CSE3024- Web Mining

Report

Application of Social Network Analysis to Investigate COVID-19 Transmission Among Contacts

By

20BCE1182	Sheral Simon Waskar
20BCE1320	Anjali Jain

B.Tech CSE

Vellore Institute of Technology,
Chennai, Tamil Nadu.

Submitted to

Dr.A.Bhuvaneswari,

Assistant Professor Senior,

SCOPE, VIT, Chennai

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

February 2023

School of Computing Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

WINTER SEM 22-23

Worklet details

Programme	B.Tech CSE	
Course Name / Code	Web Mining - CSE3024	
Slot	A2	
Faculty Name	Dr.A.Bhuvaneswari	
J Component Title	Application of Social Network Analysis to Investigate COVID-19 Transmission Among Contacts	
Team Members Name | Reg. No	Sheral Simon Waskar	20BCE1182
	Anjali Jain	20BCE1320

Team Members(s) Contributions – Tentatively planned for implementation:

Worklet Tasks	*Contributor's Names*
Topic selection & Dataset Gathering	Sheral
Preprocessing	Sheral
Model building	Anjali
Visualization	Anjali
Technical Report writing	Sheral and Anjali
Presentation preparation	Sheral and Anjali

BONAFIDE CERTIFICATE

Certified that this project report entitled “**Application of social network analysis to investigate covid-19 transmission among contacts**” is a bonafide work of **Sheral Simon Waskar- (20BCE1182) and Anjali Jain(20BCE1320)** who carried out the Project work under my supervision and guidance for **CSE 3024 -WEB MINING.**

Dr.A.Bhuvaneswari,

Assistant Professor Senior,

SCOPE, VIT,

Chennai-600127

ACKNOWLEDGMENT

We would like to express our deepest gratitude to Dr. Jani Anbarasi L for giving us this opportunity for learning and professional development. We concerned so many wonderful people and professionals who led us through this Project period.

We would also like to extend my gratitude towards our classmates for their co-operation and for sharing their knowledge with us.

I am also obliged to VIT University for giving us this opportunity that we perceive as a big milestone in my career development. We will strive to use gained skills and knowledge in the best possible way. we would also like to thank our friends and family members for their constant support and encouragement without which this Project would not have been possible.

Sheral Simon Waskar
(20BCE1182)

Anjali Jain
(20BCE1320)

ABSTRACT

Contact tracing data is used from the severe acute respiratory syndrome coronavirus 2 (SARS- CoV-2) pandemic to estimate basic epidemiological parameters. Contact tracing data may also be used to assess heterogeneity of infection at the individual patient level. Characterizing individuals based on different levels of infectivity can better inform field-level contact tracing measures.

It could help identify, test, and confine close contacts to prevent further spread. The effectiveness of this approach has been amply established in the monitoring and management of COVID-19 (coronavirus disease 2019) epidemics. The goal of this study is to use contact tracing data to examine the level of dissemination and the emergence of transmission cascades made up of numerous clusters.

This study aims to leverage contact tracing data to investigate the degree of spread and the formation of transmission cascades composed of multiple clusters.This study examines why it is important to study all combined interventions against the infection to analyse the infection outbreak size. In addition to this, it also aims to minimize the spread of virus in the given network by analysing the communities and their impact on others.

We have constructed a graph in which individuals were considered as nodes and links as the infection relationships between them. It was a directed graph in which the links will be directed from the covid positive patient who came in contact with a covid negative patient which resulted in the latter being infected with covid.

The node with the highest degree centrality will be considered as the node responsible for the spread of the virus on a large scale. Betweeness centrality measure will be used to find which node was responsible for passing on the virus from one network to another.

We will be clustering the nodes based on their level in the infection tree and analysing the impact of each level on the spread of covid.

Keywords - *Degree Centrality. Closeness Centrality, Betweenness Centrality, EigenVector Centrality,*

INTRODUCTION

Social Network Analysis (SNA) is a method used to visualize and understand the relationships and interactions among individuals in a network. With the outbreak of the COVID-19 pandemic, SNA has become an important tool to help track and control the spread of the virus. In the context of investigating COVID-19 transmission, SNA can be used to identify and analyze the connections between individuals who have been in close contact with one another and have tested positive for the virus.

This information can then be used to inform public health interventions, such as contact tracing and quarantine measures, to reduce the transmission of the virus. SNA can also help to identify potential super-spreader events, where a single individual is responsible for transmitting the virus to multiple others. This information can be used to target public health resources to the individuals and communities most in need, and to reduce the overall impact of the pandemic.

Overall, the application of SNA to investigate COVID-19 transmission is an important tool in the fight against the pandemic, and can provide valuable insights into the spread of the virus and inform effective public health strategies.

There are several challenges faced in the application of Social Network Analysis (SNA) to investigate COVID-19 transmission among contacts. Some of these challenges include:

Data Collection: One of the biggest challenges in SNA is obtaining accurate and comprehensive data on the relationships and interactions among individuals. This is particularly difficult in the context of COVID-19, as many individuals may not be willing to disclose information about their contacts or may not have a complete understanding of who they have been in close contact with.

Data Quality: The quality of the data used in SNA is critical to the accuracy of the results. In the case of COVID-19, data on close contacts may be incomplete or unreliable, as individuals may not remember all of their contacts or may not accurately report them.

Privacy Concerns: Privacy concerns are a major challenge in the use of SNA for investigating COVID-19 transmission. Individuals may be hesitant to disclose information about their close contacts due to concerns about privacy and confidentiality.

Bias in Data: SNA results can be influenced by biases in the data collected. For example, if individuals are more likely to report close contacts with certain individuals, such as family members or friends, this can skew the results and lead to a biased understanding of the spread of the virus.

Technical Challenges: SNA can be a complex and technical process, and requires specialized software and expertise to implement effectively. This can pose a challenge for public health officials and others who may not have the necessary technical skills or resources to implement SNA effectively.

Despite these challenges, SNA remains an important tool in the fight against COVID-19, and efforts are being made to overcome these challenges and ensure that SNA is used effectively to track and control the spread of the virus.

LITERATURE SURVEY

Sr. No	Title	Author / Journal name / Year	Technique	Result
1	Social network analysis methods for exploring SARS-CoV-2 contact tracing data	Nagarajan, K., Muniyandi, M., Palani, B. et al. BMC Med Res Methodol 20, 233 (2020)	Degree centrality and betweenness centrality	Network component analysis identified nineteen connected components comprising of influential patient's which have overall accounted for 26.95% of total patients (1959) and 68.74% of epidemiological contacts in the network.
2	Using high-resolution contact networks to evaluate SARS-CoV-2 transmission and control in large-scale multi-day events	Rachael Pung et al.r Nature Communications, 2022	Used contact tracking device data to analyze the risk of SARS-COV-2 outbreaks, which can provide a clear-cut evaluation of the interactions such as contact distance and duration of time of contacts	The 95th percentile of the epidemic size is around three times greater than the projected outbreak size, resulting in the largest number of infections from all other permutations of interventions

3	A social network analysis of the spread of COVID-19 in South Korea and policy implications.	Jo, W., Chang, D., You, M. et al. Sci Rep 11, 8581 (2021)	Network analysis, hypothesis tests on the distributions of network indicators, and virtual structural changes in the network.	This study utilized actual data to provide limited but meaningful results.
4	The characteristics of COVID-19 transmission from case to high-risk contact, a statistical analysis from contact tracing data	Chayanon Phucharoen Clinical Medicine,20 2 0	The paper aims to explore the various factors that contribute to the transmission of COVID-19 or SARS-CoV-2 in Phuket.	Using the Probit model, the paper analyzed the high- risk contacts of the public health office in the region who were involved in the confirmed cases of COVID-19. It also looked into the impact of quarantine measures on the probability of infection among individuals.
5	Mining relationships between transmission clusters from contact tracing data: An application for investigating COVID-19 outbreak	Tsz Ho Kwan, Ngai Sze Wong, Eng-Kiong Yeoh, Shui Shan Lee Journal of the American Medical Informatics	An algorithm on mining relationships between clusters for network analysis is proposed with 3 steps: horizontal edge creation, vertical edge	The proposed algorithm could contribute to in-depth epidemiologic investigation of infectious disease transmission to support targeted non pharmaceutical intervention policies for COVID-19 epidemic control.

