Exploring Vulnerabilities in the Central Chilean Power Grid: A Comprehensive Report

MA214: Network Analysis

Assignment 2: Group report

Registration numbers:

2312042, 2315586, 2318265, 2316207, 2316982, 2312554

Person submitting this file: 2316207

Word count: 3766

(Please note: Suggested word count was 3000 for a group of up to 5 members. As this is a larger group, the word count is higher to allow all members to contribute)

Abstract:

The power grid infrastructure serves as a foundational pillar for modern civilisation, yet it remains inherently vulnerable. This research report explores the vulnerabilities of the Central Chilean Power Grid (CCPG), an essential component of Chile's infrastructure that is critically positioned atop two tectonic plates, rendering it susceptible to natural disasters. Advanced network analysis methods, such as betweenness centrality, closeness centrality, clustering coefficient, and modularity, have been employed to identify key vulnerable elements within the CCPG's network structure. In this study, critical nodes within the power grid network—specifically nodes 105, 178, 147, and 108—have been pinpointed due to their ranking within the top 10 for both closeness centrality and betweenness centrality, underscoring their pivotal role in the network's overall functionality. The analysis reveals that these key nodes are integral to maintaining the network's stability and, by extension, ensuring the safety and well-being of the infrastructure reliant on it. Given the network's heavy reliance on the performance of these nodes, it becomes imperative for policymakers to prioritise their protection and optimisation to avoid power blackouts. Additionally, the authors propose pathways for further research, aimed at enhancing the resilience and efficiency of the power grid infrastructure, including a transition to renewable energy for increased resilience.

Introduction

Electricity, which is transferred through power grids, is an essential part of our daily lives. We often take this power source for granted and only realise its importance when we experience a power outage. Electricity is used for a multitude of purposes such as heating, cooking, refrigeration, lighting, and computation. Without it, our lives would become difficult and inconvenient [1].

This assignment aims to analyse the infrastructure of power grid networks, with a focus on the Central Chilean power grid (CCPG). To support this aim, a comprehensive literature review is provided, followed by an in-depth analysis of the CCPG. Chile, located in South America, was chosen as the case study for this report as it is prone to natural disasters, posing a threat to the stability and security of its power systems. The country is well-known for its frequent seismic and volcanic activity, emphasising the necessity of having a dependable power grid in place [2][3][4].

It is suggested that if only a small percentage (4%) of the total nodes with high load in a power grid were to collapse, the connectivity would decrease significantly by 60%. This highlights the crucial role played by such nodes, and their importance in maintaining the stability of the power grid. Therefore, critical nodes with high load should be identified and protected [5][6].

Measures such as betweenness and closeness centrality, clustering coefficients, and modularity can provide insights into the vulnerabilities within the power grid and highlight potential areas of concern [7][8].

This paper identifies potentially critical nodes based on their closeness centrality (CC) and betweenness centrality (BC) measures. The authors argue, these nodes are crucial to the optimal function of the network and should, therefore, be considered by policymakers when ensuring appropriate contingency plans are in place in case of high impact, low probability events such as natural disasters. Additionally, likely network vulnerabilities were highlighted using cluster coefficient and modularity analysis to further support this aim. The results are presented in the analysis section.

A discussion of the results of the analysis and their implication follows. Finally, concluding remarks and a brief overview of the limitations of this report, as well as areas for future results are discussed.

Literature review

The research analysing power grids focuses on preventing large-scale blackouts and ensuring the stability of high-voltage transmission systems. To make it easier to analyse power grids, they are usually transformed into network structures through aggregations and simplifications, producing convenient datasets for network analysis[9].

Most studies use simplified datasets with one transmission line and uniform power generation, mainly in the U.S. and EU countries. However, the study of power grid stability and reliability has made significant progress recently, thanks to a more diverse pool of generation and transmission technologies that respond to complex demands [9].

Power grids consist of transmission lines that function as edges between nodes, enabling high-voltage electricity to travel over long distances. Within the power grid network, generators such as coal and natural gas power plants, wind turbines, and solar panels serve as the nodes that produce electricity [10][8].

The interconnected and complex structure of the electricity grid delivers benefits such as reliability, flexibility, and economic competition. As technology improves, significant enhancements can be made to the electricity grid [10]. However, occasional blackouts serve as reminders of the importance of the crucial role played by power grid reliability [11]. The most common cause of blackouts is cascading failures, where the failure of one component (node) can trigger the successive failures of other parts of the network [12]. Techniques like optimal islanding have been developed to combat blackouts, which splits the total grid into sub-grids or "islands", complete with their own control and independent generation [13]. As each island possesses backup power sources like distributed generators (DG), islanding reduces the risk of cascading failures and by extension outages amongst the subgrids, should they be disconnected from the main power grid [13].

Analysis on power grid vulnerability to blackouts finds the distance between nodes impacts the 'efficiency' of the network which was first introduced by Latora and Machiori [14] [15]. A new 'net-ability' metric which incorporates the distance is proposed and shown to be able to identify at least some of the most critical connections in the network [15]. Common approaches regarding network centrality such as betweenness/degree centrality prove inefficient for complex networks (CN) systems such as power grids in regards to physicality and operational performance. This stems from not being able to cover the full range that arises from the complex nature of power grids. The proposed solution from studies suggests that 'net-ability' is a more viable way to showcase possible failures within a real life power grid [16].

Additionally, power grid system vulnerability to failures have been modelled in extant literature using the NetworkX software package. This analysis considered the removal of up to five nodes which would ensure a non-functioning grid system. While removing these nodes found only a 4% decrease in average global efficiency, the minimum local efficiency showed a significant 40% decrease. The transmission structure also becomes fragmented when three or more nodes fail. In the case of an electrical network, this poses challenges to power grid operation. The researchers conclude that only in a small percentage of cases do the power grids maintain operations when fragmented [17].

The modular structure of power grids is built around synchronisation stability of nodes and transmission lines dispersing power out to less dense rural areas. Measurement for dynamic stability is classified into two sections: Functional Secureness (FS), how a node is influenced in the grid, and Functional Robustness (FR), how a node influences others [18]. Critical nodes tend to be found within the large communities of the nation due to high volumes of consumer use of electricity, which creates problems for reliance if one node is to fail.

Current research on power networks has focused on various case studies. Some of these have been discussed here as the methods and concepts utilised were deemed relevant for the analysis presented in this report. However, an exhaustive review of all case studies explored in extant literature remains outside of the scope of this assignment.

Researchers analysed the structural vulnerability of a simplified Nordic transmission grid. CN theory was employed to quantify topological characteristics such as clustering coefficient and four centrality measures - betweenness, closeness, degree, and combined centrality [19].

The analysis revealed that the weighted combined centrality measure was the most effective at identifying critical nodes in a power grid from a topological standpoint. These nodes exhibit specific functions and influence on the network structure [20]. The critical nodes correlated with regions involved in a large-scale grid expansion project. This finding strengthens the method's validity with respect to a real transmission grid [19].

Researchers conducted a case study in Iran to analyse the country's power grid's condition, structure, and vulnerability. They introduced a new method for predicting power grid vulnerabilities which considers network topology, line loads, and grid failure rates. A modified version of the Weighted PageRank (WPR) algorithm, which uses a variable to determine the significance of each node in the network, was used to calculate network topology. Their method achieved consistent results with methods based on grid characteristic calculations by power grid engineers, while being computationally faster, and allowing for real-time updates [21].

An analysis on the South Korean power grid proposed a new approach to examining the resilience of such networks. The authors suggest societal factors and context should also be considered, instead of focusing only on the technological aspect. This sociotechnical approach is shown to provide more comprehensive recommendations when it comes to improving the emergency response necessary in a blackout, compared to only focusing on one of these two aspects [22].

Two of these case studies aimed to identify critical nodes in power grids in Iran and Scandinavia. Despite using different methods, both studies successfully identified efficient ways to locate these critical nodes and analyse their resilience. The analysis presented in this report seeks to accomplish a similar task using simplified methods.

