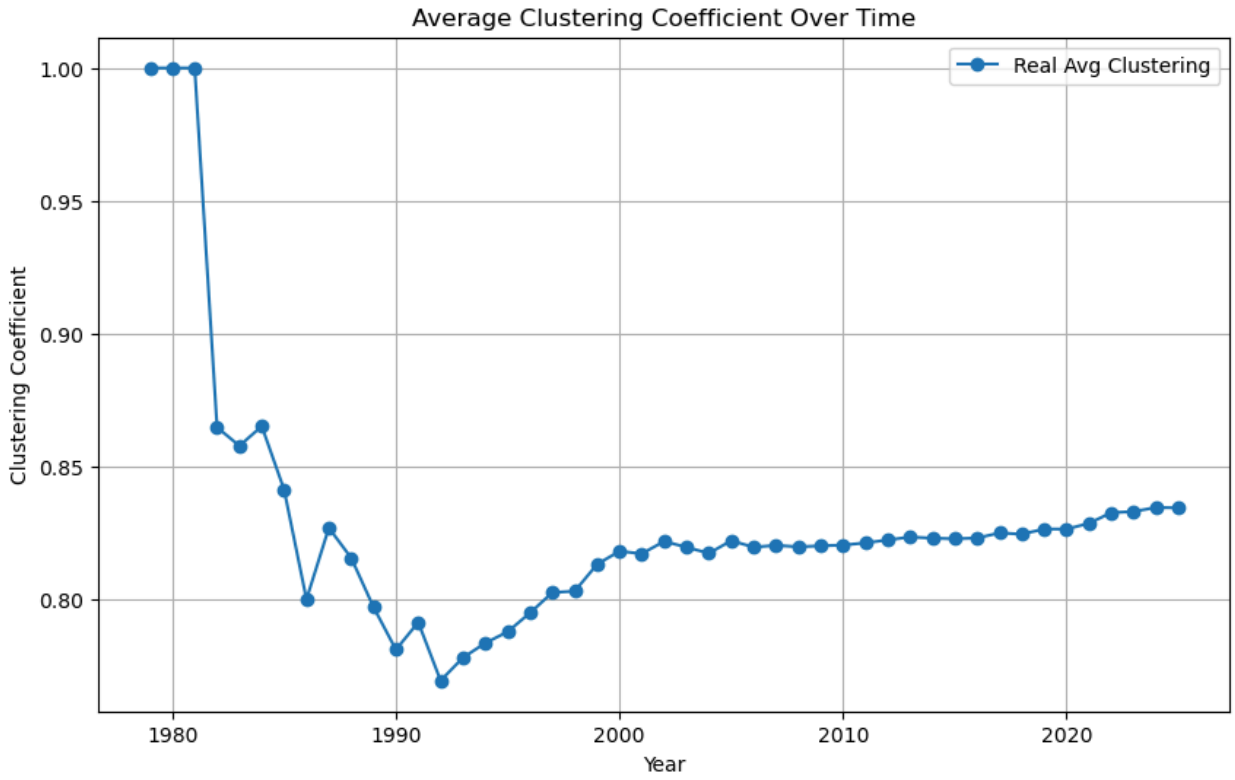
Figure 8: Degree Over Time



Degree is the number of edges or links a node has in a network. We can say that as time progresses, on average scientists have an almost constant number of connections, which no drastic increase even when the new scientists join the network. However, there are some scientists who might be more connected, who have up to 1811 connections. One hypothesis is that these scientists joined the network at an earlier age and over time, they accumulated connections with other scientists.

3.3 Clustering Coefficient

Figure 9: Average Clustering Coefficient over Time



Clustering coefficient is a measure of how neighbouring nodes are connected to a node or the cohesion in a neighbourhood of a node in the network. The graph shows the average clustering coefficient of all nodes in the network. In 1979, the average clustering coefficient was 1, which means that all neighbors of all nodes are connected. This is because in 1979, there were only 3 nodes with 3 edges. Due to the small size of the network, all nodes are connected.

From 1979 to 1992, the average clustering coefficient decreased from 1 to 0.770. However, there are ups and downs during the period, showing instability of the trend. This might be due to the small network and that clustering coefficient is sensitive to even a small change in the number of nodes and edges.

From 1992 to 2025, the average clustering coefficient increased steadily from 0.770 to 0.835. This shows that although the number of scientists are increasing, the number of connections between scientists increase even faster. This can be cross validated by the faster increase in number of edges than the number of nodes from the figure above.