		Association, (2021)	consolidation, and graph reduction.	
6	Social network analysis of COVID-19 transmission in Karnataka, India	S. Saraswathi, National Library of Medicine,20 2 0	Analyzed which area was most affected by COVID-19 by using the visualization tools like Cytoscape and Gephi, through a network and found out all the centrality measures for the data collected in India.	Bangalore had the highest number of cases recorded where more healthcare facilities were provided and priority was given to this region
7	Social network analysis of tourism data: A case study of quarantine decisions in COVID-19 pandemic	Fatma Altuntas, Serkan Altuntas, Turkay Dereli, International Journal of Information Management Data Insights (2022)	Social network analysis (SNA) based on tourism data to make the right quarantine decisions in the COVID-19 pandemic.	Quarantine authorities in each country can utilize SNA to make the right quarantine decisions to reduce the impact of the pandemic on tourism.

8	The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology	Kyra H. Grantz, Nature Communications (2020)	Analyzed the data collected using mobile phones to determine the spread of the COVID-19 pandemic. Non-pharmaceutical intervention (NPI) like social distancing was used as the primary control strategy during COVID-19. Analyzing how effectively the NPI is working to control the transmission of COVID-19 between people is a challenging task.	Call Detail Records(CDR) collected by the Mobile Phone operators which contain the timestamp and Global Positioning System(GPS) location of all subscribers is used to generate Origin-Destination matrices. This matrix helps to detect regular hotspots of attraction. The COVID-19 spread patterns analyzed using this matrix are used to inform future projections of disease risk and help in implementing travel restrictions. These patterns became the key features to detect the disease hotspots. This data can be further analyzed and put into decision-making for the restrictions of the areas which are detected as disease hotspots.
9	Constructing personal networks in light of COVID-19 containment measures	Emanuela Furfaro,Springer , 2021	The Family and Social Subjects (FSS) survey is analyzed. Due to epidemiological significance, contact tracing data helps to control the spread of disease in intervention settings	Includes rebuilding the ego-centered networks based on family relations, face-to-face contacts, and residential proximity of siblings, children, grandchildren, etc., to study social relationships

10	Network-based prediction of COVID-19 epidemic spreading in Italy	Clara Pizzuti, National Library of Medicine ,2020	If the person is infected with COVID-19, uploads the broadcasted chips which will be reflected in the diagnosis server, and whoever got in contact with the infected person will be easily identified using digital exposure notification by checking log data.,	The classic SIR model is extended to analyze the infection rates based on the changing protocols and measures followed by the government.
11	Network-inference-based prediction of the COVID-19 epidemic outbreak in the Chinese province Hubei	Bastian Prasse Applied Network Science, 2020	Network-Inference-Based Prediction Algorithm (NIPA) is used to predict the future existence of the COVID-19 pandemic in every city in Chinese province in Hubei.	Analysis is made using logistic regression curve that true fraction value of the infected persons is underestimated on the other hand true value is overestimated using Network Inference Based Algorithm
12	Bidirectional contact tracing could dramatically improve COVID-19 control	William J. Bradshaw Nature Communications	Stochastic branching process model is used to find our infected individuals and those who are infectees.	COVID-19 could be brought in control to a drastic extent using by introducing the bidirectional tracing model.

13.	Unlink the Link Between COVID-19 and 5G Networks: An NLP and SNA Based Approach	Mohammed Bahja and Ghazanfar Ali Safdar. *IEEE Access*, 2020	Latent Dirichlet Allocation models, Sentiment Analysis and SNA	For the analysis and classification of tweets, models including LDA, sentiment analysis, and social network analysis were utilized. Understanding the topic frequencies, interrelationships between topics, and geographical distribution of tweets enables policymakers to detect agencies and patterns in the dissemination of misinformation on COVID-19 and equips them with the knowledge to devise counterstrategies.
14.	Using Social Network Analysis to Identify Spatiotemporal Spread Patterns of COVID-19 around the World: Online Dashboard Development	Kyent-Yon Yie ,Tsair-Wei Chien,Yu-Tsen Yeh ,Willy Chou and Shih-Bin Su International Journal of Environmental Research and Public Health, 2021	Techniques used comprised three major stages: (1) SNA was used to classify the spread stages and occurrences of stationarity in the second wave of the COVID-19 crisis; (2) comparisons of model parameters (e.g., the location parameter or IP) were made to distinguish epidemic stages; and (3) an app was developed for understanding the transmission patterns of COVID-19 across countries/regions using the mathematical IRT model embedded in dashboards.	It was concluded that (1) the use of SNA for investigating the SSP is feasible; (2) the spread routes of COVID-19 from China to West Asia, Europe, North America, and South America are evident in three classifications (255, n = 51, 130, and 74 in stages); (3) Heilongjiang (China), Japan, Taiwan, and Qatar share a cluster with Hebei Province (China); and (4) a dashboard on Google Maps can be used to display the SSP of the COVID-19.

15.	Social network analysis methods for exploring SARS-CoV-2 contact tracing data	K. Nagarajan, M. Muniyandi Semantic Scholar ,2017	Using the centrality measures such as betweenness centrality, closeness centrality ,out degree ,the different levels of transmission is found and accordingly the infected individuals and the infectees are characterized.	Network component analysis revealed nineteen linked components made up of important patients that accounted for 26.95% of total patients (1959) and 68.74% of epidemiological connections in the network. Conclusions are hence made that Individual patient level differences in disease transmission could be measured using a social network analysis technique for SARS-CoV-2 contact tracking data.

DATASET AND TOOL USED

The dataset that we are analyzing is openly available in its csv format on Kaggle.

Our dataset consists of 4 columns named reporting user (ID), contact user (ID), contact start time and contact end time.

For our analysis, we have used the first 2 columns of our dataset for forming the network.

We used google colab to perform the metrics and for visualisation using python.

Proposed Work

- First of all, we will gather the contact tracing data of different people in a specific area and collect details such as their gender, age, whether they had come in contact with any covid positive patient before they got infected, whether they have any comorbidities or not, who all did they meet while they were covid positive and whether they had come in contact with any covid negative person after getting tested as covid positive.
- We will analyze the SARS-COV-2 outbreak size based on all the possible interventions and factors which impact this outbreak size the most, to reduce the number of cases.
- From the data that we collect, the factors that would have the maximum impact will be analyzed and considered as variables of our dataset. After that, we will apply different graph modeling and clustering techniques on our dataset. Choose the best of it, try to update it with new features and implement it in our analysis.
- We will be constructing a graph in which individuals will be considered as nodes and links will be considered as the infection relationships between them. It will be a directed graph in which the links will be directed from the covid positive patient who came in contact with a covid negative patient which resulted in the latter being infected with covid.
- The node with the highest degree centrality will be considered as the node responsible for the spread of the virus in a large scale. Betweeness centrality measure will be used to find which node was responsible for passing on the virus from one network to another. We will be clustering the nodes based on their level in the infection tree and analyzing the impact of each level on the spread of covid.
- We will be calculating the effective size of ego and then analyze it to find whether the graph has a high or a low constraint. If the graph has a low constraint, then we can infer that an ego is majorly responsible for the spread of the deadly virus. In the other case, we can infer that the both the ego and the alters are responsible for the spread of the deadly virus.
- We will be tracing the patterns in which the virus was spread in the network, in order to find a solution for curbing further spread of the virus. Homophily and heterophily will be calculated based on the gender and age to trace the patterns of the spread of virus.
- Furthermore, we are planning to do community network analysis, by checking inter-cluster and intra-cluster density. After making the complete graph consisting of covid positive as well as covid negative individuals, we will be predicting the spread of the virus to a particular node whose status is not know. By doing this, we can help in minimizing their contact with other people so that the spread of the virus can be contained.

- We also applied SIS and SIR models. The SIRS model extends the SIR and SIS models as two behavioral extremes where immunity is either permanent or nonexistent. An intermediate assumption is that immunity only lasts for a limited time, after which the individual becomes susceptible once more.

  SIR classifies the host within a population as susceptible (if previously exposed to disease), infected (if currently infected by the pathogen) or recovered (if they had successfully cleared the infection). The SIR system is an excellent illustration of a damped oscillator, which indicates that the inherent dynamics contain a strong oscillatory component, but the amplitude of these fluctuations decreases as the system reaches equilibrium. The fraction of infectivity oscillates with diminishing amplitude as it approaches equilibrium.

  There are two states in the SIS model: susceptible and infected. It permits nodes to modify their behavior over time, allowing them to revert to their previous state. In this model, steady state becomes necessary. It indicates that network nodes are infected at one rate and recovered at another. Establishing a steady state is necessary so that the rate of change in infection remains constant. A person can be infected multiple times throughout his or her lifetime without developing immunity. When susceptible nodes come into contact with an infected node, they instantly become infectious. The recovery from an infection is immediately followed by a return to the susceptible population.

