German Powergrid network analysis

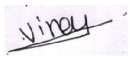
Vinay Bharambe [1]
Roll No. 21026762064
MSc(F)

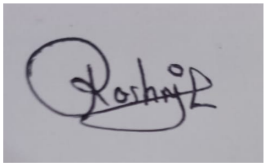
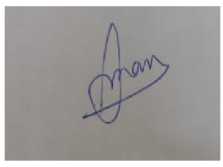
Anil Kandel [2]
Roll No. 21025762006
MSc(F)

Roshni Jharbade [3]
Roll No. 21056762059
MSc(F)

Aman Dhillon [4]
Roll No. 21025762004
MSc(F)

November 2022



Report of project work done between September and November 2022 as part of the M.Sc. course PHY-OT543 Complex Systems and Networks, Department of Physics and Astrophysics, University of Delhi

MSc(Final) Complex Systems and Networks Project,
Department of Physics and Astrophysics, Delhi University, 24 pages,
November 2022

This document is available free of charge from:
http://tiny.cc/quckuz
The programs executed in the following writeup can be found at
https://github.com/VinayBharambe/

Recommended: The document contains animated graphics and it is recommended to use javascript enables pdf viewer to have full experience, such as Adobe or KDE-Ocular.

Abstract

The field of Complex systems and Networks is an exciting, challenging, and growing subject. It has applications in almost every field from Economics, Cellular Biology, Computer science, Sociology to Physics and Cosmology. It is a realistic and interdisciplinary subject which can be used to solve real life problems in a very efficient way. It stands on the pillars of Mathematics and Computer science. The subject investigates the relationships between the elements of the problem, simplifies the problem by converting it into mathematical expressions, and provides computational solutions to the problem.

A Network is a system that has two parts, number of elements that are called "Nodes" and the connections between these Nodes are called "Edges". Now for the simplest network, there can be few number of Nodes with several Edges and It seems like the possibilities of Events in this Network can be understood manually by studying the network. But as the number of Nodes goes on increasing, the number of possible Edges goes on increasing and it becomes impossible to analyse the whole network without the help of sophisticated computing techniques. Here comes the part of computing and programming. With the help of algorithms, we automate the iterating functions of the analysis.

In our project, we are going to analyze the network of power stations in a Power grid and apply analysis to it. We'll visualize the network with various parameters of network analysis to make it visually easy to understand. We'll test its rigidity and strength by developing various attacks on the Network and also study its characteristics using network analysis tools from Networkx library of Python programming language.

We are given dataset from https://www.power.scigrid.de/. It has information about power stations from Germany. The dataset has Power station number, voltage, cables, wires, frequency, name of the operator, length of wires, etc. We'll first clean the data and then plot the data. While plotting the data we are going to make sure that the nodes are placed at the exact longitude and latitude location. This way we'll be able to locate the nodes and directions while developing the attack strategy. One amazing thing which we observed while marking the exact point is that the German Power grid is spanned in nearby countries as well. For example, one of the power station is seen in Denmark and indeed the co-ordinates of it on maps points to a power station. Although it is for a shorter distance and for nearby countries only.

In order to visualize various properties of the network, we plotted the network and used network visualization tools. The first property is that there are four different types of nodes: Substations, Plant, Generator and Auxiliary transmission nodes. We differentiated them by means of different colors. In order to develop an attack for a particular type of node, this characterization into 4 types is very important. Next, we visualized the sizes of the nodes depending on the total degree of the node, which will be further used to devise an attack depending on the maximum degree of the nodes. In addition, We made the edge width dynamic depending on the number of cables and wires between two nodes. Again, which also helped in developing a special type of attack.

We calculated the theoretic of the graphs as well as the degrees of the nodes. In this problem we have an undirected network, hence it will be total degree. We studied its distribution and inferred important hints on the network's weak nodes. We measured the Closeness centrality, Betweeness centrality, and Eigenvector centrality of the graph, as well as their distribution. The distribution and the measures of these centrality led to identifying nodes important in terms of enemy attack.