Figure 10: Number of Connected Component over Time

Figure 11: Average Shortest Path Length over Time

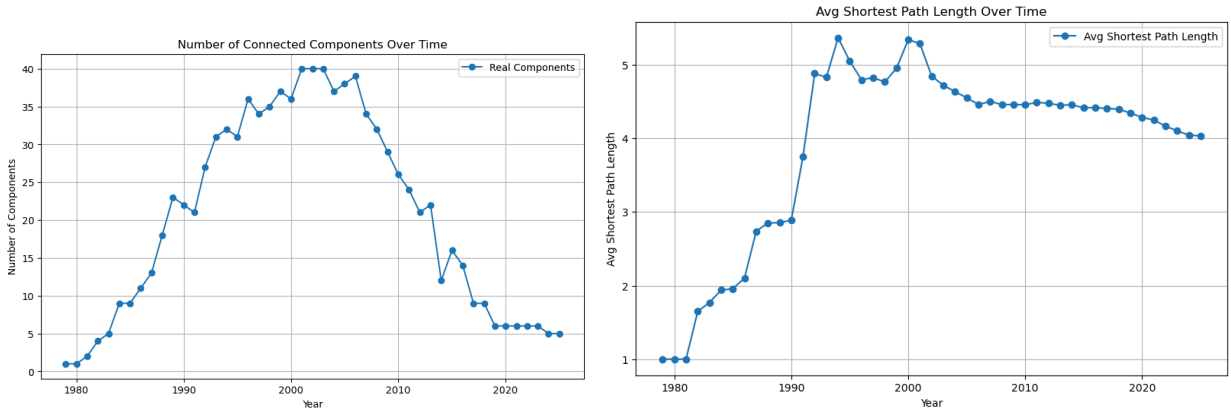
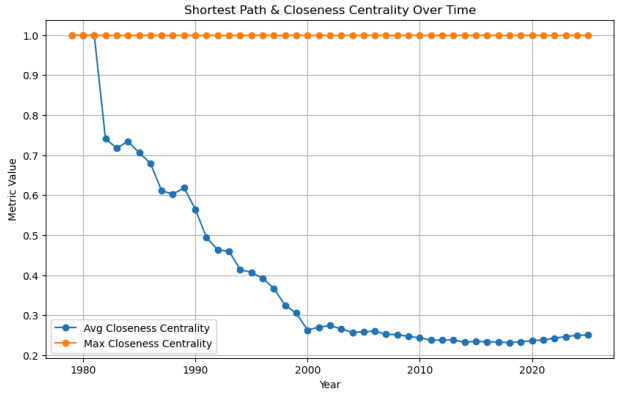


Figure 12: Shortest Path & Closeness Centrality over Time



The figures 10, 11, 12 show the number of connected components, average shortest path and closeness centrality over time. These 3 graphs have a similarity, they all have a turning point at around 2000.

Before 2000, there were a limited number of scientists and thus connections between them as shown above. This leads to an increase in the number of components from 1 to 40. This can be cross validated from the above graph where the number of nodes and edges are almost the same before 2000, showing that there are limited connections per scientist.

The average shortest path also increased from 1 to 6 in 1979 to 2000 and the closeness centrality decreased from 1 to 0.28. This is because as the number of scientists increased, the number of connections did not increase polynomially, leading to longer shortest paths and thus, lower closeness centrality.

After 2000, more scientists joined the network and the number of edges increased more than proportionally. This explains the decrease in the number of connected components from 40 to 5 as beside linking the new scientists into the network, there are connections to link the small components together. This also explains the gradual decrease in the average shortest path from 6 to 4, and thus the consistency of average closeness centrality.

Overall, before 2000, there were a limited number of scientists and the number of connections increased less than the polynomial of the number of scientists. After 2000, the number of scientists increased drastically and the number of connections increased even faster. This shows that more papers are published after 2000 or scientists are collaborating more with others who they do not know. Thus, the number of components decreased, but the size of components increased.

4 Real VS Random Network

In this section, a random graph of scientists is created according to the properties of the real connect graph. From the picture below, it can be seen that the number of nodes and the number of edges are kept constant, with just the edge randomly assigned.