In the case of Scandinavia, the use of centrality measures, specifically the weighted combined centrality, proved to be effective in identifying crucial nodes. In contrast, the South Korean study challenged the effectiveness of network centrality and advocated for a broader socio-technological approach. While centrality was utilised as part of the case study of the CCPG, conducting an analysis of the social context of the country remains beyond the scope of this assignment.

Altogether, these studies enhance our understanding of power grid vulnerabilities by incorporating topological features and real-world applications, they are not an exhaustive list of all case studies reviewed in extant literature and do not represent the full range of methods and approaches that could be adopted to examine power grids.

Dataset

The CCPG dataset from the year 2016 has been used for this report. It was obtained from the Index of Complex Networks repository (a link can be found in the corresponding reference) [23]. The specific dataset is part of a larger collection collated by Kim et al. [24].

This dataset consists of 347 nodes and 444 edges, and the network is both undirected and weighted. Each edge represents a power line, also referred to as transmission line, and each node represents a power node within the grid. Additionally, the dataset contains a voltage attribute, representing the network weights.

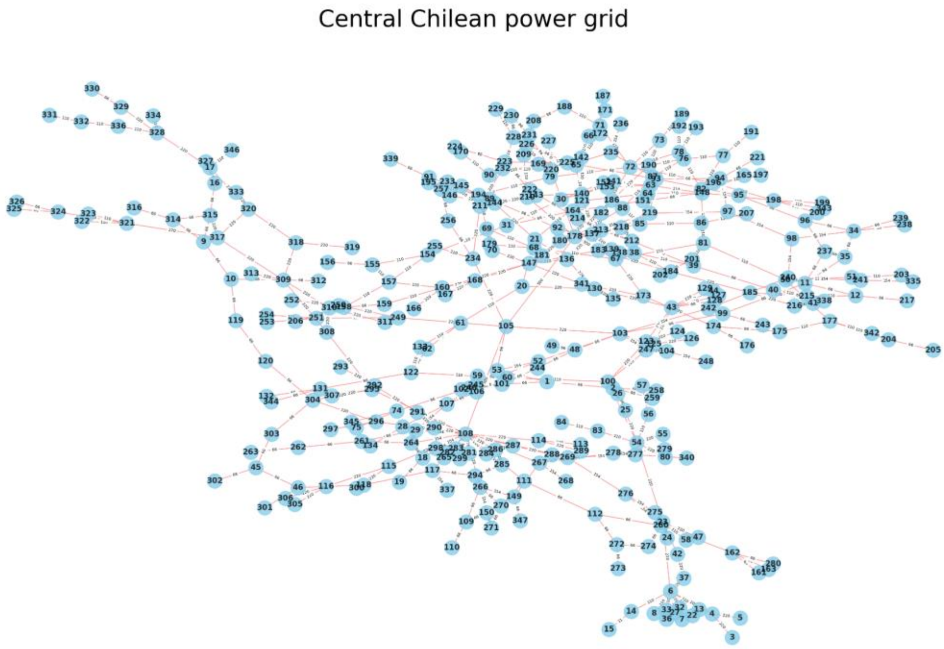

Figure 1. Central Chilean power grid

A visualisation of the CCPG is presented in Figure 1. Nodes have been labelled. Additionally, edges connecting the nodes have been labelled with their respective voltage flows. This network is cyclic which suggests that it possesses loops or cycles.

The CCPG dataset has undergone a thorough review of raw data by the original researchers. Any errors and unrealistic faults have been filtered out to create a more trustworthy dataset. Therefore, no pre-processing was required for this assignment [9]. This dataset does not provide power load data for individual nodes, so it is not possible to identify the most critical nodes based on their power load.

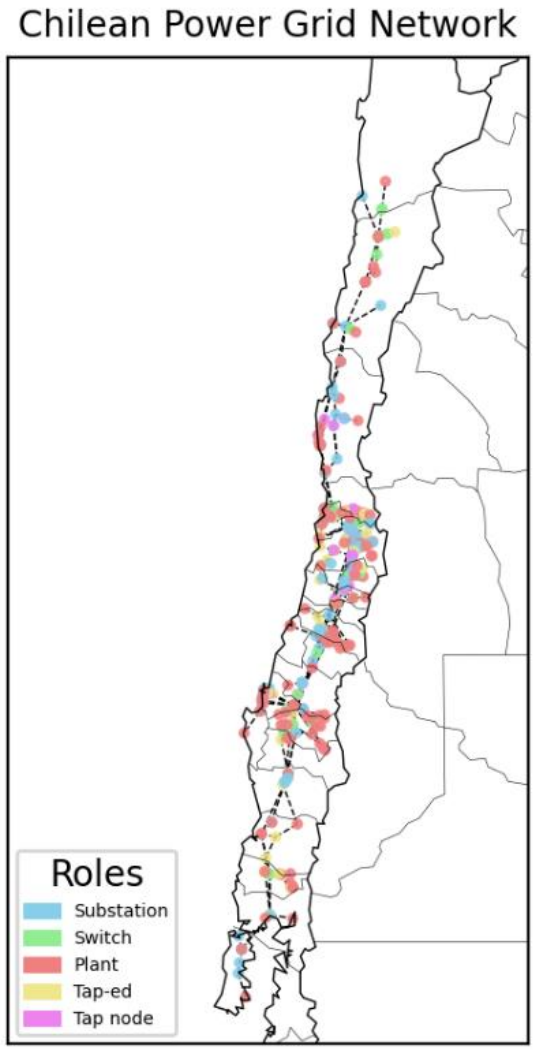

Figure 2. Map of the Central Chilean Power Grid Network

Figure 2 displays a colour code for different node roles in the CCPG Network, including substations, switches, plants, tap-ed, and tap nodes.

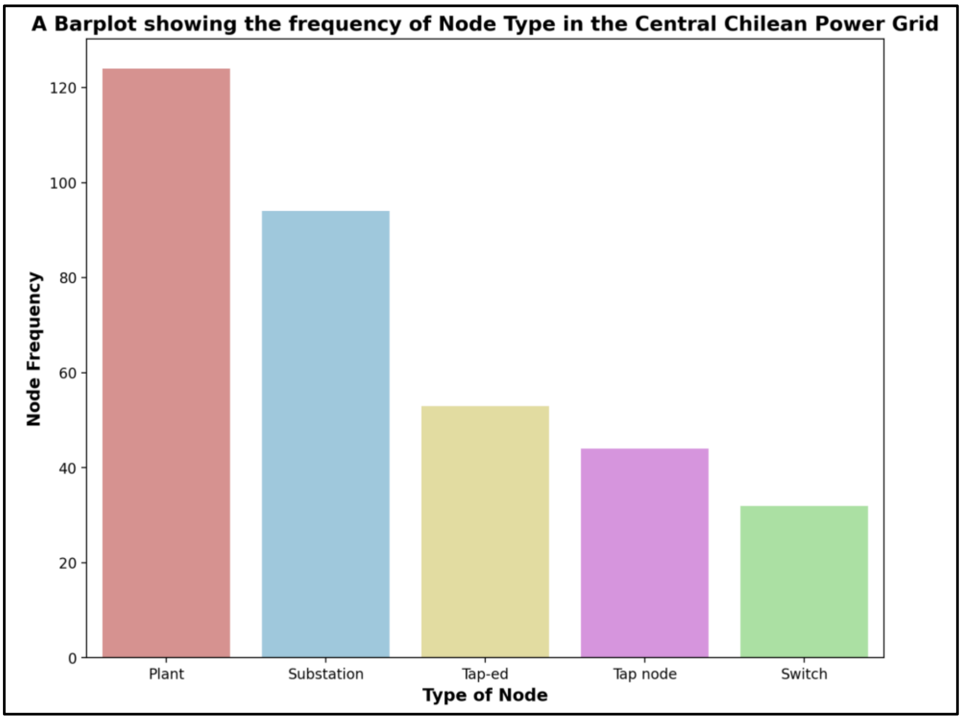

Figure 3. Distribution of the types of nodes

The predominant power node type in Chile is the plant node (Figure 3). Power plants are responsible for the generation of electricity through the spinning of a steam turbine. The steam can be generated from various sources of fossil fuels or renewable energy [1].