Since it is an open and interesting problem. We have taken the liberty to imagine and develop new types of attacks. **In a conventional Power grid failure problem the focus is mostly on the Cascading failures and Vulnerability due to inter connectivity, but in terms of an attack strategy these are the nodes that are going to be heavily protected.** Hence, we've worked around it and used imagination to develop an attack strategy. The First 4 attack algorithms make use of the centrality concept. The Fourth algorithm makes use of the fact these Plant and Generators are the special types of nodes that power the whole network and the Fifth algorithm used the first intersection of first 4 and the Geographical direction of the attack. With the above algorithms, we achieved an attack strategy with which can shut down $1/4^{th}$ of the network by attacking a small geographical region. The last type of attack uses imagination of a sabotage, in which disconnecting a single link will lead to damage to a small part of Power grid. This last attack with the help of edge width concept gives us ability to simulate the network evolution with a single wire being disconnected.

We've tried to keep it simple but the programs at few places have becomes lengthy. It is recommended to refer to the comments given above every line to keep track of the functioning of the program.

Table of Contents

List of Figures

1. Introduction[1]

Electricity is a basic necessity for humans and almost all appliances are operated using electricity from Air conditioners to Data centers, from mobile phones to Military Radars and surveillance. Although electricity was discovered in the 1700s almost all of it's Generation, transmission and supply are through the Power grid. The reason behind this is the storage of electricity on large scale is still an unsolved problem. Most of the electricity is generated in plants and is simultaneously supplied to the network. Hence in case of Power grid failure, the whole network will face power cut for very long time until the network is fully restored.

Due to the above reasons Power grid is a subject of national power security, economic development, and people's daily life. Although Power grid failure can happen due to various reasons from Climate change to natural calamities like floods and earthquakes. There are equal chances of Power grid failure due to uncontrolled load and most importantly security reasons. We are going to analyze the Power grid network in the case of on attack by an enemy state.

"A cascading failure in a complex system is a process in which an initial failure in one or several of its elements leads to a sequence of failures which spread to other elements and in some cases collapses the whole system"[2]. There have been many such cascading failures in the past. The 2003 North American Blackout is a classical example of it. In India July 2012 North India blackout happened due to a similar scenario which affected

[1]Author: Vinay Bharambe

almost 670 Million people. The vulnerability due to interconnectivity is the cause of the Grid failure. If stopped in time this can be prevented.

1.1 Objectives of the Problem

We'll be dealing with a special scenario in which the failure will be directed in a specific direction of the Grid. Since, attacking over a small region is easier. We've five objectives in front of us.

1. Plot the network on Geographical map of Germany
2. Measure the different theoretic of the Network
3. Visualise the network with various graph parameters
4. Devise an Attack strategy and observe its effects on the network
5. Suggest a Defense strategy

While addressing the first question we are going to plot network over fixed co-ordinates of the Nodes. The nodes will be plotted over a background of Germany map. Plotting of network will take place with NetworkX package of Python.

We'll measure the various parameters of the Graph such as the Degree of the nodes, Betweenness centrality, Eigenvector centrality, Closeness centrality, etc. All of these will take place by using python functions and at few places some for loops. We'll study the degree distribution and centrality distribution of all 3 centralities.

To dynamically visualise the network we will change the parameters of the Nodes and Edges such as Node size, Edge width, Node color based on namely total degree, weight of edge and Node type.

We tackle this question in innovative way and have devised 6 different ways of attacking the network, where each shows different effect on the network. Finally, use a collective attack strategy and observe its effect on the Power grid.

In this section, we give suggestions on decreasing the steep centrality distribution and making the network more decentralized.Also, use of smart grid to prevent small scale failures to escalate into Power grid failures.

1.2 Packages Used in the Program

We've used 5 main packages in the analysis of the network. These are main stream packages used in any Network analysis and Network Visualisation program.

1. NetworkX : NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
2. NumPy : NumPy is the fundamental package for scientific computing in Python. It provides many inbuilt mathematical functions for the analysis.
3. Matplotlib : Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
4. Math : This package provides access to the mathematical functions defined by the C standard. It provides exponential, trigonometric and logarithmic funtions.
5. Plotly : Plotly is a free, open-source library in python. It provides with interactive plots integration.