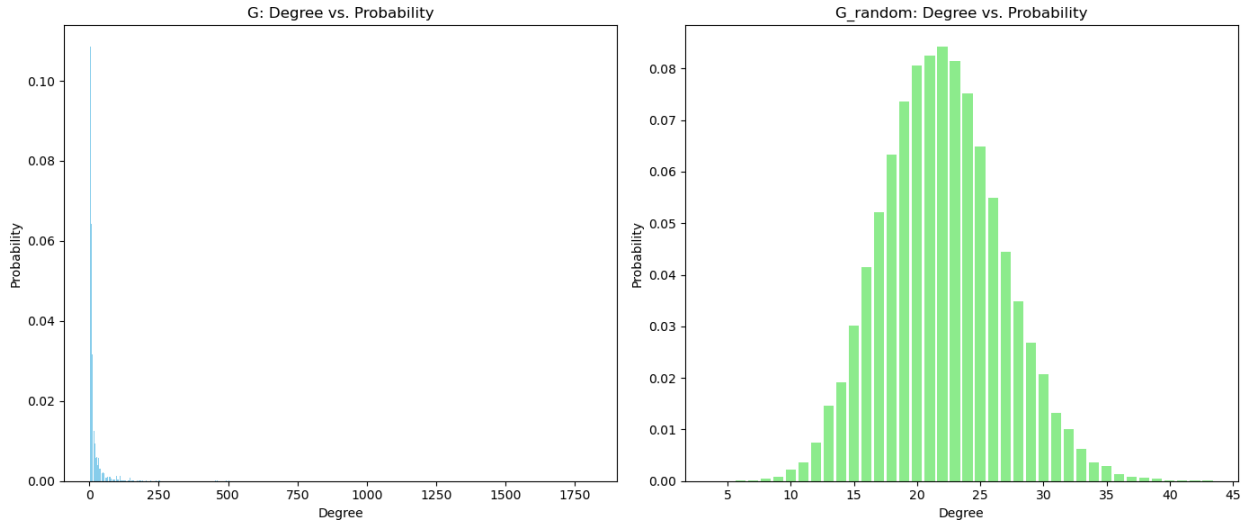
```
random_nodes = scientists_df['pid'].astype(str).unique()
n_random = len(random_nodes)

# Get the last year's real edge count
final_year = real_stats_df.index.max()
m_real = int(real_stats_df.loc[final_year, 'real_num_edges'])

# Generate a single random graph
G_random = nx.gnm_random_graph(n_random, m_real)
mapping = {i: random_nodes[i] for i in range(n_random)}
G_random = nx.relabel_nodes(G_random, mapping)
g = ig.Graph.TupleList(G_random.edges(), directed=False, vertex_name_attr='name')
```

4.1 Degree

From question 2, we can see that in 2025, the average degree is 22 but the maximum degree is 1811. This huge difference indicates that there are extreme values and that the distribution of degree is not even.

The graph below shows the comparison of degree distribution between the real network and the random network just created.

Figure 13: Real vs Random: Degree vs Probability

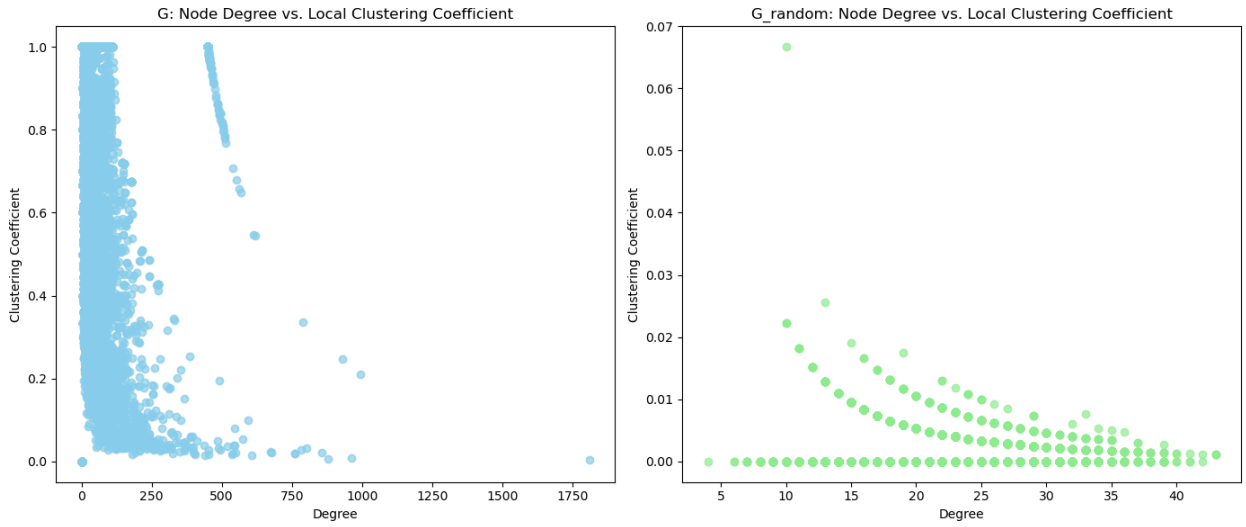


Just as hypothesized, the distribution of degree of real network is highly skewed to the left whereas in random network, it follows a binomial distribution. One of the possible reasons for the low average degree in the real network as compared to the random network is that scientists are connected via publishing a paper together. It is almost impossible for scientists to publish around 840 papers and each with different other scientists. Thus, the random graph does not follow the logic of connections within scientists. Another reason is the power law degree distribution, where the real network is a scale free network and that there are some nodes acting as hubs with very high degree.

4.2 Clustering Coefficient vs Degree

Other than the difference in degree probability, it is also worth noting the connections in the neighbourhood. This can be seen from plotting degree vs clustering coefficient as seen below.