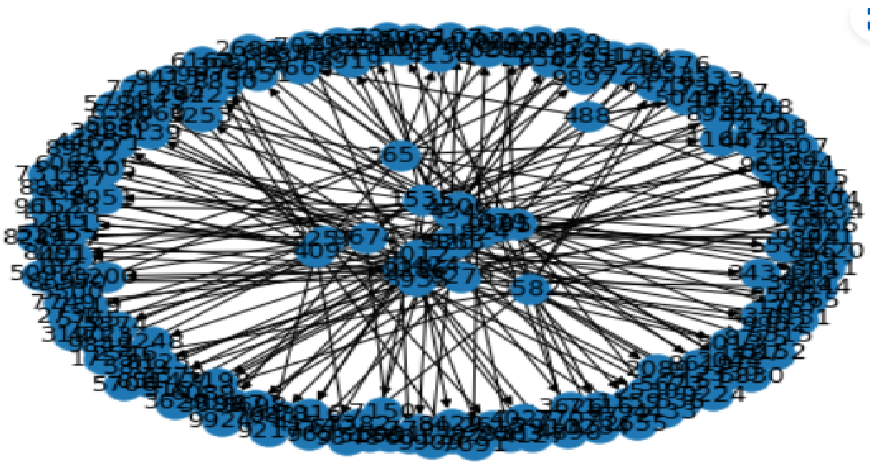
ALGORITHMS/TECHNIQUES DESCRIPTION

Construction of graph

Using the data from contacts.csv we constructed the graph using networkx package in python

```
import networkx as nx
G = nx.DiGraph()
with open('contacts.csv', 'r') as f:
    data = csv.reader(f)
    i=1
    headers = next(data)
    for row in tqdm(data):
        i=i+1
        if i==201:
            break
        G.add_node(row[0]) #superhero in first column
        G.add_node(row[1]) #superhero in second column
        if G.has_edge(row[0], row[1]):
            # edge already exists, increase weight by one
            G[row[0]][row[1]]['weight'] += 1
        else:
            # add new edge with weight 1
            G.add_edge(row[0], row[1], weight = 1)
```

```
nx.draw(G, with_labels=True)

plt.show()
```



Contact tracing data of different people is being collected in a specific area along with other details such as their gender, age, whether they had come in contact with any COVID-19 positive patient before they got infected, whether they have any comorbidities or not, whom all did they meet while they were COVID-19 positive and whether they had come in contact with any COVID-19 negative person after getting tested as COVID-19 positive. SARS-COV-2 outbreak size will be analyzed based on all the possible interventions and factors which impact this outbreak size the most, to reduce the number of cases. The contact tracing data set is converted into a network with nodes and edges representing persons and their interactions among them. Various SNA metrices are analyzed to identify the spread among them. The node with the highest degree centrality will be considered as the node responsible for the spread of the virus on a large scale. Betweenness centrality measure will be used to find which node was responsible for passing on the virus from one network to another. Clustering the nodes will be done based on their level in the infection tree and analyzing the impact of each level on the spread of COVID-19. Tracing the patterns will be done in which the virus was spread in the network, in order to find a solution for curbing further spread of the virus.

Social network analysis performance metrics

Various centrality measures like degree centrality, betweenness centrality, and eigenvector centrality were used to understand investigate covid-19 transmission among contacts.

Density

Density is defined as the number of connections a participant has divided by the total possible connections a participant could have. For example, if there are 20 people participating each person could potentially connect to 19 other people.

$$UndirectedNetworkDensity = \frac{TotalEdges}{TotalPossibleEdges} = \frac{Cardinality}{Size} = \frac{m}{n(n-1)/2}$$

$$DirectedNetworkDensity = \frac{TotalEdges}{TotalPossibleEdges} = \frac{Cardinality}{Size} = \frac{m}{n(n-1)}$$

Degree centrality

Degree centrality measures the number of ties for each node in the network. The centrality increases with increase in the number of neighbour nodes directly connected to each node. Degree centrality is computed as shown below. A high degree centrality means that the node has many ties directed to it

$$C_{\mathrm{D}}(i) = \sum_{j=1}^{n} a_{ij}$$

Betweeness Centrality

Betweenness centrality measures the bridge role a particular node plays in the network. It calculates a node's centrality based on the number of shortest paths for all pairs of vertices that pass through that node. In simple terms, betweenness centrality is the sum of the shortest paths that pass through a node.

$$C_{\mathrm{B}}(i) = \sum_{j<k}^{n} \frac{g_{jk}(i)}{g_{jk}}$$

In the formula, "gjk" is the number of shortest paths between j and k and "gjk(i)" is the frequency of times the link passed through i.

Closeness centrality

Closeness centrality denotes the node's reach in a given network.

The connection of two nodes is the least number of hops necessary to reach one node from the other in undirected and unweighted graphs.

On the other hand, in networks that are weighted and directed, the distance between two nodes is affected by direction and magnitude.

The closeness centrality of a node u is defined as follows where n is total number of nodes and d(u, v) is the shortest distance between two nodes u and v

$$(u) = \frac{n-1}{\sum_{\forall v} d(u, v)}$$

Eigenvector centrality:

Eigenvector centrality (also called eigen centrality) is a measure of the influence of a node in a network. It indicates which of the nodes have influenced the maximum other nodes.

The eigenvector centrality network metric takes into consideration not only how many connections a vertex has (i.e. its degree), but also the degree of the vertices that it is connected to.

This measure describes the behaviour of a node in the network by allocating relative scores to all nodes in the network based on connections with the high scoring nodes.

Eigenvector defines a node’s importance in the network is based on the number of connections to significant nodes.

Besides that, the eigenvector centrality can tell us the most influential nodes in a network. The most influential/important node is a node with highest eigenvector value among the other nodes in a network (Barbasi and Albert 1999)

Agglomerative Hierarchical Clustering

The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It’s also known as *AGNES* (*Agglomerative Nesting*). The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named *dendrogram*.

Steps to agglomerative hierarchical clustering

1. Preparing the data
2. Computing (dis)similarity information between every pair of objects in the data set.
3. Using linkage function to group objects into hierarchical cluster tree, based on the distance information generated at step 1. Objects/clusters that are in close proximity are linked together using the linkage function.

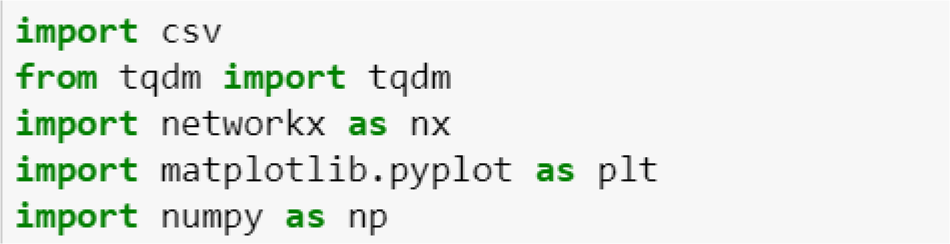
4. Determining where to cut the hierarchical tree into clusters. This creates a partition of the data.

Analysis

The information conveyed by the network formed can be understood from simple measures such as Centrality measures, Page Rank Algorithms, Network Density as well as Clustering mechanisms. At present we have completed the implementation and application of centrality measures and page rank algorithms on our network. Centrality Measures allows us to pinpoint the most important nodes of a Graph. This essentially helps us to identify :

- Influential persons in a Social Network.
- Important persons in the Web

The packages which are required for analysing this study are imported in the python.

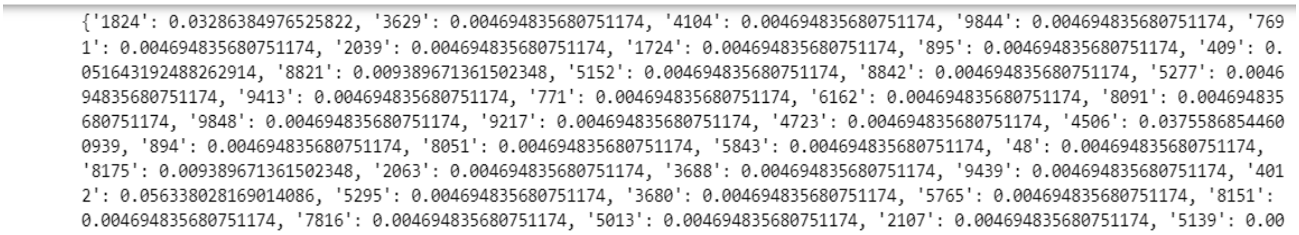


```
import csv
from tqdm import tqdm
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
```

Using Histograms,.in this study,we noticed that most of the nodes in the graph has degree centrality in the range of 0 to 0.01.Degree centrality helps us to understand indegree and outdegree of all the nodes present in our network.