In recent times, there has been an attack on the Power grid from cyberspace. Hacking incidents from foreign countries have been detected on the Computer systems of the National Power grid. In this scenario, the network is vulnerable not only on the borders of the country, but also at a power station well within the country that can lead to cascading effect.

1.3 Data cleaning

We have used `loadtxt()` function of the `NumPy` library of Python to clean the data and import it as required for the network analysis.

2. Study of Network parameters [2]

A **Network** is, in its simplest form, a collection of points joined together in pairs by lines[1]. The two components of any network are nodes and edges. Building upon these two components the most complex systems we can imagine are formed. In this section, we are going to introduce to the various parameters and simultaneously measure those parameters in our network.

1. Nodes : These are the points or vertices in a graph. These can be animals in a food chain, person in a Society, ants in a community,etc.
   To find the number of nodes in a graph we use NetworkX command

   ```
   len(nx.Graph().nodes())
   ```

   We have 511 nodes in our network.

2. Edges : Edges are the connections or lines between two nodes. These can be food chain pattern between two species of animals, social interaction between two people, two ants meeting each other while searching for food, etc.
   To find number of edges in a graph we use NetworkX command

   ```
   len(nx.Graph().edges())
   ```

   We have 679 edges in our network.

3. Directivity of Graph : While forming a network there can be two types of Directivity: Directed Graph and Undirected Graph.

   An **Undirected Graph** `G = G(V,E)` consists of set V of vertices `(V =` V_1`,` V_2`,` $V_3, \ldots V_n$ and set E of edges `(E =` $E_1, E_2, E_3, \ldots E_k$`)`, where each edge E_k is an unordered pair of vertices E_k=`(`V_i`,` V_j`)`.

   A **Directed Graph** `G = G(V,E)` consists of set V of vertices `(V =` V_1`,` V_2`,` $V_3, \ldots V_n$ and set E of edges `(E =` $E_1, E_2, E_3, \ldots E_k$`)`, where each edge E_k is an ordered pair of vertices E_k=`(`V_i`,` V_j`)`.

   In our network we have undirected graph. Since, If a power station A is connected with power station B, then obviously Power station B is connected with power station A.

[2] Author: Vinay Bharambe

4. Degree of a node : In an undirected graph, the degee of a node `i` is the number of undirected edges K_i, that contain the node.

$$K_i = \Sigma_{j=1}^{n} A_{ij} = A_{ji}$$

To find number of degrees in a graph we use NetworkX command

```
nx.degree(G)
```

Here onwards we will be using, `G = nx.Graph()`.

The maximum degree of a node in our graph is **15**. Further we plotted the degree distribution of the graph and found that the graph has very high number of nodes with low degree and only 16 nodes have degree > 7.

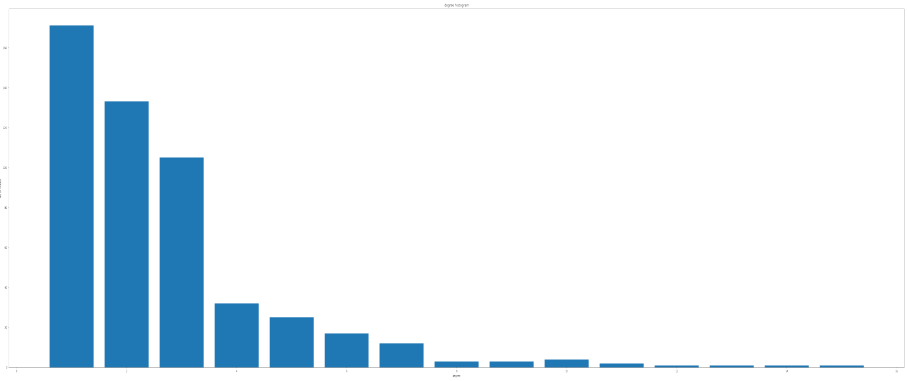


Fig. 1. Degree Distribution

```
X-axis :  Degree of a node (n)
Y-axis :  Number of nodes with (n) degree
```

5. Degree Centrality : Degree centrality is the concept of centrality dependent on its total degree, that is number of edges a node have. More the number of edges more will be the Degree centrality of the node.
To find Degree centrality of nodes in a graph we use NetworkX command

```
nx.degree_centrality(G).values()
```

The maximum Degree centrality that we found was 0.0294 and was of node number 354.