The second most common node is the substation, these nodes play a key role in the integration of electricity into the power grid, distributing electricity at suitable voltage levels for use in industrial, commercial and residential sectors [25]. The next two nodes, tap-ed and tap-node, are used to connect a node to the middle of a transmission line. This connection is used to maintain connectivity between other nodes [9]. The least frequent is the switch node, switches are used so that substations can be disconnected from the transmission grid when necessary [1].

Analysis and results

General Network Overview

A preliminary analysis of the network characteristics was undertaken as part of this report (Table 1). The CCPG network consists of only 444 edges whilst the maximum number of edges available is 60031 suggesting that the network is sparse [8]. The average degree of 2.559 indicates that each node is connected to approximately 2 to 3 other nodes, which suggests the idea that the network is sparse [8]. The average clustering coefficient value of 0.0865 which is low suggests the network does not have many triangles of nodes clustered together [8]. This implies the network lacks redundancy between connections [26]. This affects the resilience of the nodes to failure. This is further supported by the density score of 0.0074, indicating a limited number of connections between nodes.

Table 1. General Network Information	
Measure	**Value**
Number of Edges	444
Number of Nodes	347
Average Degree	2.5591
Density	0.0074
Average Clustering Coefficient	0.0865

Centrality Analysis

CC is a measure that identifies how close a node is from one to another, considering the closest paths [8]. The mean 0.1282 and median 0.1279 indicate low levels of closeness centrality suggesting, on average, the nodes are not close to other nodes (Table 2).

BC quantifies the significance of a node in the flow across the total number of shortest paths throughout the network [27]. The high range of 0.6008 suggests that there are some crucial nodes required for connecting to other sections of the network. Although, the mean 0.0207 and median 0.0058 are relatively low which indicates that the majority of the nodes are not crucial in the network. Comparing the range and mean shows a large difference which could indicate some outliers. The 10% winsorized mean is 0.01 which is lower than the mean, supporting the assumption that there are outliers present.

Table 2. Degree, Closeness, and Betweenness Statistics

	Range	Mean	Median	Winsorized mean 10%
Degree	23	2.5591	2	2.3285
Closeness Centrality	0.1405	0.1282	0.1279	0.1279
Betweenness Centrality	0.6008	0.0207	0.0058	0.01

The histogram diagram (Figure 4) and statistical table (Table 2), both indicate that the degrees for the CCPG are right-skewed. The frequency peaks around 2 before drastically decreasing. This is reflected in the statistics, as the nodes are mostly connected to between 2 and 3 other nodes. This could be linked to population density, resulting in higher consumption in some areas [28]. Closeness centrality is normally distributed, peeking around a score of 0.120-0.130. This indicates that the distribution of nodes is not too far from each other, which enhances efficient energy transfer from one node to another [29].

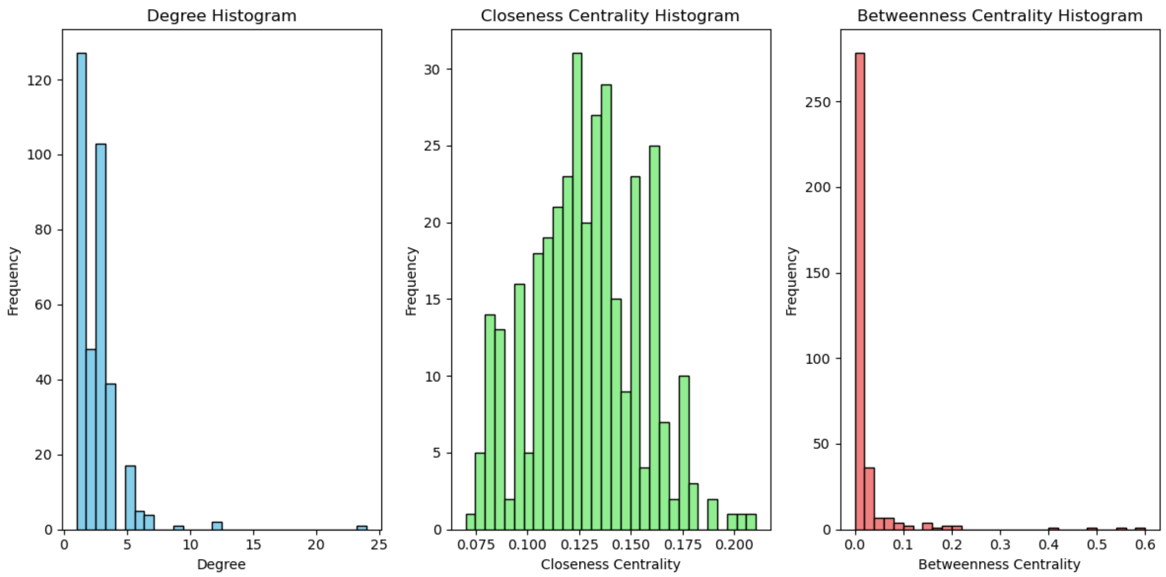

Figure 4: Histogram plots for degree, closeness centrality and betweenness centrality.

Lastly, the BC histogram shows a clear right-skew with a peak at 0.02. The mean BC of 0.02 suggests most of the nodes do not form connections through the shortest path [26]. Nodes 178 and 105 are examples of nodes which do play a more critical role since their betweenness centralities are 0.6 and 0.55 respectively. Further analysis on this subject was performed and is presented in Appendix A. Both node 178 and 105 as substations which are expected to have high BC measures, due to their role in transmission of electricity.

Node 178 has the closeness centrality of 0.2106 (Table 4). This value suggests that node 178 on average has the shortest path to all other nodes within the CCPG network. This is further supported by its BC measure of 0.6008, the highest in the network. This indicates that this node has the highest number of shortest paths going through it [26].

Table 4: Top 10 nodes for closeness centrality measure.

Node	Role	Centrality
178	Substation	0.2106
105	Substation	0.2057
147	Substation	0.1970
108	Substation	0.1912
183	Tap-ed	0.1891
67	Substation	0.1789
219	Tap node	0.1784
186	Substation	0.1783
218	Tap node	0.1776
220	Substation	0.1765

Table 4 shows the top 10 nodes for closeness centrality. These are the main nodes that could be responsible for the distribution of energy to other nodes via the shortest paths available. This is important for efficient energy transfer as distance affects energy loss [29]. This set of nodes could be important in the distribution of electricity and if left vulnerable to threats could lead to damage and major blackouts within the power grid [7][20].

Table 5: Top 10 nodes for betweenness centrality measure.

Node	Role	Betweenness Centrality
178	Substation	0.6008
105	Substation	0.5539
108	Substation	0.5002
147	Substation	0.4142
136	Switch	0.2197
130	Substation	0.205
43	Substation	0.198
295	Substation	0.1899
308	Substation	0.1666
100	Substation	0.1597

Similarly, the 9 out of 10 nodes for BC are substations which is expected since the BC of a substation is directly related to the levels of energy passed through it (Table 5) [30]. BC is a key statistic used to identify critical nodes in a network via the number of total shortest paths that pass through said node [19].

The BC for the top 10 nodes is considerably higher than the BC mean of the CCPG of 0.02. Substations play a pivotal role in the power grid via their processing and transmission of usable electricity to various sectors [7][30][25].

Clustering Coefficient

The measure of degree in which nodes tend to cluster in the form of triangles, linked to the connectivity of a network is the clustering coefficient [13]. The majority of the nodes have a clustering coefficient of 0 which indicates not many nodes form triangles with their neighbouring nodes, meaning that there is low connectivity within the network (Figure 5) [26]. A few nodes have a high clustering coefficient of close to 1 indicating there are some closed triangles present and suggest these components are strongly connected [26].