Figure 14: Real vs Random: Degree vs Clustering Coefficient



It can be seen that in a random network (right side), there is a weak reverse relationship between degree and clustering. This matches the theoretical expectation for an Erdos Renyi random graph with: $C \approx p$, where C is the clustering coefficient and k is the degree. The relationship is weak as the clustering coefficient is consistently low across all degrees with some noise.

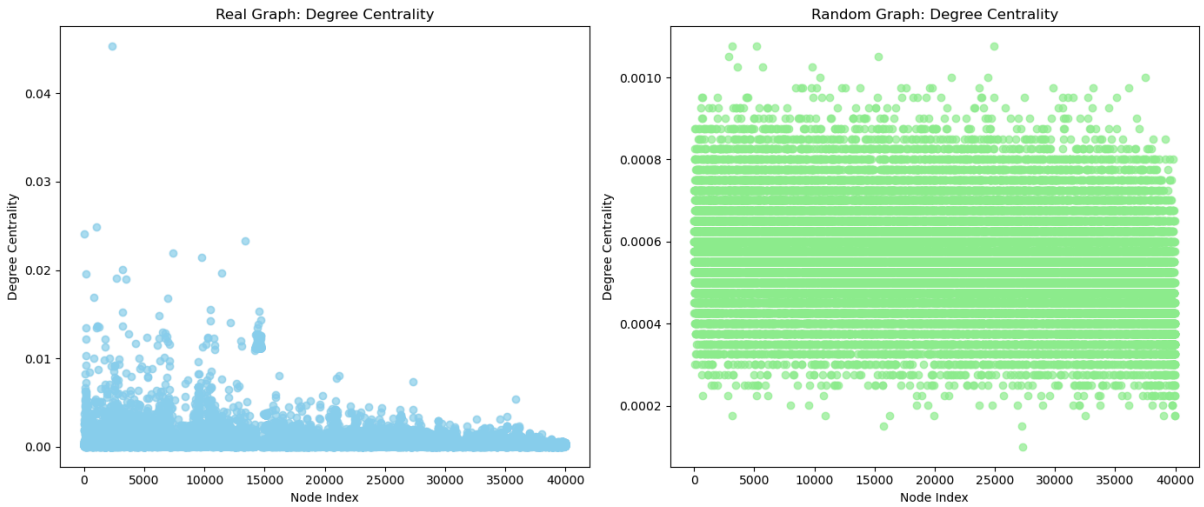
Different from the random network, the real network shows a strong relationship where there is also an inverse relationship between clustering coefficient and degree, indicating that higher-degree nodes are less embedded in tightly-knit groups. In other words, when a scientist

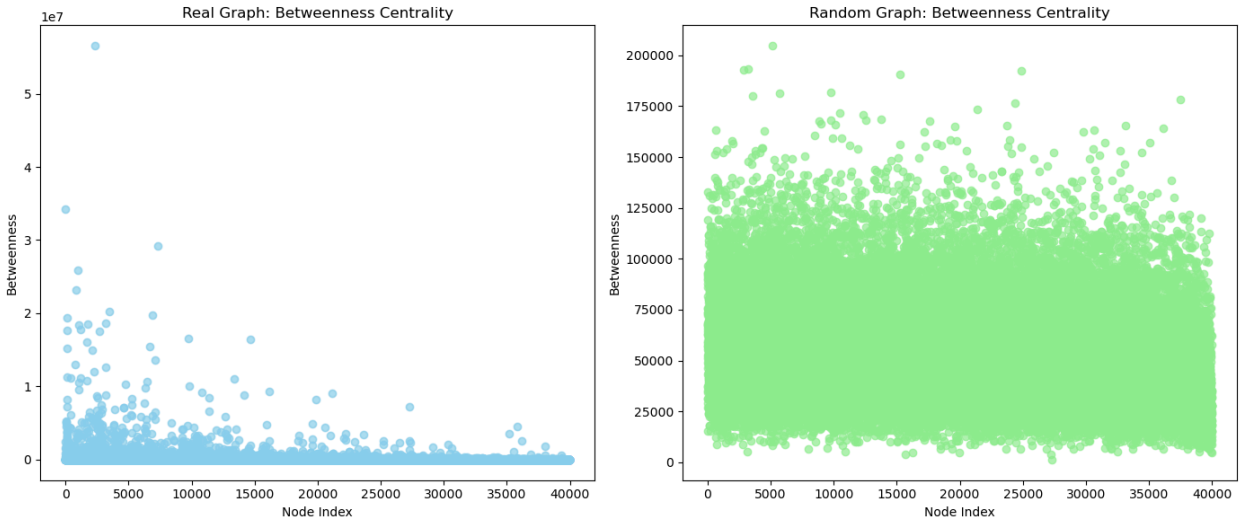
connects with more of other scientists, these connected scientists have a lower probability of connecting to each other. One of the possible explanations for this is preferential attachment, where scientists are more willing to connect and collaborate with scientists of higher degree due to fame or other reasons. This creates hubs which a random network does not have, where inside a hub, as the number of scientists are large, the probability of them knowing and collaborating with each other are low. This contrast highlights the limitations of the ER model in capturing the modular, hierarchical nature of real networks.