6. Betweenness Centrality : In a network, between two nodes there can be different paths to go from one node to another. Betweenness centrality is the measure of such paths where the path is the shortest.
   To find Betweenness centrality of nodes in a graph we use NetworkX command

   ```
   nx.betweenness_centrality(G).values()
   ```

   The distribution of measure of Betweenness centrality of various nodes can be seen in plot below.

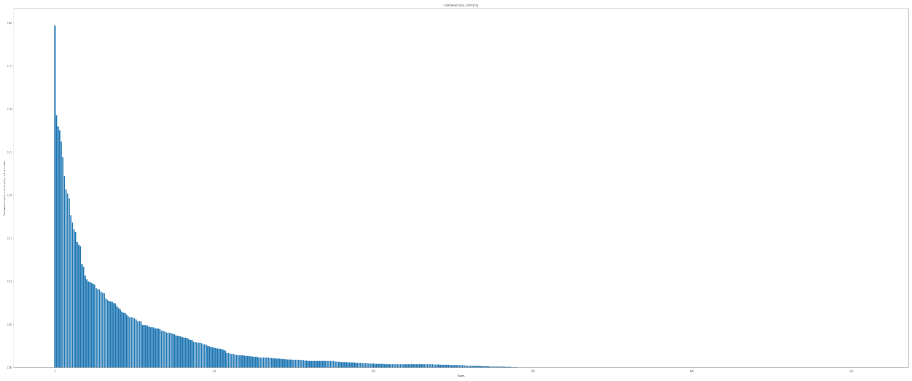


Fig. 2. Betweenness Distribution

```
X-axis :  Nodes
Y-axis :  Measure of Betweenness centrality for the Node
```

   The graph shows a steep transition between nodes with lower betweenness centrality and nodes with higher centrality. This means few nodes that have shorter length access to the network compared to the other nodes. The maximum Betweenness centrality that we found was 0.3967 and was of node number 68.

7. Eigenvector Centrality : Eigenvector centrality is a relative concept. It represent strength of connection of a node compared to other nodes in the network. Depending on the influence of the node, it is alloted a value. The value is such that a node with connections with nodes who itself have higher connections will be greater.
   Google's PageRank centrality and the Katz centrality are variants of the eigenvector centrality.
   To find Eigenvector centrality of nodes in a graph we use NetworkX command

   ```
   nx.eigenvector_centrality(G).values()
   ```

   The distribution of measure of Eigenvector centrality of various nodes can be seen in plot below. The maximum Eigenvector centrality that we found was 0.4627 and was of node number 116.

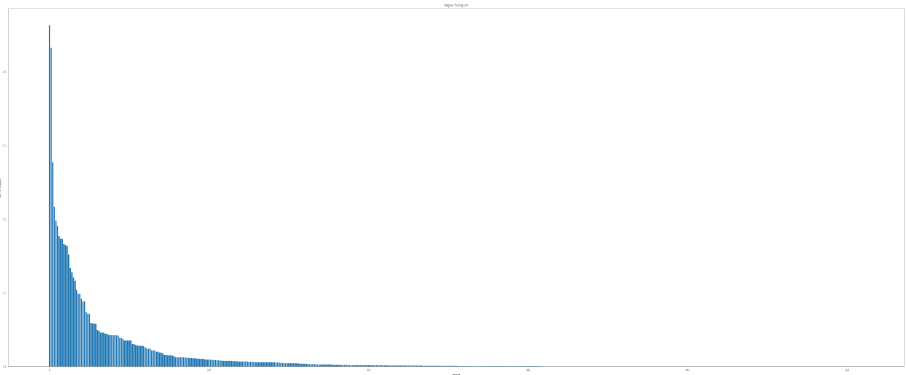


Fig. 3. Eigenvector Distribution

```
X-axis :  Nodes
Y-axis :  Measure of Eigenvector centrality for the Node
```

The plot shows steep change in the Eigenvector centrality. Few number of nodes have more influence on network compared to others.

8. Closeness Centrality : Closeness centrality indicates how close a node is to all other nodes in the network. It is calculated as the average of the shortest path length from the node to every other node in the network[3].