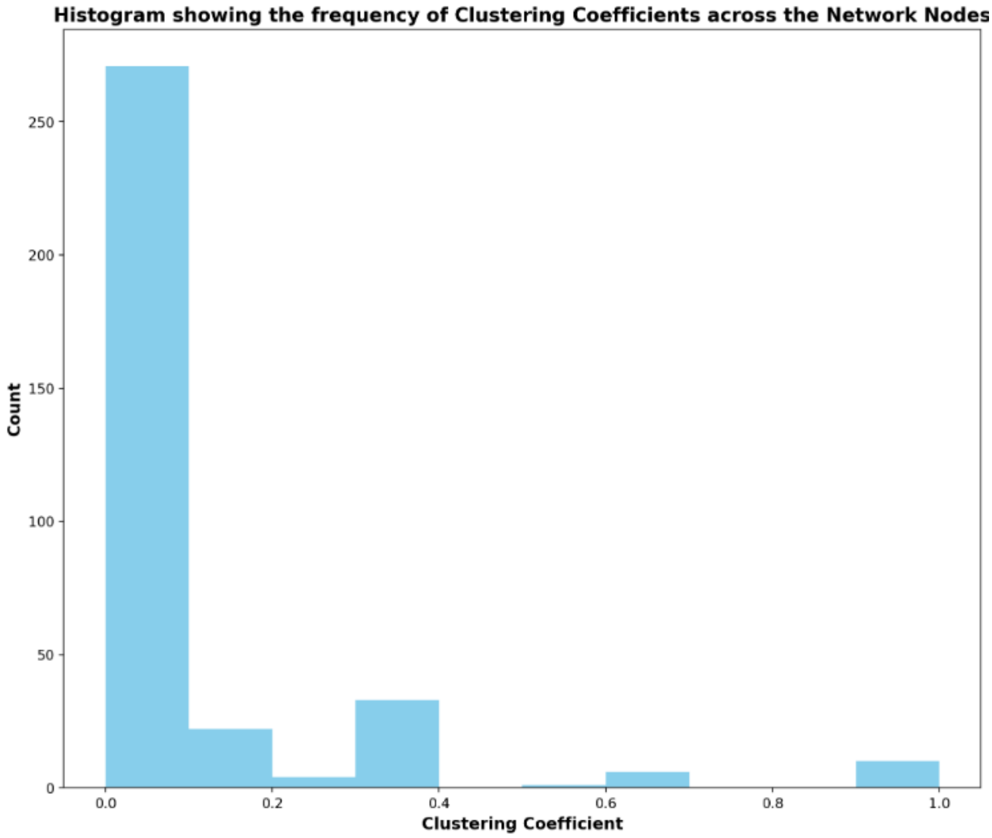

Figure 5: Histogram of clustering coefficient of the Chilean Powergrid.

Modularity

Figure 6 illustrates the maximum modularity of the CCPG which originally consisted of 347 nodes. This resulted in a maximum modularity score of 0.820 which is a high modularity measure that suggests there are dense connections within the communities but could have sparse connection between communities [8]. These communities consist of triad nodes, these nodes are all connected together [8]. However, as seen in Figure 6 there are some communities i.e. 5, 15 and 1 that have minimum intercommunity links, indicating that the communication may not be ideal or efficient from these communities [8].

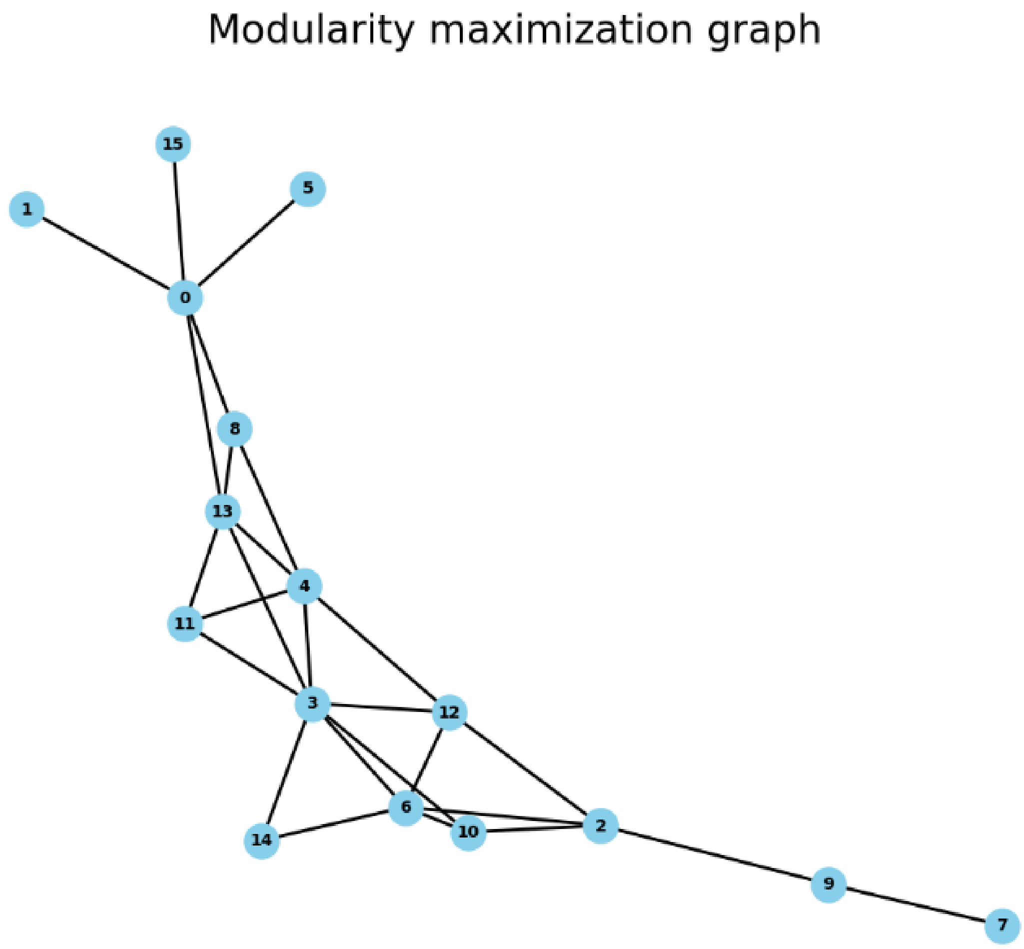

Figure 6: Modularity maximization graph.

Discussion

There were four nodes (105, 178, 147, and 108) in common between the top 10 highest CC measures and the top 10 highest BC measures. BC and CC are among the three main features chosen to understand node centrality and thereby both are important measures of criticality in nodes [7][19]. Since these nodes both share top 10 status in both categories, they are particularly critical to the optimal function of the network.

Due to historical natural disasters that have occurred in Chile, CCPG can be at risk of power outages and blackouts [2][4]. Policymakers should focus on nodes critical to the operation of the network to prepare for such eventualities. Nodes with high BC, and with joint highest BC/CC measures, should be prioritised in terms of contingency solutions to ensure a higher chance of CCPG operational continuance in the wake of natural disasters. This is supported by Saleh et al, who argue a loss of 4% of substations with high BC could result in 60% of network connectivity loss [30].

The impact of outliers in the BC distribution is highlighted in the disparity between the winsorized mean and the standard mean of said distribution. The standard mean of 0.0207 is more than double the winsorized mean of 0.01. This implies a strong upper-bound outlier component to the BC distribution and suggests these outliers are critical nodes.

Such critical nodes often act as “control nodes” to downstream large communities (sub-grids) of the main power grid that use considerable volumes of consumer electricity. A failure of one of them can create significant issues in the community [18]. This stresses the importance of “islanding” since many of these critical nodes operate as a “gateway” to a sub-grid, or island. This is needed to prevent outages and the effect of cascading failures from the main power grid [18][13]. These nodes are required to continue to produce and transmit electricity to islands within CCPG even if isolated from the main power grid. Thus, they are worthy of special attention.

Analysis conducted on the CCPG found that 9 of the top 10 critical nodes that possess the highest BC are substation nodes, which are known to act as control nodes [18]. To provide adequate mitigation across the majority of the power grid against cascading failures and outages, these nodes should be a key focus of policymakers. Investing in redundancy, and DG backup systems for these nodes will ensure said islands will continue to operate, even if said control nodes are isolated from the power grid, or there have been failures in the main power grid itself.