Calculating Degree Centrality

1. **deg_centrality = nx.degree_centrality(G)**
2. **print(deg_centrality)**

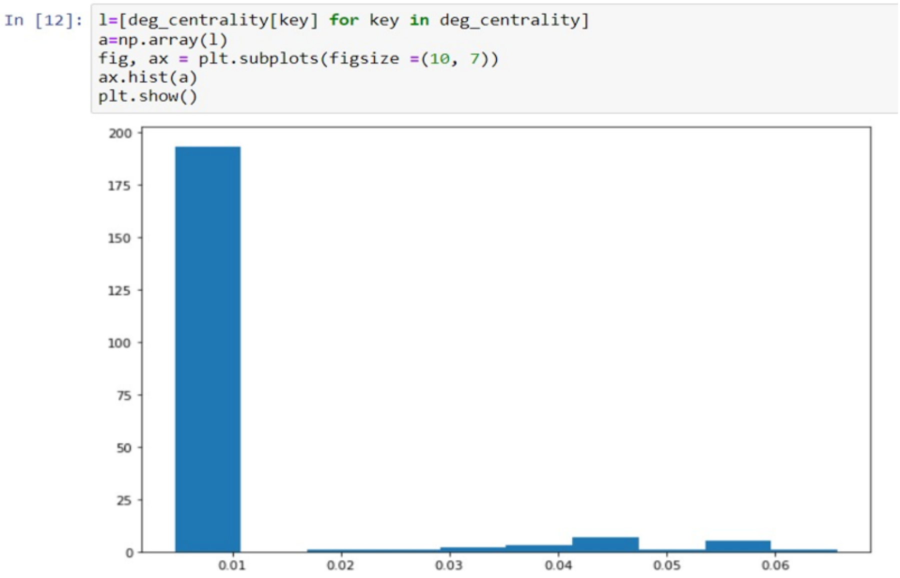


```
{'1824': 0.03286384976525822, '3629': 0.004694835680751174, '4104': 0.004694835680751174, '9844': 0.004694835680751174, '769
1': 0.004694835680751174, '2039': 0.004694835680751174, '1724': 0.004694835680751174, '895': 0.004694835680751174, '409': 0.
051643192488262914, '8821': 0.009389671361502348, '5152': 0.004694835680751174, '8842': 0.004694835680751174, '5277': 0.0046
94835680751174, '9413': 0.004694835680751174, '771': 0.004694835680751174, '6162': 0.004694835680751174, '8091': 0.004694835
680751174, '9848': 0.004694835680751174, '9217': 0.004694835680751174, '4723': 0.004694835680751174, '4506': 0.0375586854460
0939, '894': 0.004694835680751174, '8051': 0.004694835680751174, '5843': 0.004694835680751174, '48': 0.004694835680751174,
'8175': 0.009389671361502348, '2063': 0.004694835680751174, '3688': 0.004694835680751174, '9439': 0.004694835680751174, '401
2': 0.056338028169014086, '5295': 0.004694835680751174, '3680': 0.004694835680751174, '5765': 0.004694835680751174, '8151':
0.004694835680751174, '7816': 0.004694835680751174, '5013': 0.004694835680751174, '2107': 0.004694835680751174, '5139': 0.00
```

Visualising Degree Centrality

1. **l=[deg_centrality[key] for key in deg_centrality]**

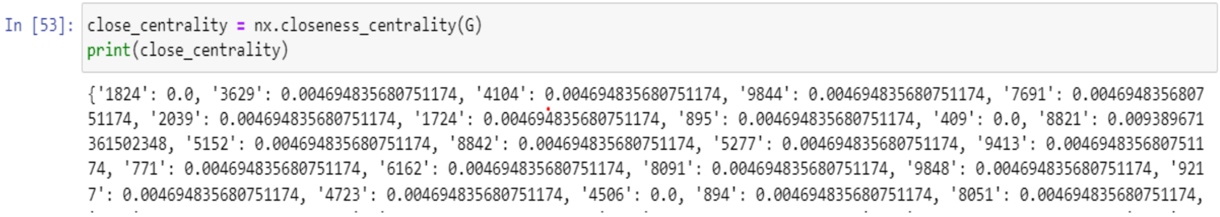
2. **a=np.array(l)**
3. **fig, ax = plt.subplots(figsize =(10, 7))**
4. **ax.hist(a)**
5. **plt.show()**



```
In [12]: l=[deg_centrality[key] for key in deg_centrality]
         a=np.array(l)
         fig, ax = plt.subplots(figsize =(10, 7))
         ax.hist(a)
         plt.show()
```

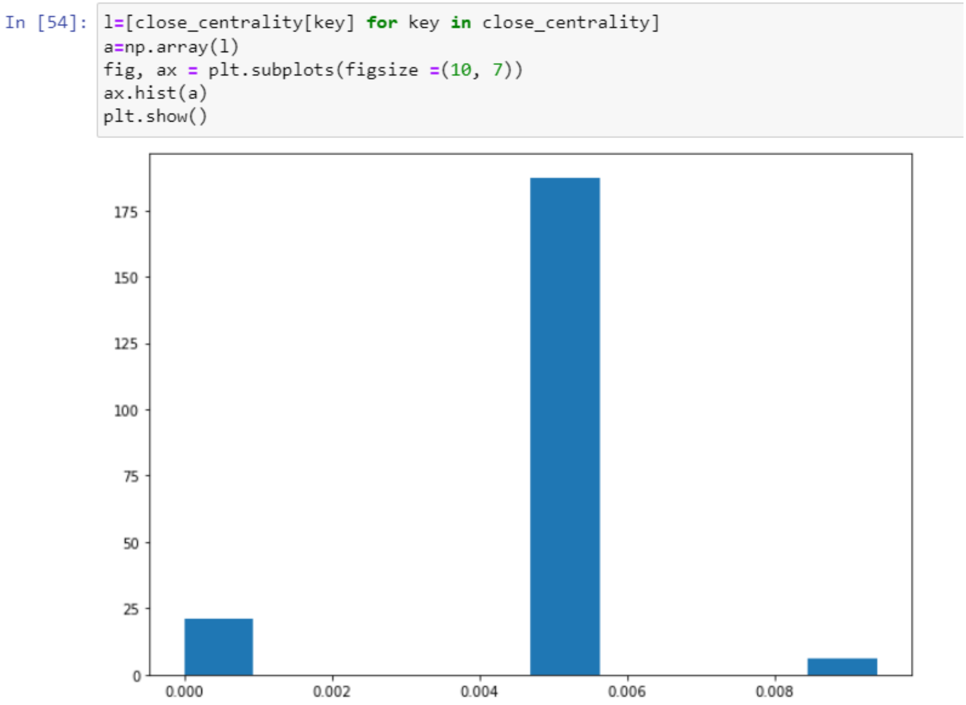
Calculating ClosenessCentrality

1. **close_centrality = nx.closeness_centrality(G)**
2. **print(close_centrality)**



```
In [53]: close_centrality = nx.closeness_centrality(G)
         print(close_centrality)

{'1824': 0.0, '3629': 0.004694835680751174, '4104': 0.004694835680751174, '9844': 0.004694835680751174, '7691': 0.0046948356807
51174, '2039': 0.004694835680751174, '1724': 0.004694835680751174, '895': 0.004694835680751174, '409': 0.0, '8821': 0.009389671
361502348, '5152': 0.004694835680751174, '8842': 0.004694835680751174, '5277': 0.004694835680751174, '9413': 0.0046948356807511
74, '771': 0.004694835680751174, '6162': 0.004694835680751174, '8091': 0.004694835680751174, '9848': 0.004694835680751174, '921
7': 0.004694835680751174, '4723': 0.004694835680751174, '4506': 0.0, '894': 0.004694835680751174, '8051': 0.004694835680751174,
```

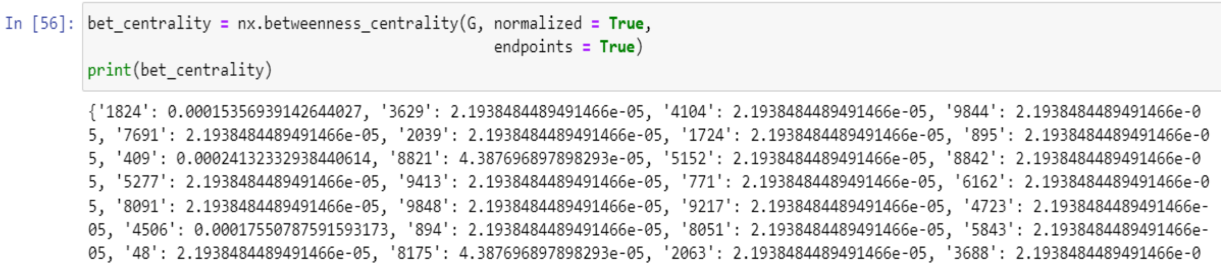
1. **l=[close_centrality[key] for key in close_centrality]**
2. **a=np.array(l)**
3. **fig, ax = plt.subplots(figsize =(10, 7))**
4. **ax.hist(a)**
5. **plt.show()**



```
In [54]: l=[close_centrality[key] for key in close_centrality]
         a=np.array(l)
         fig, ax = plt.subplots(figsize =(10, 7))
         ax.hist(a)
         plt.show()
```