   To find Closeness centrality of nodes in a graph we use NetworkX command

   ```
   nx.closeness_centrality(G).values()
   ```

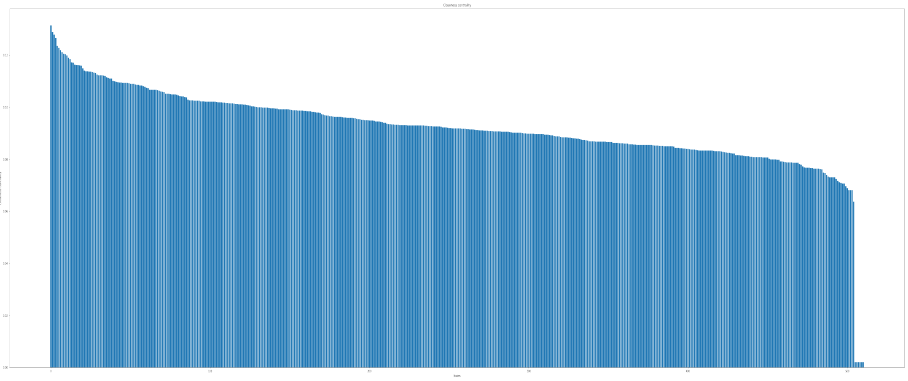


Fig. 4. Closeness Distribution

```
X-axis :  Nodes
Y-axis :  Measure of Closeness centrality for the Node
```

The distribution of measure of Closeness centrality of various nodes can be seen in plot above.

The maximum Closeness centrality that we found was 0.1313 and was of node number 68. The plot shows an almost equal distribution of the closeness centrality. This shows that almost all the nodes have approximately same average shortest path.

9. Clustering coefficient: Clustering coefficient is a concept from Graph theory. It is the measure of how much clustered the nodes are to each other. More the clustering coefficient shows more is the togetherness of the graph.
   To find Eigenvector centrality of nodes in a graph we use NetworkX command

   ```
   average_clustering(G)
   ```

   The average clustering coefficient of the graph was found to be 0.1357, which is on about 14 percent that means the network is not highly clustered.

3. Network Visualisation [3]

3.1 Node Characteristics

3.1.1 Nodesize

We have used the concept of the degree of a node to determine the size of the node. The higher the number of edges, the larger is the size of the node. By this, we can clearly see which node has the highest degree and have maximum connections with nearby nodes.

```
#Adding nodesizes

degs=[ ]
nodesize=[ ]
for i,deg in nx.degree(G):
    degs.append(deg)
    nodesize.append(150*deg)
```

3.1.2 Node type

Color assignment can be done by using the degree as a reference. A Degree less than 5 is assigned a blue color, five to ten is assigned an orange, and degree more than 10 is assigned red color. Also, It can be done by type of node. In this dataset, we have four types of nodes namely Substation, Plant, Generator and Auxiliary transmission nodes. We've used blue, yellow, green, and orange colors respectively for above types.

```
#color coding using station type as reference

color=["#0f03fc"]*511
for i in range(0,len(x),1):
    if (type[i]==1):
        color[i]="#0f03fc"
    elif(type[i]==2):
        color[i]="#fcfc03"
```

[3] Author: Anil Kandel

```
    elif(type[i]==3):
        color[i]="#3aeb34"
    else:
        color[i]="#fc7b03"
```

3.2 Edge characteristics

In our dataset, we are given additional information about edges. Since it is an electric network, we are given the number of Cables and wires that are between two nodes. Based on this concept we can assign some weight to Cables and wires and change the width of the edge depending on number of cables and wires between two nodes. We have assigned a weight of 0.2 to wires and 1 to cables.

```
#adding links with weights (weight assumptions, cables=1 ,wires=0.2)

weight_edge = [ ]
for i in range(0,len(p),1):
    weight_edge.append(math.ceil(1+cables[i]+(0.2*wires[i])))
    G.add_edge(p[i],q[i],weight=weight_edge[i])
```

3.3 Positioning of nodes

One more important thing to take into account was the positions of the nodes. We are handling a problem in which the direction of the network is going to play a crucial role. A network can be built with different types of layouts but here we need to fix the nodes to it's geographical co-ordinates. We used `pos` attribute of the NetworkX package to do this.
We additionally plotted the network over the map of Germany to make the directions of the network more visible and realistic. We used `subplot` feature of the Matplotlib library to do it.
After all these modifications the Network is plotted and can be seen in the below plot.