4.3 Degree and Betweenness Centrality

The following graph validates the presence of hubs in the real network and not in the random network.

FIgure 15: Real vs Random: Degree Centrality, Betweenness Centrality





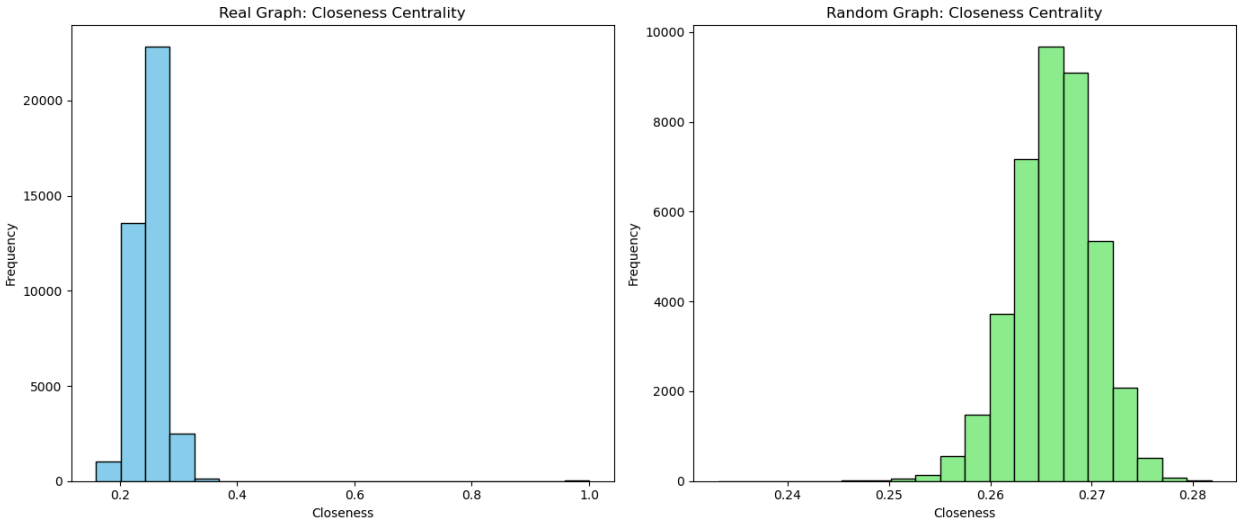
From the top graph, it can be seen that in the real network, some nodes have high degree centrality of around 0.045 while most of the nodes have degree centrality of close to 0. Whereas in the random network, there is an even distribution of degree centrality among all nodes. This shows that in the real network, there is a group of scientists who are more important and most scientists are not important, stating the presence of hubs. In a random network, most scientists are of equal importance and there are no scientists who are more important than others.

Similarly, this trend can be seen from the bottom graph where some scientists have higher betweenness centrality of 50,000,000 in the real network and most scientists have betweenness centrality close to 0. In the random network, the betweenness centrality is random and there is no obvious group of scientists with a much higher degree centrality. This shows that in the real network, some scientists are more essential and act as a link between many other scientists, confirming the existence of hubs.

4.4 Closeness Centrality

In the following graph, closeness centrality is measured, there it is the average shortest path to all nodes, averaging of all nodes.

Figure 16:Real vs Random: Closeness Centrality



It can be seen that in the real network, the low closeness centrality has higher frequency. As closeness centrality is calculated as the inverse of the shortest path to all nodes, a low closeness centrality value means most scientists require on average around 5 other scientists to know each other. While a small number of scientists has high closeness centrality and shorter shortest path to know others. However, in a random network, the distribution is not tailed, but a normal distribution. This shows that all scientists are almost the same, with no special0 scientists who are able to have shorter shortest paths. This again confirms the presence of hub in real network but not in random network

5 Network Transformation

In this section, we will develop a function that will generate a new network from the existing collaboration network of the data scientists into a new network.

The new network will be created based on the following goals:

1. Compared to the initial collaboration network, the modified network will contain more isolates and a smaller huge component.
2. Maximum degree of any node will not go beyond collaboration cutoff (a user-specified k_max, which will be smaller than the degrees of hubs.
3. In order to find connections between well-known data scientists (high degree) and their less well-known co-authors (low degree nodes), the network will be altered.