While this paper attempted to identify areas of the network which are at risk based on their clustering coefficient and modularity measure, this task proved more difficult as the majority of the network is sparse. Natural disasters pose an issue to the integrity of the power grid. Therefore, it is essential to understand the connectivity of the critical nodes when optimising the power grid.

Further research

It is important to note, this report has several key limitations. Firstly, the analysis is focused on a single case study. Thus, caution must be exercised when attempting to generalise the results or apply the findings outlined above to different power networks. Additional research into the local context and the specifications of the networks in questions must be undertaken. One of the limitations in terms of the Chilean power grid is limited research in the English language.

Furthermore, this analysis utilises a narrow set of analytical tools. While their usage was appropriate for this report, other methods and metrics such as ‘net-ability’, PageRank or exploring the social context of the country alongside the technical attributes of the network may be better suited for use when analysing other networks. These methods could be used to confirm the validity of the results presented here [15][16][21][22]. Comparing the results obtained using these more sophisticated methods to this analysis will ensure the findings and recommendations discussed in this report are robust.

Furthermore, the global shift towards a higher reliance on renewable energy sources may have implications for the CCPG. The transition towards renewable energy which is supported by market mechanisms and new grid technologies is crucial for power grid resilience and efficiency [31]. The impact of this change to the network should be explored further to ensure the results presented in this paper remain relevant.

Conclusion

This report provided an overview of the CCPG Network examined through the lens of network theory. A literature review highlighting relevant extant literature was provided. Several case studies of power grids conducted in other countries were discussed. Some of the methods utilised in these case studies were applied to the case study presented in the current report. All results and analysis were outlined in the analysis section. The authors of this paper identified critical nodes using centrality measures, as well as particular areas of network vulnerability based on clustering coefficient and modularity.

Critical nodes are important and have important functions within a network, in this report they were found to be substations. Chile has a history of natural disasters and in cases of these events this type of nodes should be protected for optimised function of the power grid during these events.

The main recommendations of the paper are for policymakers. The findings of this report could be used to help inform expansion and improvement strategies, as well as the development of contingency plans and coping strategies in case of network failure.

The challenges identified within the CCPG network are complex and diverse. This report aims to be a stepping stone for the future. The insights within this study will serve as critical research for those who seek to optimise power grids against potential risks.

References

1. Brain M. How Power Grids Work. In: Smith.edu [Internet]. Clark Science Center Smith College; [cited 2024 Mar 7]. Available from: https://www.science.smith.edu/~jcardell/Courses/EGR220/ElecPwr_HSW.html#:~:text=You%20use%20it%20for%20heating.

2. Contreras D, Shaw D. Disaster management and resilience in electric power systems: the case of Chile. In6th International Disaster and Risk Conference (IDRC Davos 2016) 2016. Newcastle University.

3. Xu Y, Xing Y, Huang Q, Li J, Zhang G, Bamisile O, Huang Q. A review of resilience enhancement strategies in renewable power system under HILP events. Energy Reports. 2023 May 1;9:200-9.

4. Moreno J, Shaw D. Community resilience to power outages after disaster: A case study of the 2010 Chile earthquake and tsunami. International journal of disaster risk reduction. 2019 Mar 1;34:448-58.

5. Li C, Liu W, Cao Y, Chen H, Fang B, Zhang W, Shi H. Method for evaluating the importance of power grid nodes based on PageRank algorithm. IET Generation, Transmission & Distribution. 2014 Nov;8(11):1843-7.

6. Reka Albert, Istvan Albert, and Gary L Nakarado. “Structural Vulnerability of the North American Power Grid.” *arXiv.org* (2004): n. pag. Web.

7. Forsberg S, Thomas K, Bergkvist M. Power grid vulnerability analysis using complex network theory: A topological study of the Nordic transmission grid. Physica A: Statistical Mechanics and its Applications. 2023 Sep 15;626:129072.

8. Newman M. Networks. In: Oxford University Press eBooks. Oxford University Press; 2018.

9. Kim H, Olave-Rojas D, Álvarez-Miranda E, Son SW. In-depth data on the network structure and hourly activity of the central Chilean power grid. Scientific data. 2018 Oct 23;5(1):1-0.

10. UCSUSA. How the Electricity Grid Works | Union of Concerned Scientists. In: www.ucsusa.org [Internet]. 2015 [cited 2024 Mar 7]. Available from: https://www.ucsusa.org/resources/how-electricity-grid-works#:~:text=Reliability%3A%20Since%20the%20grid%20is.

11. Pagani GA, Aiello M. The power grid as a complex network: a survey. Physica A: Statistical Mechanics and its Applications. 2013 Jun 1;392(11):2688-700.

12. Schäfer B, Witthaut D, Timme M, Latora V. Dynamically induced cascading failures in power grids. Nature communications. 2018 May 17;9(1):1975.

13. Pahwa S, Youssef M, Schumm P, Scoglio C, Schulz N. Optimal intentional islanding to enhance the robustness of power grid networks. Physica A: Statistical Mechanics and its Applications. 2013 Sep 1;392(17):3741-54.

14. Latora V, Marchiori M. Efficient behavior of small-world networks. Physical review letters. 2001 Oct 17;87(19):198701.

15. Arianos S, Bompard E, Carbone A, Xue F. Power grid vulnerability: A complex network approach. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2009 Mar 1;19(1).

16. A. M. Amani and M. Jalili, "Power Grids as Complex Networks: Resilience and Reliability Analysis," in IEEE Access, vol. 9, pp. 119010-119031, 2021, doi: 10.1109/ACCESS.2021.3107492.

17. Hadaj P, Strzałka D, Nowak M, Łatka M, Dymora P. The use of PLANS and NetworkX in modeling power grid system failures. Scientific Reports. 2022 Oct 19;12(1):17445.

18. Kim H. How modular structure determines operational resilience of power grids. New Journal of Physics. 2021 Jun 1;23(6):063029.

19. Wu X, Cao W, Wang J, Zhang Y, Yang W, Liu Y. A spatial interaction incorporated betweenness centrality measure. Plos one. 2022 May 20;17(5):e0268203.

20. Sun Z, Sun Y, Chang X, Wang F, Wang Q, Ullah A, et al. Finding critical nodes in a complex network from information diffusion and Matthew effect aggregation. Expert Systems with Applications. Elsevier BV; 2023; 233(1):120927–7.

21. Shahpari A, Khansari M, Moeini A. Vulnerability analysis of power grid with the network science approach based on actual grid characteristics: A case study in Iran. Physica A: Statistical Mechanics and its Applications. 2019 Jan 1;513:14-21.

22. Eisenberg DA, Park J, Seager TP. Sociotechnical network analysis for power grid resilience in South Korea. Complexity. 2017 Oct 23;2017.

23. Clauset A, Tucker E, Sainz M. The Colorado index of complex networks. Retrieved July. 2016;20(2018):22.

24. Kim H, Olave-Rojas D, Álvarez-Miranda E, Son SW. In-depth data on the network structure and hourly activity of the central Chilean power grid. Scientific data. 2018 Oct 23;5(1):1-0.

25. Refaat SS, Mohamed A. Smart management system for improving the reliability and availability of substations in smart grid with distributed generation. The Journal of Engineering. 2019 Jun;2019(17):4236-40.

26. Filippo Menczer, Fortunato S, Davis CA. A First Course in Network Science. Cambridge

University Press; 2020.

27. Zaoli S, Mazzarisi P, Lillo F. Betweenness centrality for temporal multiplexes. Scientific reports. 2021 Mar 1;11(1):4919.

28. Muzayanah IFU, Lean HH, Hartono D, Indraswari KD, Partama R. Population density and energy consumption: A study in Indonesian provinces. Heliyon. 2022; 8(9):e10634.

29. Belitsyn IV, Khomutov SO, Belitsyn AI. Methods for reducing electricity losses in main electrical networks. IOP Conference Series: Materials Science and Engineering. IOP Publishing; 2021; 1100(1):012015–5.