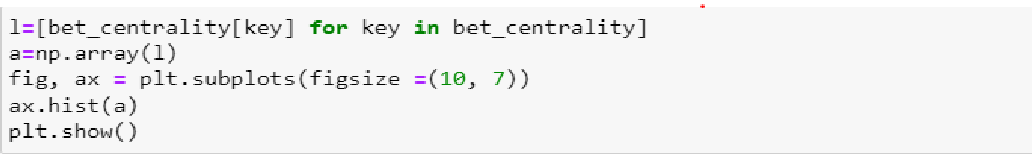
Calculating betweeness Centrality

1. **bet_centrality = nx.betweenness_centrality(G, normalized = True, endpoints = True)**
2. **print(bet_centrality)**

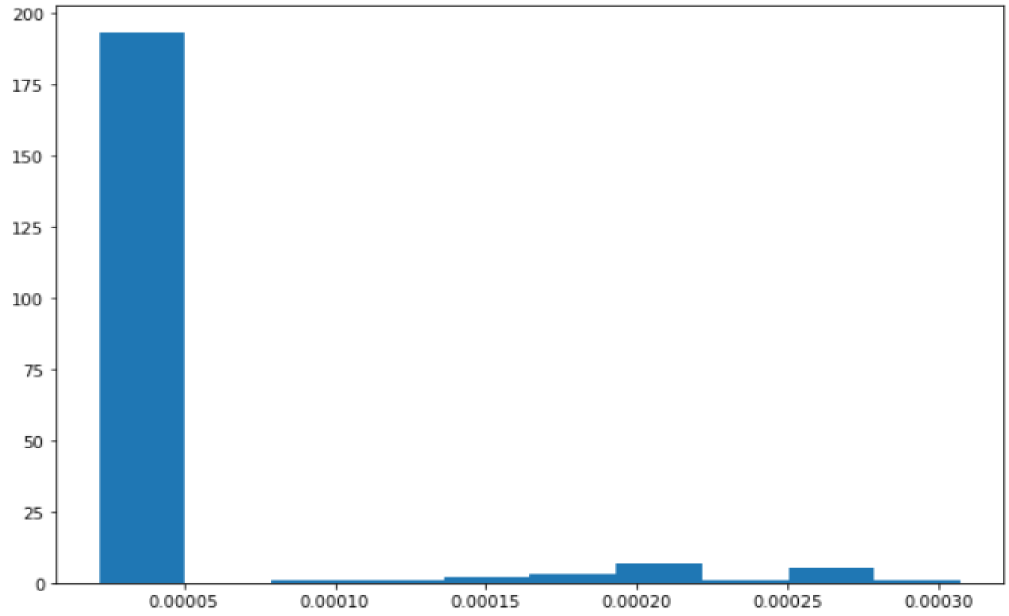


```
In [56]: bet_centrality = nx.betweenness_centrality(G, normalized = True,
                                                     endpoints = True)
         print(bet_centrality)

{'1824': 0.00015356939142644027, '3629': 2.1938484489491466e-05, '4104': 2.1938484489491466e-05, '9844': 2.1938484489491466e-0
5, '7691': 2.1938484489491466e-05, '2039': 2.1938484489491466e-05, '1724': 2.1938484489491466e-05, '895': 2.1938484489491466e-0
5, '409': 0.00024132332938440614, '8821': 4.387696897898293e-05, '5152': 2.1938484489491466e-05, '8842': 2.1938484489491466e-0
5, '5277': 2.1938484489491466e-05, '9413': 2.1938484489491466e-05, '771': 2.1938484489491466e-05, '6162': 2.1938484489491466e-0
5, '8091': 2.1938484489491466e-05, '9848': 2.1938484489491466e-05, '9217': 2.1938484489491466e-05, '4723': 2.1938484489491466e-
05, '4506': 0.00017550787591593173, '894': 2.1938484489491466e-05, '8051': 2.1938484489491466e-05, '5843': 2.1938484489491466e-
05, '48': 2.1938484489491466e-05, '8175': 4.387696897898293e-05, '2063': 2.1938484489491466e-05, '3688': 2.1938484489491466e-0
```

1. **l=[bet_centrality[key] for key in bet_centrality]**
2. **a=np.array(l)**
3. **fig, ax = plt.subplots(figsize =(10, 7))**
4. **ax.hist(a)**
5. **plt.show()**



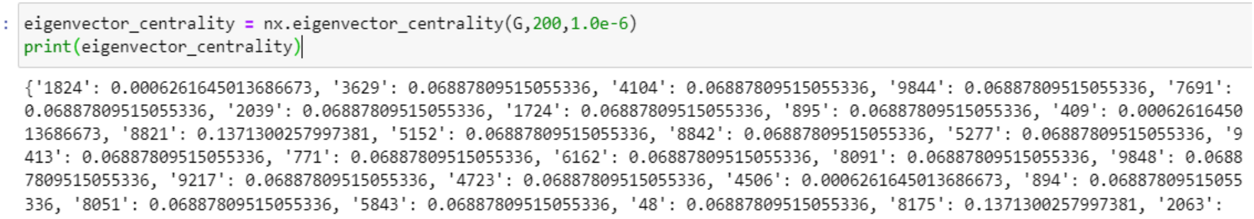
```
l=[bet_centrality[key] for key in bet_centrality]
a=np.array(l)
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a)
plt.show()
```



In eigen vector centrality we analyzed that nodes '8821', '8175', '4017', '6089', '921', '8697' have highest influence in the network and we drew a conclusion that if these nodes are infected then most of the nodes in the graph will be infected.

Calculating Eigen Value Centrality

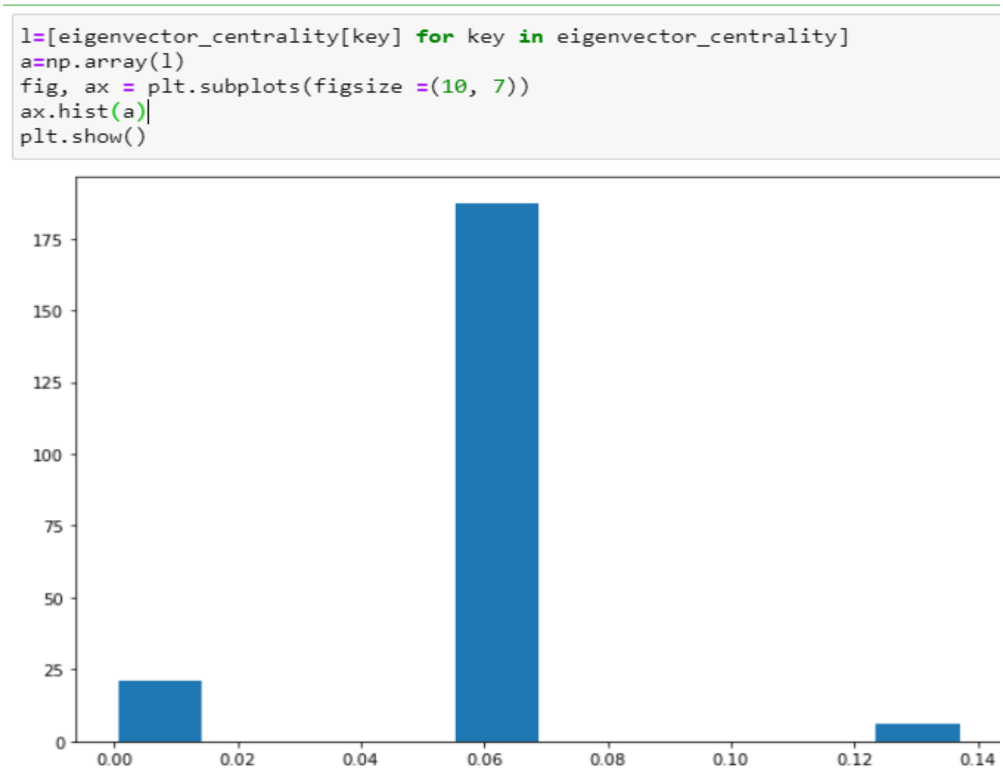
1. **eigenvector_centrality = nx.eigenvector_centrality(G,200,1.0e-6)**
2. **print(eigenvector_centrality)**



```
: eigenvector_centrality = nx.eigenvector_centrality(G,200,1.0e-6)
  print(eigenvector_centrality)
```

```
{'1824': 0.0006261645013686673, '3629': 0.06887809515055336, '4104': 0.06887809515055336, '9844': 0.06887809515055336, '7691':
0.06887809515055336, '2039': 0.06887809515055336, '1724': 0.06887809515055336, '895': 0.06887809515055336, '409': 0.00062616450
13686673, '8821': 0.1371300257997381, '5152': 0.06887809515055336, '8842': 0.06887809515055336, '5277': 0.06887809515055336, '9
413': 0.06887809515055336, '771': 0.06887809515055336, '6162': 0.06887809515055336, '8091': 0.06887809515055336, '9848': 0.0688
7809515055336, '9217': 0.06887809515055336, '4723': 0.06887809515055336, '4506': 0.0006261645013686673, '894': 0.06887809515055
336, '8051': 0.06887809515055336, '5843': 0.06887809515055336, '48': 0.06887809515055336, '8175': 0.1371300257997381, '2063':
```