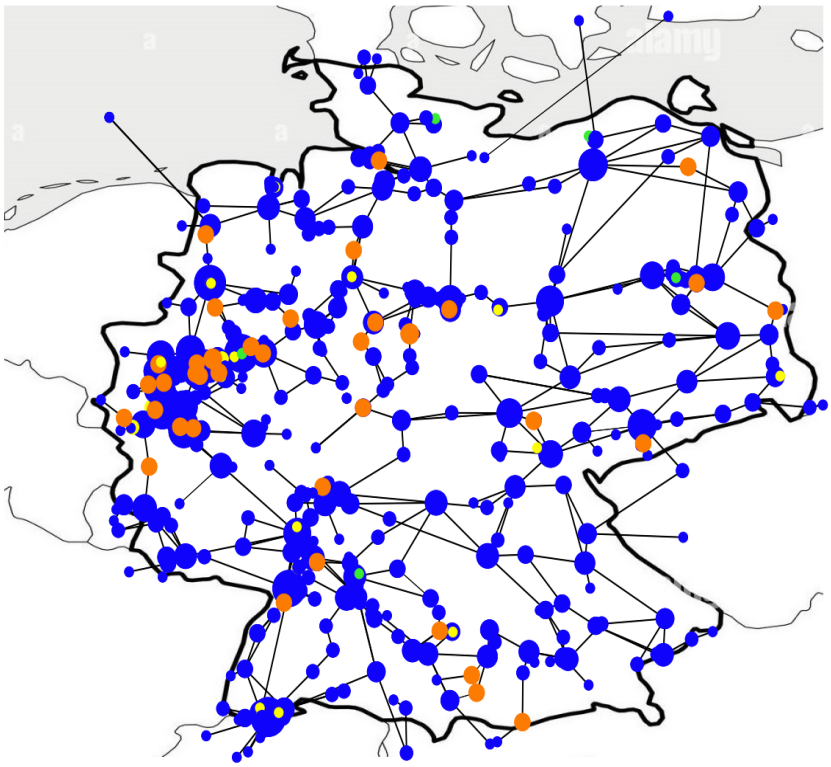


Fig. 5. Network Visualisation

Attacking strategy: - Attacking strategy can be done in two ways:

- Attacking the nodes with maximum degree. This way of attacking is efficient because it will destroy the inter-connectivity of huge numbers of links.
- Attacking a power plant or generator directly .This will have huge impact on the power supply to substations.

Attacking the nodes with maximum degree.

To do this, we've to search for the node with the maximum number of degrees. To distinguish it from other nodes, we will change the nodes color with particular degrees and then delete the node to see the effect on the network. Finally, we will choose to attack one or more nodes which will take down most of the networks.

We have highlighted nodes with maximum degrees in our program by using a for loop to identify nodes with degrees greater than seven. The nodes with degrees higher than seven can be selected out and attacked. In order to get a new set of data, we can flush data and again apply different conditions on degree. we can then highlight the node again and continue this process. We have used red color to the node that is going to be attacked. We can also see the affected links after a attack on particular nodes.

Attaking a power plant or generator directly

This seems a highly effective strategy to attack with an optimal shot. On the graph, power plants are marked by yellow blips, and generators by green blips. These are the nodes that supply the sub-stations with electricity and thus the electricity network is created, In case of an attack on a power station, not only that single power station and its links, but also the adjacent substations will be shut down due to a power cut and their subsequent links.

Given the possibility that one substation could be powered by two or more power plants, we assume that only until the nearest second substation suffers a power failure. For example, in case of a power plant failure, only the adjacent nodes of those nodes will be switched off. Any new substation linked to the plant will be unaffected.

In making this assumption, we consider the number of total node(511) and number of power plants(24). The substations are sufficiently sparsely placed such that the cascading effect is effective only between two adjacent connections. However, it can be changed.