30. Saleh M, Esa Y, Mohamed A. Applications of complex network analysis in electric power systems. Energies. 2018 May 29;11(6):1381.

31. Gonzales LE, Ito K, Reguant M. The Investment Effects of Market Integration: Evidence From Renewable Energy Expansion in Chile. Econometrica. 2023 Sep;91(5):1659-93.

32. Vargas-Ferrer P, Álvarez-Miranda E, Tenreiro C, Jalil-Vega F. Assessing flexibility for integrating renewable energies into carbon neutral multi-regional systems: The case of the Chilean power system. Energy for Sustainable Development. 2022 Oct 1;70:442-55.

33. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. Available from: https://numpy.org/doc/stable/reference/routines.statistics.html

34. basemap 1.4.1 documentation [Internet]. matplotlib.org. Available from: https://matplotlib.org/basemap/stable/index.html#welcome-to-the-basemap-documentation

35. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020;17:261–72. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.winsorize.html

36. Tutorial — NetworkX 2.5 documentation [Internet]. networkx.org. Available from: https://networkx.org/documentation/stable/tutorial.html

37. sys — System-specific parameters and functions — Python 3.7.3 documentation [Internet]. Python.org. 2010. Available from: https://docs.python.org/3/library/sys.html

38. matplotlib.pyplot.hist — Matplotlib 3.5.1 documentation [Internet]. matplotlib.org. Available from: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

39. matplotlib.patches.Patch — Matplotlib 3.5.2 documentation [Internet]. matplotlib.org. Available from: https://matplotlib.org/stable/api/_as_gen/matplotlib.patches.Patch.html

Appendix A

1. Import Packages

Import packages

import pandas as pd # Used for loading in our dataset

from scipy.stats.mstats import winsorize # Used to winsorize the centrality measures [42].

import seaborn as sns # Used to add aesthetics to graphs such as the barplot

import sys # Used to adjust adjacency matrix to allow us to see the full size [44].

import networkx as nx # Used frequently throughout the whole code so we could analyse our network [43].

import matplotlib.pyplot as plt # Used to plot graphs such as the histogram for centrality measures [45].

import numpy as np # We used numpy frequently for statistical calculations [40].

import networkx.algorithms.community as nx_comm # To find the modularity of our network from MA214 PGT lab 2

from mpl_toolkits.basemap import Basemap as Basemap # This was used to create a map of Chile [41].

import matplotlib.patches as mpatches # This was used to display a legend on the map of Chile [46].

import statistics as st # For clustering coefficient statistics

2. File directory

Set directory

READ IN THE DATA

These are nodes

power_nodes = pd.read_csv('/Documents/Chilean_PowerGrid_Data/WithTap_node.csv')

These are edges

edges = pd.read_csv('/Chilean_PowerGrid_Data/WithTap_edge.csv')

3. Checking Data

Checking data

Using head function to view the first 5 rows in each dataset.

print(power_nodes.head())

print(edges.head())

4. NETWORK MAP OVERLAYING CHILE

REFERENCED FROM MA214-LAB-2 AND [34][39]

```
# Chilean Power grid network map


# Create Graph from the edges CSV, using "Source" & "Target" to direct edges, and assigning each edge "Voltage".

G = nx.from_pandas_edgelist(edges, source = 'Source', target = 'Target',

                            edge_attr = 'Voltage', create_using = nx.Graph())


# Set Figure and DPI to suitable size for Visualisation

plt.figure(figsize=(15, 5), dpi = 200)


# Plotting coordinates to show a map of Chile.

m = Basemap(

        projection='merc',

      llcrnrlon=-80.0,

    llcrnrlat=-44.0,

    urcrnrlon=-66.0,

    urcrnrlat=-22.0,

        lat_ts=0,

        resolution='l',

        suppress_ticks=True)


# Assigning a colour code to each node role.

roles = power_nodes['Role'].unique()

role_colours = {'Substation':'skyblue', 'Switch':'lightgreen', 'Plant':'lightcoral', 'Tap-ed':'khaki', 'Tap node':'violet'}

node_colors = [role_colours[role] for role in power_nodes['Role']]
```

```
# Apply Basemap to the Longitude and latitude series, which as a tuple are then placed as a value in a dict with

# the power_nodes['Id'] column as the key.

mx, my = m(power_nodes['Longitude'], power_nodes['Latitude'])

pos = dict(zip(power_nodes['Id'], zip(mx, my)))


# Draw the Network.

# I have colour coded the nodes depending on their role. Which will be shown via a legend.

nx.draw_networkx_nodes(G, pos=pos, nodelist=power_nodes['Id'], node_color=node_colors,

                       node_size=5, alpha = 0.8)


# Drawing the network edges. I have changed the style of the edges for dotted lines.

nx.draw_networkx_edges(G, pos=pos, edge_color= 'Black', width=0.5, style = '--')


# Adding legend to display different node roles.

legend_labels = {role: mpatches.Patch(color=role_colours[role], label=role) for role in roles}

plt.legend(handles=list(legend_labels.values()), loc='lower left', title='Roles', fontsize= 5)


# Add the Underlying Chile Map

m.drawcountries(linewidth = 0.5) # Adjusting line width between countries

m.drawstates(linewidth = 0.2) # Adjusting line width for locations within the country

m.drawcoastlines(linewidth=0.5) # Adjusting line width for coastlines


plt.title('Chilean Power Grid Network', fontsize = 10) # Plotting title

plt.savefig('Network_map.png', bbox_inches='tight', pad_inches=0.1) # Saving figure for analysis

plt.show() # Showing plot below


# Basemap code from [34]].
```

mpatches legend code from [39].

5. CENTRAL CHILEAN POWER GRID

REFERENCED FROM LAB 2 AND BELOW

```
# Produces Central Chilean Power Grid


# Set Figure size and resolution

plt.figure(figsize=(20,12), dpi=200)

# Create a Blank Graph.

PowerGrid = nx.Graph()


# Using the fn add_nodes_from(), add nodes for the power_nodes['Id'] column. The PID of the Dataset.

PowerGrid.add_nodes_from(power_nodes['Id'])


# Add edges from branch to target, set voltage values on the edges

PowerGrid.add_edges_from([(edges['Source'], edges['Target'], {'voltage': edges['Voltage']}) for idx, edges in edges.iterrows()])


# Set New Layout

pos = nx.spring_layout(PowerGrid)


# Draw graph and set font weights, size and aesthetics

nx.draw(PowerGrid, pos, with_labels = True, font_weight = 'bold', node_size = 350, font_size = 10, node_color = 'skyblue', alpha = 0.8, edge_color='lightcoral')


# Isolate Voltage and draw as an edge label

edge_labels = nx.get_edge_attributes(PowerGrid, 'voltage')

nx.draw_networkx_edge_labels(PowerGrid, pos, edge_labels=edge_labels, font_size=5)
```

plt.title('Central Chilean power grid', fontsize = 30) # Plotting title

plt.savefig('Central_Chilean_PowerGrid.png', bbox_inches='tight') # Saving figure for analysis

plt.show() # Shows the plot

6. CONFIRM NUMBER OF NODES ASSIGNED TO EACH ROLE (PLANT, SUBSTATION, TAP-ED, TAP NODE, SWITCH)

plt.figure(figsize=(10,8), dpi=200) # Setting figure size

plt.ylabel("Node Frequency", fontdict={'fontsize': 12, 'fontweight': 'bold'}) # Y-axis label

plt.xlabel("Type of Node", fontdict={'fontsize': 12, 'fontweight': 'bold'}) # X-axis label

Confirm no null values under the Roles series

len(power_nodes) # 347

sum(power_nodes["Role"].notnull())

Create Value counts of the Role Series

values = power_nodes["Role"].value_counts()

Create the desired colour mapping

role_colours = {'Substation':'skyblue', 'Switch':'lightgreen', 'Plant':'lightcoral', 'Tap-ed':'khaki', 'Tap node':'violet'}

order = values.index

Order the Colours to match with required bars

colour_palette = [role_colours[role] for role in order]