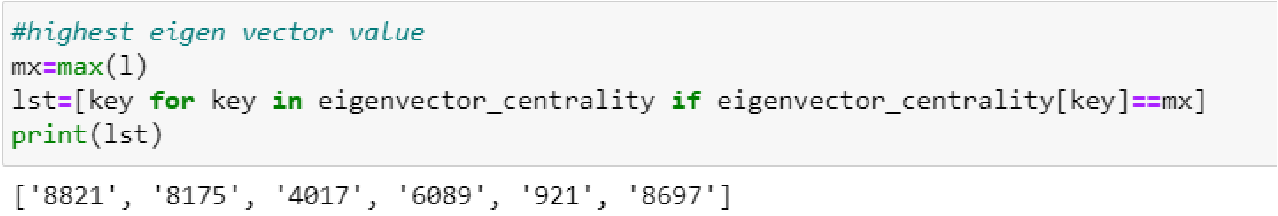
1. **l=[eigenvector_centrality[key] for key in eigenvector_centrality]**
2. **a=np.array(l)**
3. **fig, ax = plt.subplots(figsize =(10, 7))**
4. **ax.hist(a)**
5. **plt.show()**



```
l=[eigenvector_centrality[key] for key in eigenvector_centrality]
a=np.array(l)
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a)
plt.show()
```

Highest Eigen Value

1. **mx=max(l)**
2. **lst=[key for key in eigenvector_centrality if eigenvector_centrality[key]==mx]**
3. **print(lst)**



```
#highest eigen vector value
mx=max(l)
lst=[key for key in eigenvector_centrality if eigenvector_centrality[key]==mx]
print(lst)
```

```
['8821', '8175', '4017', '6089', '921', '8697']
```

Visualization:

import pandas as pd

df=pd.read_csv("contacts.csv")

covid=nx.from_pandas_edgelist(df,source="reporting_user",target="contact_user")

plt.figure(figsize=(100,100))

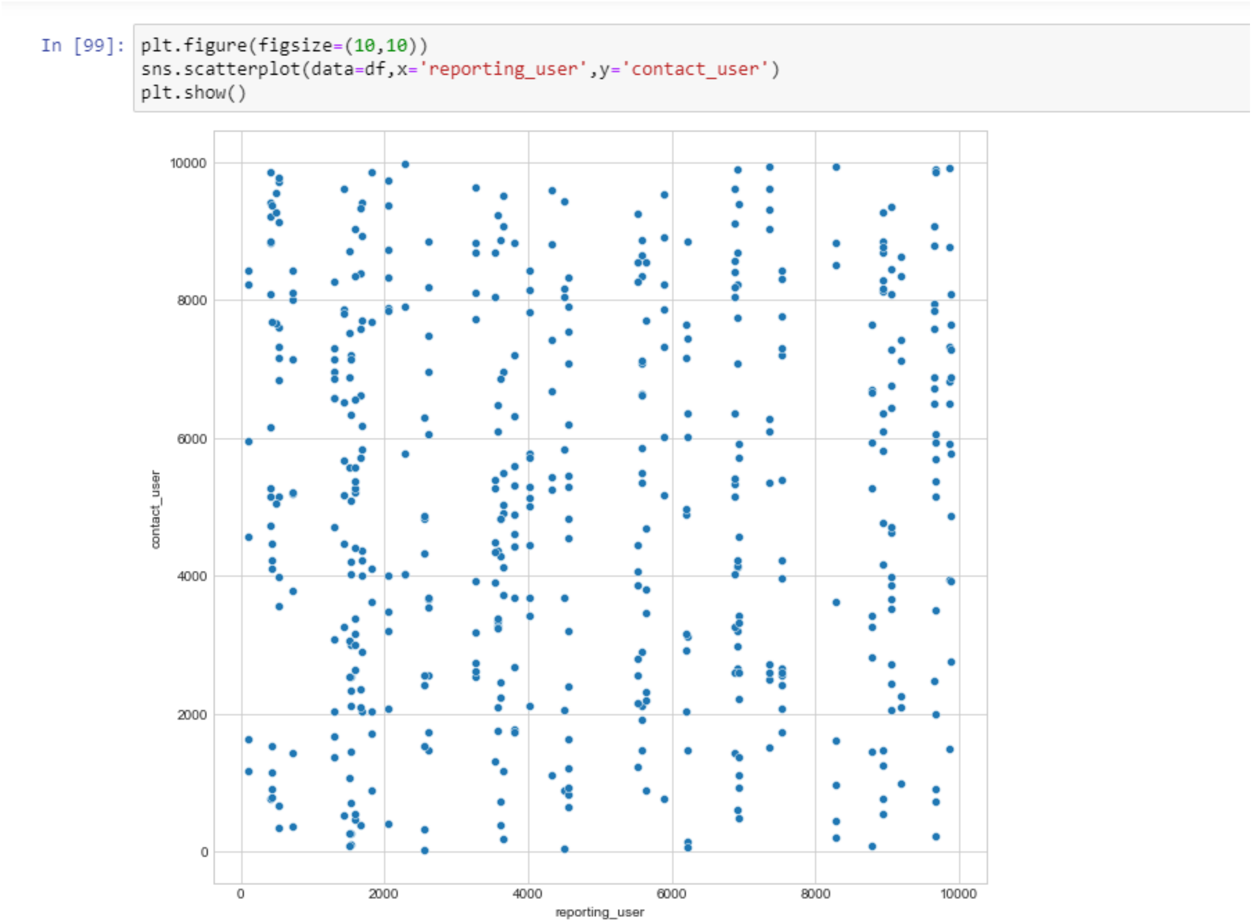
nx.draw_networkx(covid)

plt.show()

plt.figure(figsize=(10,10))

sns.scatterplot(data=df,x='reporting_user',y='contact_user')

plt.show()



```
In [99]: plt.figure(figsize=(10,10))
         sns.scatterplot(data=df,x='reporting_user',y='contact_user')
         plt.show()
```

Density:

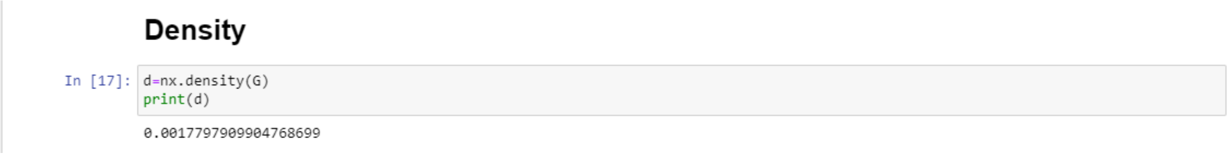
Density captures how many edges there are in a network divided by the total possible number of edges.It is a measure of network health and effectiveness. The density of a network is the fraction between 0 and 1 that tells us what portion of all possible edges are actually realized in the network.

The density that we obtained for our network is 0.001779. Since the density of a graph is measure of connectedness of nodes in the graph.It is often viewed as a metric of efficiency since a high-density network has more connections and thus better exploits the total number of possible interactions.