4. Attack Strategy [4]

Given it is an open problem we have tried to develop different types of attacks to test the rigidity of the network. We chose to not use the concept of Vulnerability due to interconnectivity since these are the nodes which are going to be most protected by nation.
In order to damage the power grid in most efficient way, one needs to direct the attack to certain nodes which have property of damaging its subsequent connections. Also, this needs to be done in quick time and hence the area of attack also should be small.
We'll start with the obvious way of attack that is attacking the nodes with maximum connections.

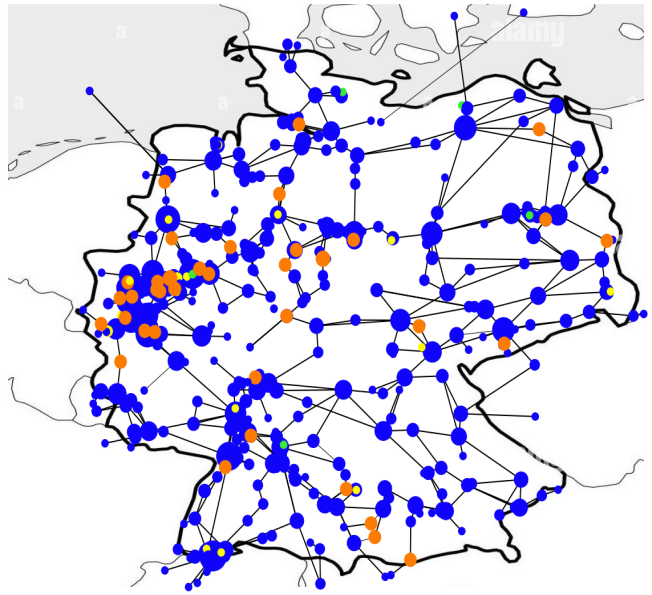
4.1 Attack using Degree of node or Degree centrality:

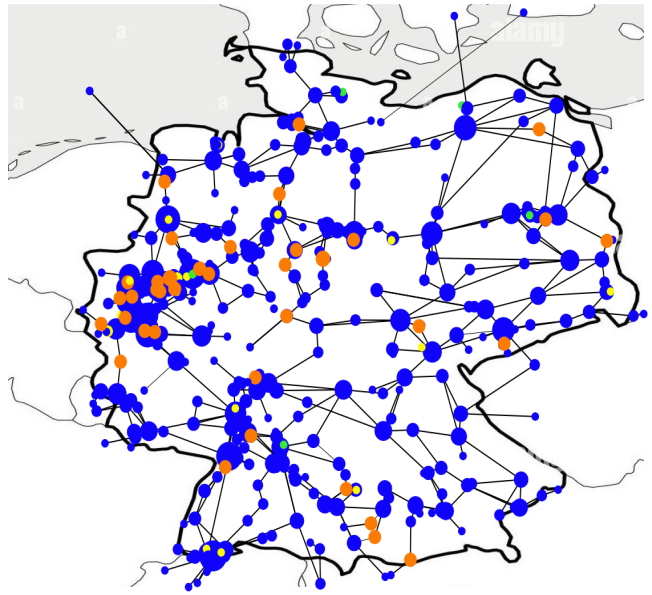
In this type, we'll first measure the degree of all the nodes in the network. We found that the maximum degree of a node in our graph is 15 and it belongs to node number 354. On observing the degree distribution of the network we see that only 16 nodes have
degree > 7 out of total 511 nodes. Hence, these nodes can be center for attack.
We visualise the time evolution in 3 steps. First we highlight the node, then we highlight the links that are connected with the nodes and then we delete the nodes and plot the network again. We calculated the number of nodes, edges again and calculate the percentage damage to the network. We infer from the attack that on attacking these 16 nodes $1/4^{th}$ of the network can be taken down. The animation below on next page shows its effect.

4.2 Attack using Betweenness centrality:

We make use of Betweenness centrality concept to carryout this attack. In a network, between two nodes there can be different paths to go from one node to another. Betweenness centrality is the measure of such paths where the path is the shortest. We found maximum Betweenness centrality of 0.3967. Out of all 3 percent of nodes had 20 percent of betweeeness. We carried out attack in these nodes and found that most of the nodes with high betweenness are at center of the graph and are not very effective to carryout geographically. While attacking these nodes by other means like cyber attack can lead to very high damage to the network. The animation below on next page shows its effect.