Adding title and colours to the plot

plt.title("A Barplot showing the frequency of Node Type in the Central Chilean Power Grid", fontdict={'fontsize': 14, 'fontweight': 'bold'})

sns.barplot(y=values.values, x=order, palette=colour_palette)

```
# Saving barplot figure to include in our dataset explanation
plt.savefig("A Barplot showing the frequency of Node Type in the Central Chilean Power Grid");
```

7. ISOLATE NUMBER OF NODES AND EDGES IN NETWORK

REFERENCED FROM CODE IN MA214-LAB-1

```
# Print the number of nodes
print(f'number of nodes: {PowerGrid.number_of_nodes()}')

# Print the number of edges
print(f'number of edges: {PowerGrid.number_of_edges()}')
```

8. CALCULATE ADJACENCY MATRIX

REFERENCED FROM MA214-LAB-2

```
# Create an adjacency matrix whilst enabling maximum display of node connections
np.set_printoptions(threshold=sys.maxsize) # Allows us to view the full adjacency matrix
A = nx.adjacency_matrix(PowerGrid)
print(A.todense())
```

9. CONFIRM WHETHER NETWORK IS CYCLIC OR ACYCLIC

REFERENCED FROM MA214-LAB-2

```
# Confirm whether the referenced network is cyclic or acyclic


nx.is_directed_acyclic_graph(PowerGrid)
```

10. FIND SHORTEST PATHS BETWEEN KEY NODES

REFERENCED FROM MA214-LAB-4

```
# Print Density of "PowerGrid"
print(nx.density(PowerGrid))
```

```
# Find degree of PowerGrid Nodes

degree_powerGrid=PowerGrid.degree()


print(type(degree_powerGrid)) # - Confirm Type to show need for conversion

degree_powerGrid
```

11. CALCULATE DEGREES, CLOSENESS AND BETWEENNESS OF NETWORK

REFERENCED FROM MA214-LAB-5 AND MA214-LAB-6

```
# Creating dictionaries for each measure and printing their outputs

degrees = dict(nx.degree(PowerGrid))

closeness = dict(nx.closeness_centrality(PowerGrid))

betweenness = dict(nx.betweenness_centrality(PowerGrid))

print("The degrees of each node:\n", degrees)

print('The closeness centrality of each node:\n', closeness)

print('The betweenness centrality of each node:\n', betweenness)
```

12. Calculating Statistics for Centrality Measures

With the use of numpy functions [33]

```
# Values were extracted from dictionaries and assigned to list for statistical calculations.
degree_values = list(degrees.values())
closeness_values = list(closeness.values())
betweenness_values = list(betweenness.values())


# Here, numpy functions were utilized to find the range, mean, and median for degrees, closeness,
# and betweenness.
degree_range = np.ptp(degree_values) # The np.ptp function was used to calculate the range.
degree_mean = np.mean(degree_values) # The np.mean function was used to calculate the mean.
degree_median = np.median(degree_values) # The np.median function was used for calculating the median.
```

```
closeness_range = np.ptp(closeness_values)
closeness_mean = np.mean(closeness_values)
closeness_median = np.median(closeness_values)

betweenness_range = np.ptp(betweenness_values)
betweenness_mean = np.mean(betweenness_values)
betweenness_median = np.median(betweenness_values)

# Printing each calculations while clearly labelling each one. Ready for interpretation.

# Degree Statistics
print("Degree Range:", degree_range)
print("Degree Mean:", degree_mean)
print("Degree Median:", degree_median)

# Closeness Centrality Statistics
print("Closeness Centrality Range:", closeness_range)
print("Closeness Centrality Mean:", closeness_mean)
print("Closeness Centrality Median:", closeness_median)

# Betweennesss Centrality Statistics
print("Betweenness Centrality Range:", betweenness_range)
print("Betweenness Centrality Mean:", betweenness_mean)
print("Betweenness Centrality Median:", betweenness_median)
# The following Numpy statistical functions are from [33]
```

13. Histogram plot for Centrality measures

With the use of matplotlib [38]

```
# Histograms for Degree, Closeness Centrality, Betweenness Centrality

plt.figure(figsize=(12, 6)) # Plotting figure size

# Plotting Degree Histogram. Creating three subplot to show all three histograms in one figure.
```

plt.subplot(1, 3, 1) # Creating a subplot to add to our main figure

plt.hist(list(degrees.values()), bins=30, color='skyblue', edgecolor='black') # Plotting histogram with aesthetics

Set the bins to 30 so we can clearly see the distribution across the X axis

plt.title('Degree Histogram') # Adding title

plt.xlabel('Degree') # Adding X axis label

plt.ylabel('Frequency') # Adding Y axis label

Plotting closeness centrality histogram

plt.subplot(1, 3, 2) # Creating a subplot to add to our main figure

plt.hist(list(closeness.values()), bins=30, color='lightgreen', edgecolor='black') # Plotting histogram with aesthetics

plt.title('Closeness Centrality Histogram') # Adding title

plt.xlabel('Closeness Centrality') # Adding X axis label

plt.ylabel('Frequency') # Adding Y axis label

Plotting betweenness centrality histogram

plt.subplot(1, 3, 3) # Creating a subplot to add to our main figure

plt.hist(list(betweenness.values()), bins=30, color='lightcoral', edgecolor='black') # Plotting histogram with aesthetics

plt.title('Betweenness Centrality Histogram') # Adding title

plt.xlabel('Betweenness Centrality') # Adding X axis label

plt.ylabel('Frequency') # Adding Y axis label

plt.tight_layout() # Adjusting layout for a clearer image

Saving figure and using bbox_inches to control the image proportions.

plt.savefig('Degreemclosenessbetweenness.png', bbox_inches='tight')

plt.show() # Displaying plot

The following histograms were created using the following [38].

14. IDENTIFY NODE TYPE FOR MAXIMAL CENTRALITY NODE

print(power_nodes.head())

print(edges.head())

print(power_nodes["Role"].unique())

```
# Identify Node type for Maximal Centrality Node
print(power_nodes.iloc[107])
print(power_nodes.iloc[177])
```

15. THE TOP 10 NODES FOR BETWEENNESS CENTRALITY

```
# A Breakdown of the top 10 Nodes for Closeness Centrality

# Convert NX object to dict format

closeness = nx.closeness_centrality(PowerGrid)


# Sort nodes by closeness centrality

sorted_closeness = sorted(closeness.items(), key=lambda x: x[1], reverse=True)


# Create a list using comprehension that creates nested lists comprised of the node index and
related centrality

top_closeness = [[node ,round(value, 4)] for node, value in sorted_closeness[0:10]]

print(top_closeness)


# Isolate the Index of the Top 10 Closeness Nodes

top_closeness_index = [idx - 1 for idx, cent in top_closeness]


# Generate Type of node for Top Closeness Centralities via DF ILOC:

power_nodes.iloc[top_closeness_index][["Id", "Role"]].set_index("Id")[0:10]
```

16. THE TOP 10 NODES FOR BETWEENNESS CENTRALITY

```
# A Breakdown of the top 10 Nodes for Betweenness Centrality

betweenness = nx.betweenness_centrality(PowerGrid)
```

Sort nodes by betweenness

sorted_betweenness = sorted(betweenness.items(), key=lambda x: x[1], reverse=True)

top_betweenness = [[node ,round(value, 4)] for node, value in sorted_betweenness[0:10]]

print(top_betweenness)

Isolate the Index of the Top 10 Betweenness Nodes

top_betweenness_index = [idx - 1 for idx, cent in top_betweenness]

Generate Type of node for Top Betweenness Centralities via DF ILOC:

power_nodes.iloc[top_betweenness_index][["Id", "Role"]].set_index("Id")[0:10]

17. SHARED NODES WITHIN TOP 10 OF BOTH BETWEENNESS AND CLOSENESS CENTRALITY

Identify shared nodes within top 10 of both betweenness and closeness centrality

Reset Index for top_betweenness and top_closeness

top_betweenness_index = [idx for idx, cent in top_betweenness]

top_closeness_index = [idx for idx, cent in top_closeness]

print("Top betweenness index:",top_betweenness_index); print("Top closeness index:", top_closeness_index)