To maintain the diversity of nodes in the network, we decided against removing any nodes from the network. Therefore, the transformed network will have the same number of nodes as the untransformed network. We instead focus on removing unwanted edges from the network. In general every network transformation technique had two parts i.e define a metric of edge importance and define a policy to prune edges of less importance

5.1 Algorithm - Hard Cutoff for Normalised Degree Difference

This algorithm has two parts:

1. Edge Importance Metric - To measure the importance of each edge we defined a metric Normalised Degree Difference as:

$$Normalised\ Degree\ Difference\ (E_{n1,n2}) = \frac{|C_D(n1) - C_D(n2)|}{\max\left(C_D(n1), C_D(n2)\right)}$$

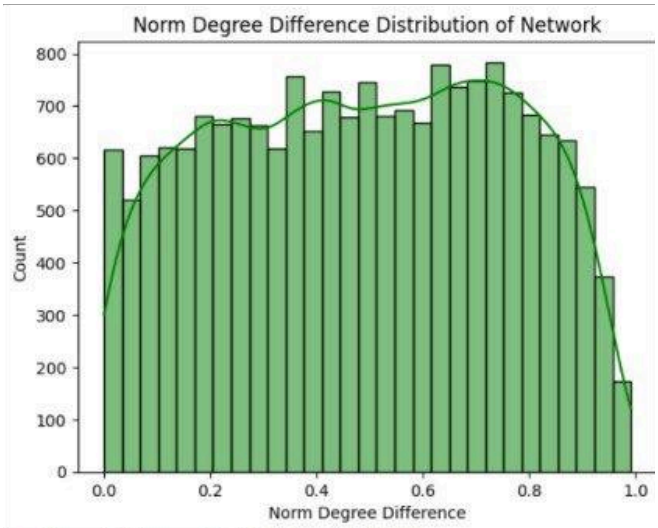
where n1 and n2 are nodes in the graph, En1, n2 is the edge connecting n1 & n2, and CD(x) is the degree of node x. The expected values of Normalised Degree Difference for different edges are:

a. Hub - Hub Edges => Small Numerator and Large Denominator => Very Small - Small Value
b. Hub - Normal Edges => Large Numerator and Large Denominator => Medium - Large Value
c. Normal - Normal Edges => Small Numerator and Small Denominator => Small - Medium Value

Hence, edges with a smaller Normalised Degree Difference are more likely to be edges between hubs or between normal nodes, rather than those between hubs and normal nodes.

2. **Edge Pruning Policy** - We establish a hard cutoff, determined by the user-specified k_max, to trim the network's edges. The network will be cleared of all edges whose edge relevance is less than this cutoff. The network's Normalized Degree Difference distribution is displayed in the figure below. This metric is perfect for usage with a hard threshold and numerous values of k_max because of the uniformity of the distribution. With a hard threshold, metrics with exponential distributions perform poorly because, depending on the user-specified value of k_max, they either eliminate too many edges or too few.

Figure 17



The relationship between the user-specified value of k_max and the size of the huge component of the modified network, using the Hard Cutoff for Normalized Degree Difference algorithm, is shown in the figure below. Depending on the input k_max, we can see that the method returns a range of values for the enormous component's size.

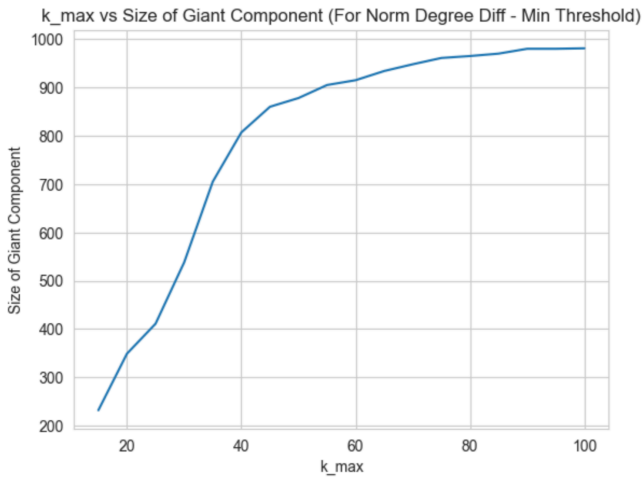
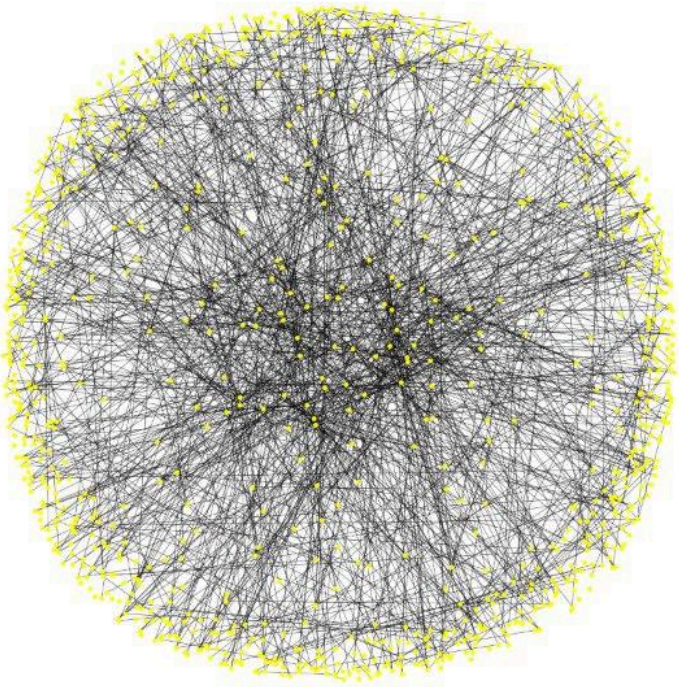