Therefore the network is not a high density network.

d=nx.density(G)

print(d)



Density

```
In [17]: d=nx.density(G)
         print(d)

         0.0017797909904768699
```

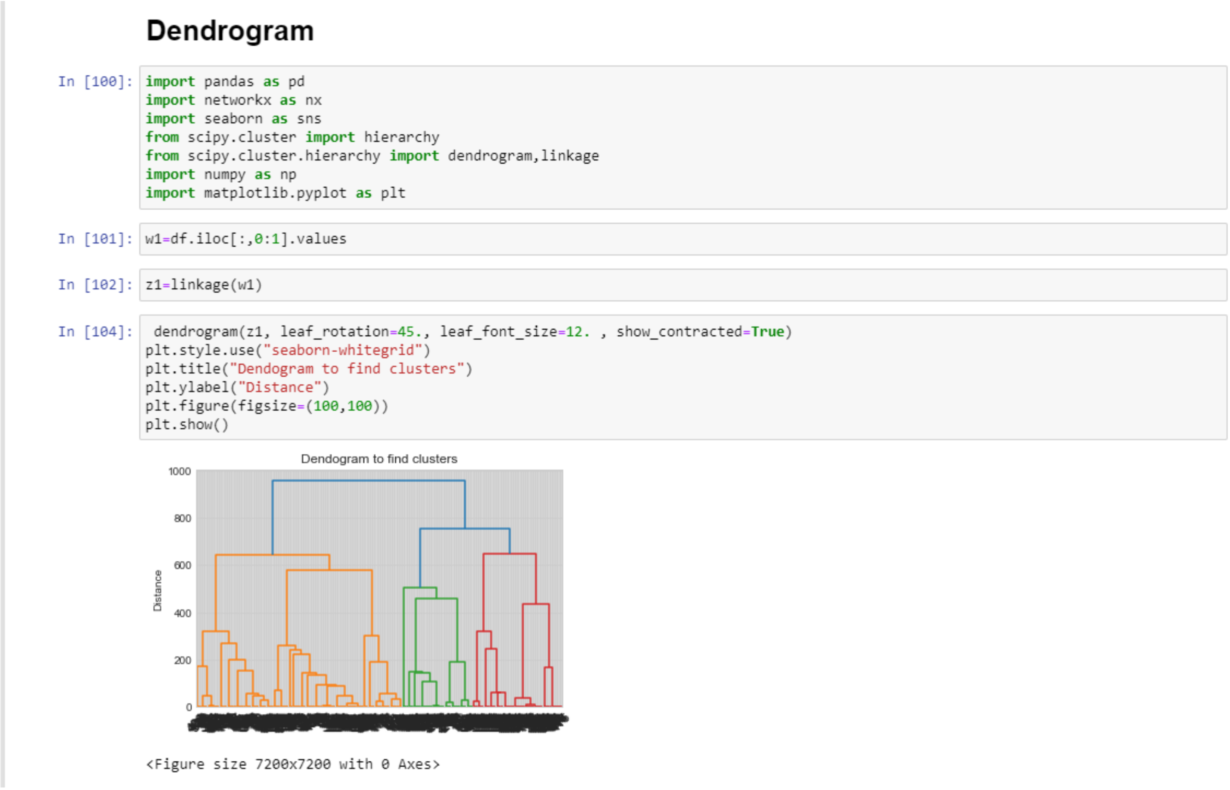
Clustering:

Clustering is the task of assigning a set of objects to groups (also called classes or categories) so that the objects in the same cluster are more similar (according to a predefined property) to each other than to those in other clusters

Agglomerative Hierarchical Clustering(Dendogram):

Agglomerative Clustering is a type of hierarchical clustering algorithm. It is an unsupervised machine learning technique that divides the population into several clusters such that data points in the same cluster are more similar and data points in different clusters are dissimilar.

```
import pandas as pd

import networkx as nx

import seaborn as sns

from scipy.cluster import hierarchy

from scipy.cluster.hierarchy import dendrogram,linkage

import numpy as np

import matplotlib.pyplot as plt

w1=df.iloc[:,0:1].values

z1=linkage(w1)

 dendrogram(z1, leaf_rotation=45., leaf_font_size=12. , show_contracted=True)

plt.style.use("seaborn-whitegrid")

plt.title("Dendogram to find clusters")

plt.ylabel("Distance")

plt.figure(figsize=(100,100))

plt.show()
```



Dendrogram

```
In [100]: import pandas as pd
          import networkx as nx
          import seaborn as sns
          from scipy.cluster import hierarchy
          from scipy.cluster.hierarchy import dendrogram,linkage
          import numpy as np
          import matplotlib.pyplot as plt
```

```
In [101]: w1=df.iloc[:,0:1].values
```

```
In [102]: z1=linkage(w1)
```

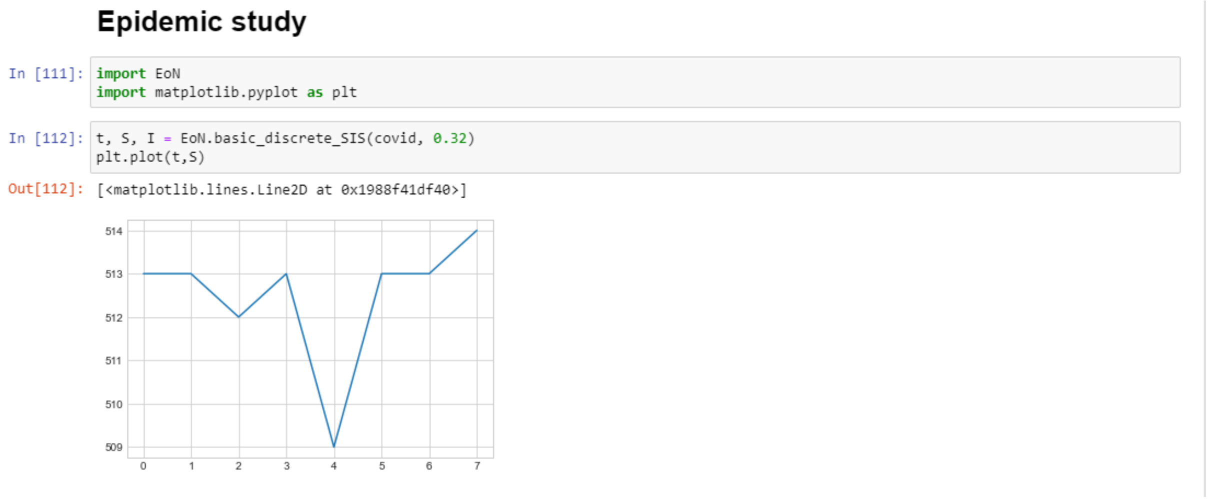
```
In [104]:  dendrogram(z1, leaf_rotation=45., leaf_font_size=12. , show_contracted=True)
          plt.style.use("seaborn-whitegrid")
          plt.title("Dendogram to find clusters")
          plt.ylabel("Distance")
          plt.figure(figsize=(100,100))
          plt.show()
```

```
<Figure size 7200x7200 with 0 Axes>
```

Epidemic Study:

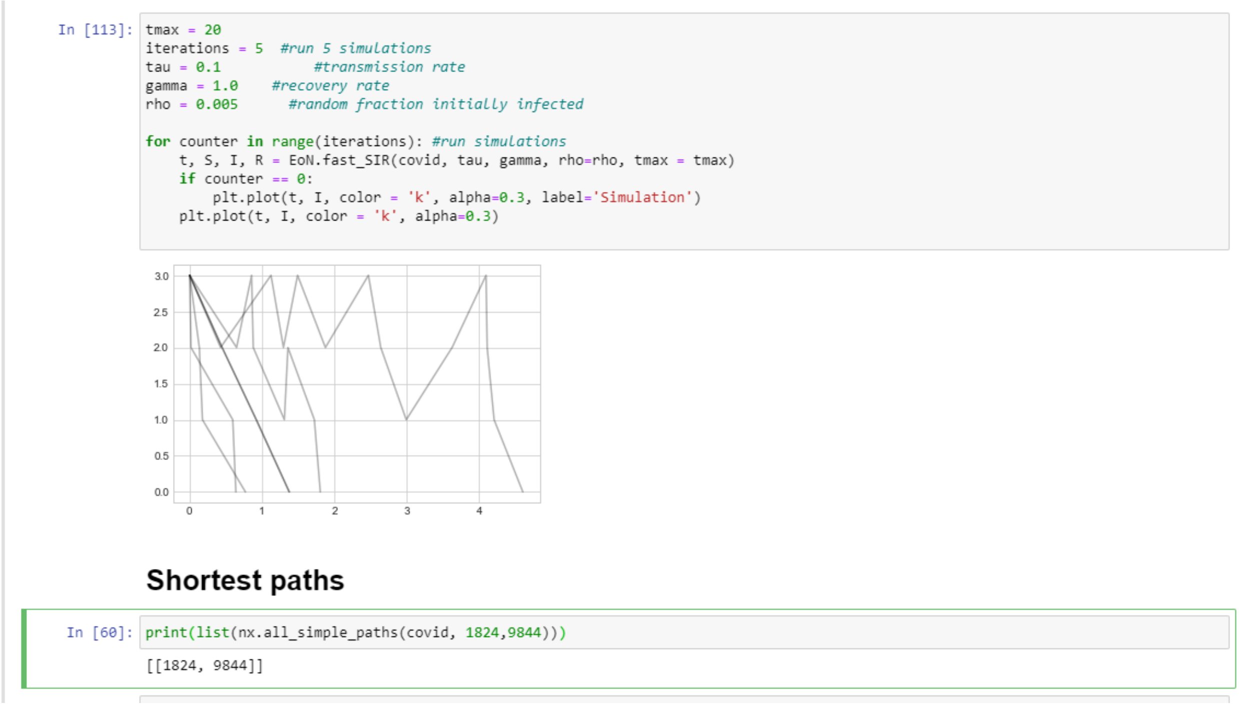
Studies of epidemics on networks aim to explain the spread of an epidemic in a society or between societies via interactions through the nodes of a network. Current human mobility patterns are mostly long distance travels, as hopping between nodes of a graph.

```
import EoN

import matplotlib.pyplot as plt

t, S, I = EoN.basic_discrete_SIS(covid, 0.32)

plt.plot(t,S)

tmax = 20

iterations = 5  #run 5 simulations

tau = 0.1         #transmission rate

gamma = 1.0    #recovery rate

rho = 0.005      #random fraction initially infected


for counter in range(iterations): #run simulations

    t, S, I, R = EoN.fast_SIR(covid, tau, gamma, rho=rho, tmax = tmax)

    if counter == 0:
        plt.plot(t, I, color = 'k', alpha=0.3, label='Simulation')
    plt.plot(t, I, color = 'k', alpha=0.3)
print(list(nx.all_simple_paths(covid, 1824,9844)))
```