#Find common items

common_elements = list(set(top_betweenness_index) & set(top_closeness_index))

print("Column elements found in both top 10 CC & BC Measures are: {}".format(common_elements))

18. IDENTIFY OUTLIERS FOR BETWEENNESS CENTRALITY

```
betweenness = nx.betweenness_centrality(PowerGrid)

betweenness_values = list(betweenness.values())

betweenness_values

betweenness_values = list(betweenness.values())


# Calculate Q1 and Q3

Q1 = np.percentile(betweenness_values, 25)

Q3 = np.percentile(betweenness_values, 75)


# GET IQR

IQR = Q3 - Q1


# Set Bounds for Outside IQR

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR


# Identify outliers

outliers = [[node, centrality] for node, centrality in betweenness.items()

                if centrality < lower_bound or centrality > upper_bound]




# Sorting the nodes within bounds by their centrality values in descending order

sorted_outliers = sorted(outliers, key=lambda x: x[1], reverse=True)


sorted_outliers
```

19. FINDING WINSORIZED MEAN TO FURTHER ANALYSE THE CENTRALITY MEASURES

With the use of Scipy and Numpy functions [35][33].

```
#Convert the centrality measures to arrays

degrees_array = np.array(list(degrees.values()))

closeness_array = np.array(list(closeness.values()))

betweenness_array = np.array(list(betweenness.values()))


#Winsorize the centrality measures

winsorized_degrees = winsorize(degrees_array, limits=[0.10, 0.10])

winsorized_closeness = winsorize(closeness_array, limits=[0.10, 0.10])

winsorized_betweenness = winsorize(betweenness_array, limits=[0.10, 0.10])


#Calculate the mean of winsorized centrality measures

mean_winsorized_degrees = round(np.mean(winsorized_degrees), 4)

mean_winsorized_closeness = round(np.mean(winsorized_closeness), 4)

mean_winsorized_betweenness = np.mean(winsorized_betweenness)


#Print the mean winsorized centrality measures

print("Mean winsorized degrees:", mean_winsorized_degrees)

print('Mean winsorized closeness centrality:', mean_winsorized_closeness)

print('Mean winsorized betweenness centrality:', mean_winsorized_betweenness)


# The winsorize function was sourced from the following [35].

# The numpy functions [33].

.
```

20. CALCULATE LOCAL CLUSTERING COEFFICIENT

CLUSTERING COEFFICIENCY CODE - REFERENCED FROM MA214-LAB-7

```
cluster_coef = nx.clustering(PowerGrid, nodes=None, weight=None)

# Calculate the mean, std deviation, median and range of the local clustering coefficients.

mean = st.mean(cluster_coef.values())

median = st.median(cluster_coef.values())

std = st.stdev(cluster_coef.values())


# Calculate Range

min = min(cluster_coef.values())

max = max(cluster_coef.values())


print("The mean of the Clustering Coefficient is: {}".format(mean))

print("The median of the Clustering Coefficient is: {}".format(median))

print("The Standard Deviation of the Clustering Coefficient is: {}".format(std))

print("The Range of the Clustering Coefficient is: {}".format(max-min))
```

21. PLOT HISTOGRAM FOR CLUSTERING COEFFICIENTS

```
plt.figure(figsize = (12,10), dpi=200) # Plotting figure size and and image resolution


plt.hist(cluster_coef.values(), density=False, bins=10, color = 'skyblue') # Plotting histogram


plt.xlabel("Clustering Coefficient", fontdict={'fontsize': 12, 'fontweight': 'bold'}) # X-axis labels


plt.ylabel("Count", fontdict={'fontsize': 12, 'fontweight': 'bold'}) # Y-axis label
```

Plotting title

plt.title("Histogram showing the frequency of Clustering Coefficients across the Network Nodes", fontdict={'fontsize': 14, 'fontweight': 'bold'})

Saving figure for analysis

plt.savefig("Histogram showing the frequency of Clustering Coefficients across the Network Nodes");

22. MODULARITY

REFERENCED FROM MA214-PGT-Lecture-2

Greedy_modularity_communities

FROM MA214 PGT lecture 2

Greedy_modularity_communities

FROM MA214 PGT lecture 2

We are looking to maximise modularity

we use the greedy modularity algorithm creating communities in the variable 'parition'

partition=list(nx_comm.greedy_modularity_communities(PowerGrid))

print("The partition that maximize the modularity: ", partition)

next calculate the maximum modularity from the partition above

max_modularity=nx_comm.modularity(PowerGrid,partition)

print("The maximum modularity is ", max_modularity)

create two new variables to store the lengths of our new partitions and our original nodes

partition_groups = len(partition)

original_nodes = len(PowerGrid.nodes())

Print the new partitions

```
print(f'New Groups: {partition_groups}')


# Print the original nodes

print(f'Original nodes: {original_nodes}')


# We have found that the new graph will consist of 16 nodes/communities.


# Finding all the neighbours of the nodes in each community

neighbors=[] # empty list used for storage of neighbors from our for loops below

for i in range(0,len(partition)): # For loop scrub through the partition storing any neighbors in our blank list above

    neighbors.append(set()) # add empty set to neighbors


for i in range(0, len(partition)): # For loop scrub through the partition storing any neighbors in our blank list above

    for j in range(0, len(partition[i])):

        neighbors[i]=neighbors[i].union(set(PowerGrid.neighbors(list(partition[i])[j])))



# Create an indicator for each community.

        Indicator=[] # another empty list to store new indicators from our for loops below

for i in range(0,len(partition)): #for loops used to scrub each community finding indicator values. then storing in blank list above

    Indicator.append([])


for i in range(0, len(partition)): # we are using the neighbours of nodes in community i

    for j in range(0, len(partition)): # that are located in community j

        Indicator[i].append(len(neighbors[i].intersection(partition[j])))
```

```
# Here, we are creating a list of nodes for the new communities we have created above

        nodeslist=list(range(0,len(partition)))

edgeslist=[] #create a blank list for our edges above

for i in range(0,len(partition)): ## for loop used to scrub through our partitions and find indicators that are !=0 adding them to the edge list

    for j in range(i+1,len(partition)):

        if Indicator[i][j]!=0:

            edgeslist.append((i,j))


#plot the figure from the above edges and nodes

plt.figure(figsize=(5,4), dpi=200)


G=nx.Graph() # Create a blank network graph

G.add_nodes_from(nodeslist) # Add the nodes to the network

G.add_edges_from(edgeslist) # Add the edges to the network


#  create labels, pick a colour scheme, add definition to the numbers/nodes

nx.draw(G,with_labels=True, node_color = 'skyblue', font_weight='bold', node_size = 100, font_size = 5)


# Add a title

plt.title('Modularity maximization graph');


# Save the figures

plt.savefig('Modularity maximization graph.png')
```

print("We have created 16 nodes, which are distinct groups of nodes.\nWe have separated the network in 4 communities/networks.\nNot many edges between communities. Partition is bad")

23. CODE FILE REFERENCES:

Also provided in our references section

33. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. Available from: https://numpy.org/doc/stable/reference/routines.statistics.html
34. basemap 1.4.1 documentation. matplotlib.org. Available from: https://matplotlib.org/basemap/stable/index.html#welcome-to-the-basemap-documentation
35. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020;17:261–72. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.winsorize.html
36. Tutorial — NetworkX 2.5 documentation [Internet]. networkx.org. Available from: https://networkx.org/documentation/stable/tutorial.html
37. sys — System-specific parameters and functions — Python 3.7.3 documentation [Internet]. Python.org. 2010. Available from: https://docs.python.org/3/library/sys.html
38. matplotlib.pyplot.hist — Matplotlib 3.5.1 documentation [Internet]. matplotlib.org. Available from: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html
39. matplotlib.patches.Patch — Matplotlib 3.5.2 documentation [Internet]. matplotlib.org. Available from: https://matplotlib.org/stable/api/_as_gen/matplotlib.patches.Patch.html