Figure 18

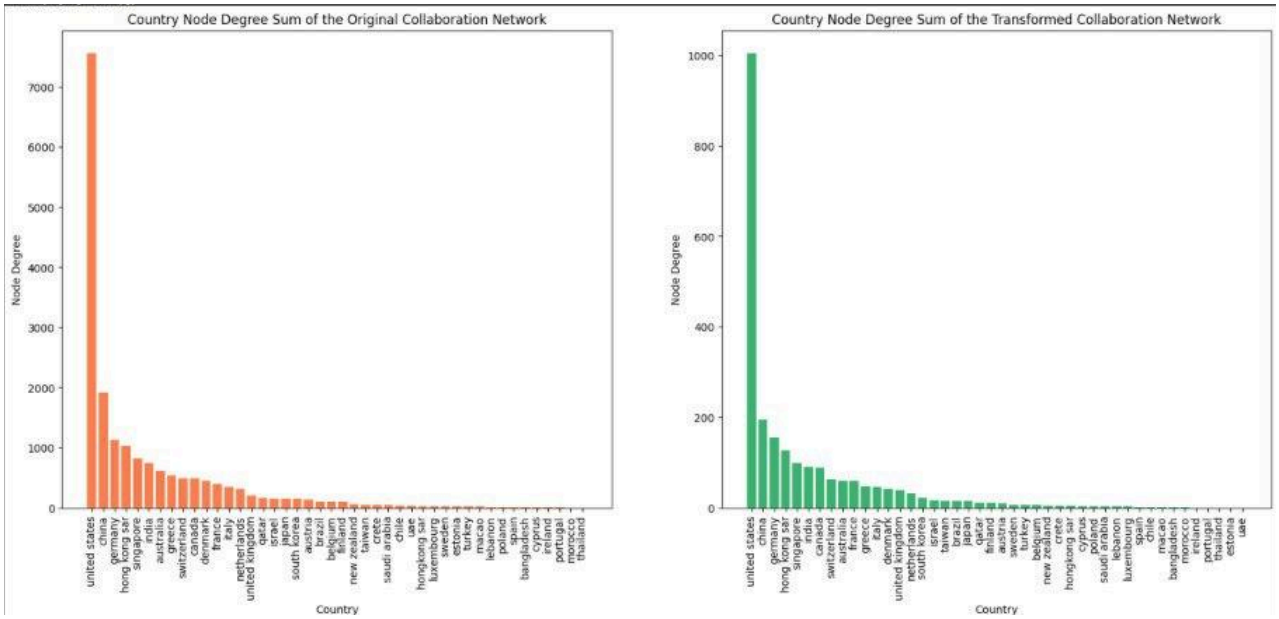
In Figure below we can see a transformed network with k_max = 35. It has a giant component with 705 nodes, lesser compared to the original network

Figure 19 - Transformed Collaboration Network of Data Scientist



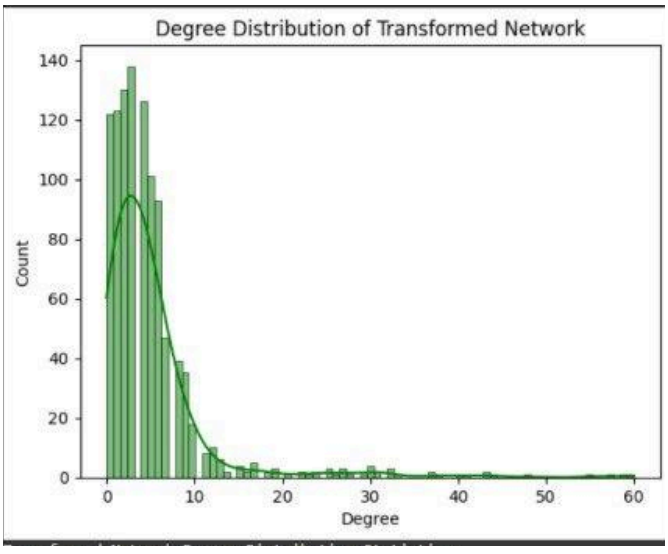
Finally, to make sure that the newly changed network maintains the same variety as the original, we will compare the node degree sum for the nation, the scientists' institution, and their level of expertise. The node degree sum distribution for the nation stayed nearly unchanged when the amount was reduced seven times, as shown in the figure below. This demonstrates that we were able to lower the network's node degrees while preserving the scientists' original country distribution.