Epidemic study

```
In [111]: import EoN
          import matplotlib.pyplot as plt

In [112]: t, S, I = EoN.basic_discrete_SIS(covid, 0.32)
          plt.plot(t,S)

Out[112]: [<matplotlib.lines.Line2D at 0x1988f41df40>]
```



```
In [113]: tmax = 20
          iterations = 5  #run 5 simulations
          tau = 0.1           #transmission rate
          gamma = 1.0    #recovery rate
          rho = 0.005      #random fraction initially infected

          for counter in range(iterations): #run simulations
              t, S, I, R = EoN.fast_SIR(covid, tau, gamma, rho=rho, tmax = tmax)
              if counter == 0:
                  plt.plot(t, I, color = 'k', alpha=0.3, label='Simulation')
              plt.plot(t, I, color = 'k', alpha=0.3)
```

Shortest paths

```
In [60]: print(list(nx.all_simple_paths(covid, 1824,9844)))

         [[1824, 9844]]
```

From the above epidemic study we have analysed that SIS graph is increasing graph which denotes that the once a person is susceptible to COVID 19 Disease then there is a possibility that the person infects the people surrounding him and is again susceptible to COVID 19 Disease .Whereas in the SIR graph once the person is susceptible and then infected with COVID 19 there is a possibility of the person benign recovered not being susceptible again

Results and Discussion

In this paper, we have implemented the following centrality measures -Degree Centrality, Betweenness Centrality, Closeness Centrality and Eigen Value Centrality. Using Histograms in this study, we noticed that most of the nodes in the graph have degree centrality in the range of 0 to 0.01. The average shortest distance between two nodes of our network is between 0.004 to 0.006 for majority of the pair of nodes. In eigen vector centrality we analyzed that nodes '8821', '8175', '4017', '6089', '921', '8697' have highest influence in the network and we drew a conclusion that if these nodes are infected then most of the nodes in the graph will be infected.

SARS-COV-2 heterogeneity can be captured by analyzing transmission at the individual patient level the contact tracing data from a network standpoint. The method can assist in identifying the key individual patients and components that could assist the public. Health implementers should concentrate their contact tracing efforts. Network metrics and graphical tools could be useful to add to the existing contact tracing indicators The potential use of network analysis could aid in the investigation. Large amounts of contact tracing data to detect heterogeneity, which could aid in the implementation of contact tracing activities in a more informed way.

Conclusion

In this project, we utilized all available metrics and centrality measures to analyse the graph and comprehend the spread of COVID19 among the contacts in the given network. We compared 215 and 514 individuals, and discovered that as network density increases, graph analysis is simplified and network analysis trends are more easily discerned. We have determined that the density of the graph is not extremely dense. We have also conducted an epidemic study and developed the SIS and SIR models, which aid in determining which nodes will be susceptible to and infected by the COVID 19 virus. Therefore, we can conclude that our network is not dense and that the various metrics and centrality measures applied to the network allow us to comprehend the pattern of COVID 19's spread.

GITHUB REPOSITORY LINK

https://github.com/Sheral18/Web-Mining-CSE-3024---Digital-Assignment-1
https://github.com/anjalijain2002/Web_mining_SNA

REFERENCES

[1] Nagarajan, K., Muniyandi, M., Palani, B. et al. Social network analysis methods for exploring SARS-CoV-2 contact tracing data. BMC Med Res Methodol 20, 233 (2020). https://doi.org/10.1186/s12874-020-01119-3

[2] Pung, R., Firth, J.A., Spurgin, L.G. et al. Using high-resolution contact networks to evaluate SARS-CoV-2 transmission and control in large-scale multi-day events. Nat Commun 13, 1956 (2022). https://doi.org/10.1038/s41467-022-29522-y

[3] Jo, W., Chang, D., You, M. et al. A social network analysis of the spread of COVID-19 in South Korea and policy implications. Sci Rep 11, 8581 (2021). https://doi.org/10.1038/s41598-021-87837-0

[4] Chayanon Phucharoen, Nichapat Sangkaew, Kristina Stosic, The characteristics of COVID-19 transmission from case to high-risk contact, a statistical analysis from contact tracing data, EClinicalMedicine, Volume 27, 2020, 100543, ISSN 2589-5370, https://doi.org/10.1016/j.eclinm.2020.100543.

[5] Tsz Ho Kwan, Ngai Sze Wong, Eng-Kiong Yeoh, Shui Shan Lee, Mining relationships between transmission clusters from contact tracing data: An application for investigating COVID-19 outbreak, Journal of the American Medical Informatics Association, Volume 28,
Issue 11, November 2021, Pages 2385–2392, https://doi.org/10.1093/jamia/ocab175

[6] Sakranaik, Saraswathi & Mukhopadhyay, Amita & Shah, Het & Ranganath, T. (2020). Social Network Analysis of COVID-19 Transmission in Karnataka, India. Epidemiology and infection. 148. 1-30. https://doi.org/10.1017/S095026882000223X.

[7] Fatma Altuntas, Serkan Altuntas, Turkay Dereli, Social network analysis of tourism data: A case study of quarantine decisions in COVID-19 pandemic, International Journal of Information Management Data Insights,Volume 2, Issue 2, 2022, 100108, ISSN 2667-0968, https://doi.org/10.1016/j.jjimei.2022.100108.

[8] Grantz, K.H., Meredith, H.R., Cummings, D.A.T. et al. The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology. Nat Commun 11, 4961 (2020). https://doi.org/10.1038/s41467-020-18190-5

[9] Furfaro, E., Rivellini, G., Pelle, E. et al. Constructing personal networks in light of COVID-19 containment measures. Genus 77, 17 (2021). https://doi.org/10.1186/s41118-021-00128-4

[10] Pizzuti, C., Socievole, A., Prasse, B. et al. Network-based prediction of COVID-19 epidemic spreading in Italy. Appl Netw Sci 5, 91 (2020). https://doi.org/10.1007/s41109-020-00333-8

[11] Prasse, B., Achterberg, M.A., Ma, L. et al. Network-inference-based prediction of the COVID-19 epidemic outbreak in the Chinese province Hubei. Appl Netw Sci 5, 35 (2020). https://doi.org/10.1007/s41109-020-00274-2

[12] Bradshaw, W.J., Alley, E.C., Huggins, J.H. et al. Bidirectional contact tracing could dramatically improve COVID-19 control. Nat Commun 12, 232 (2021). https://doi.org/10.1038/s41467-020-20325-7

[13] M. Bahja and G. A. Safdar, Unlink the Link Between COVID-19 and 5G Networks: An NLP and SNA Based Approach, in *IEEE Access*, vol. 8, pp. 209127-209137, 2020, doi: 10.1109/ACCESS.2020.3039168.

[14] Kyent-Yon Yie ,Tsair-Wei Chien,Yu-Tsen Yeh ,Willy Chou and Shih-Bin Su, Using Social Network Analysis to Identify Spatiotemporal Spread Patterns of COVID-19 around the World: Online Dashboard Development, International Journal of Environmental Research and Public Health (2021), 18(5), 2461; https://doi.org/10.3390/ijerph18052461

[15] Nagarajan K, Das B. Tuberculosis and Social Networks:A Narrative Review on How Social Network Data and Metrics Help Explain Tuberculosis Transmission. Curr Sci. 2019;116:1068–80. https://doi.org/10.18520/cs/v116/i7/1068-1080