Figure 20



The same thing can be seen in the figure below too, again, the distribution of the scientists' expertise remained the same with just the sum being reduced significantly.

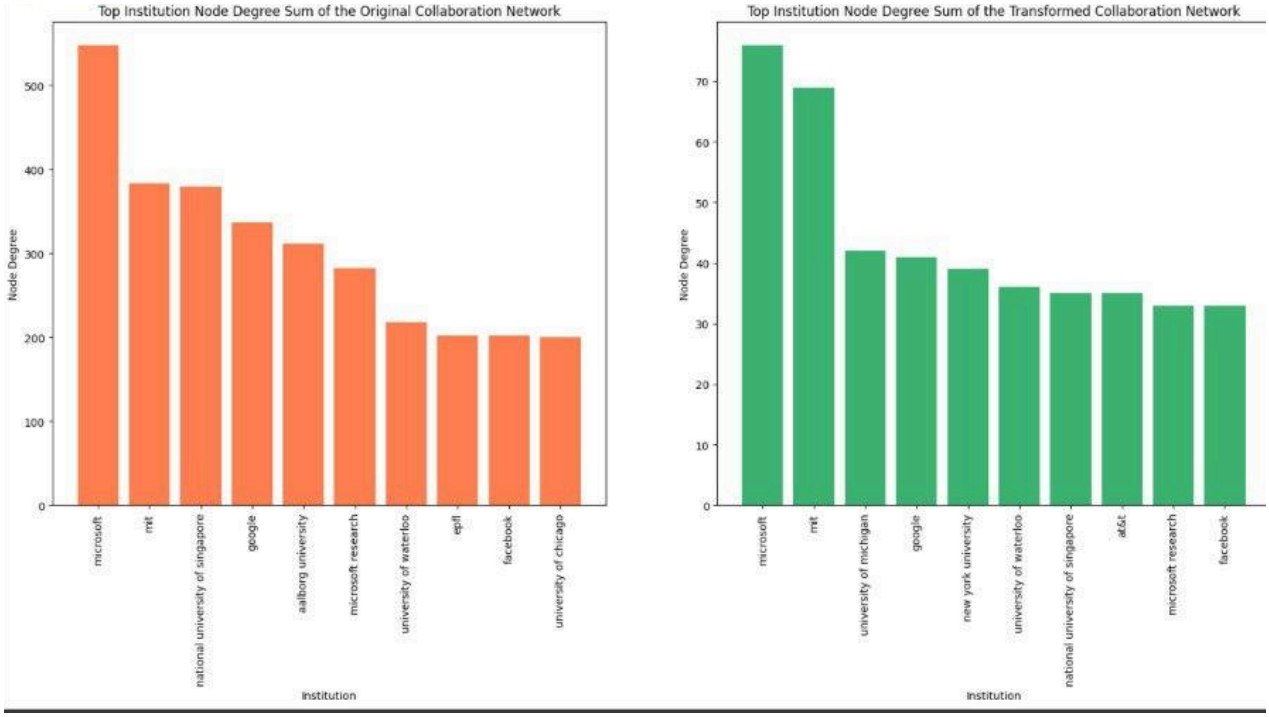
Figure 21



Lastly, we were also able to keep the scientists' institutions distributed similarly to how they were in the original network. As seen in the figure below, the majority of the institutions with the

greatest degree sums stayed the same, even though some places altered. Because it has many more discrete values than the other two distributions, this one is the most difficult to maintain. In accordance with the network transforming model's specifications, the sums' values were also decreased.

Figure 22



In conclusion, judging by the figures above, we can guarantee that our algorithm successfully manages to reduce the k_max and the node degrees k_i in the network while maintaining all distributions and diversities from the original network.

Limitation and Conclusion

In this report, there are some limitations that could be improved in further studies.

1. Restricted data

   Due to the limitation of time and cpu resources, for some sections of this report we only used 1039 scientists from the given list. This may cause biases in the network structure and some characteristics may not be captured correctly.

2. Static view only

   This report only generated static visualization of graphs and figures. If possible, dynamic or videos on the network revolution would yield more insight to the graph properties over time.

With this, future studies could include more data for analysis and may include multi model comparison such as the Watts-Strogatz graph against the real collaboration network. Node or edge removal could also be included if sufficient data is provided.

In conclusion, this report confirms that scientist collaborations are not random but also not uniform. They form clusters and hubs especially after 2000. There is also clear evidence of preferential attachment due to natural human behavior.

Appendix

Github Link:

https://github.com/Xkpd/Network-Science-Project/tree/main