[4] Author: Vinay Bharambe





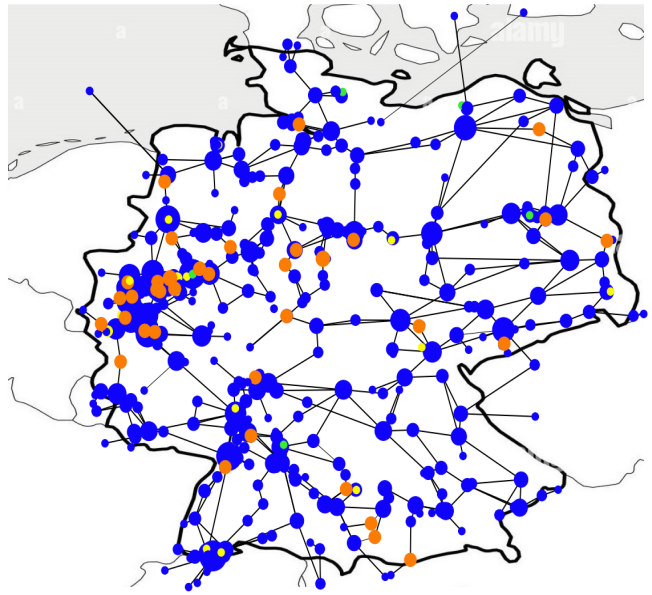
4.3 Attack using Degree of node or Eigenvector centrality:

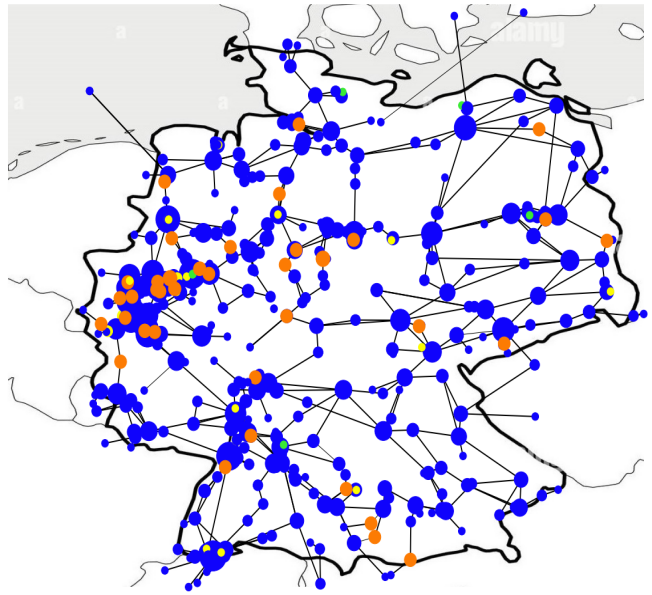
In Eigenvector centrality we measure the relative connectedness of the node. We assign a relative value to a node compared to other nodes in the network. It is possible that the node do not have high degree but it is connected with nodes who have relatively good connections with the network and hence the Eigenvector centrality of the node will be greater than other nodes who have only higher degrees. Eigenvector centrality is a very good way to measure the centrality of a network. Google's PageRank centrality and the Katz centrality are variants of the eigenvector centrality.
We found out 23 nodes above centrality of 0.08 then We delete the nodes and plot the network again. We calculated the number of nodes, edges again and calculate the percentage damage to the network. We infer from the attack that on attacking these nodes 14 percent ofthe network can be taken down. The animation below on next page shows its effect.

4.4 Attack using Degree of node or Closeness centrality:

This is not so effective attack in the case of this network since the distribution of Closeness centrality of all nodes is almost the same. That is the average of the shortest path length from the node to every other node in the network is almost the same. The maximum Closeness centrality that we found was 0.1313 and most of the values are around 0.10 and 0.11. We found 20 nodes above the Closeness centrality of 0.115 and on selecting those nodes as attack points we get damage of about 14 percent of the network.

The animation below shows the effect of the attack on the network.



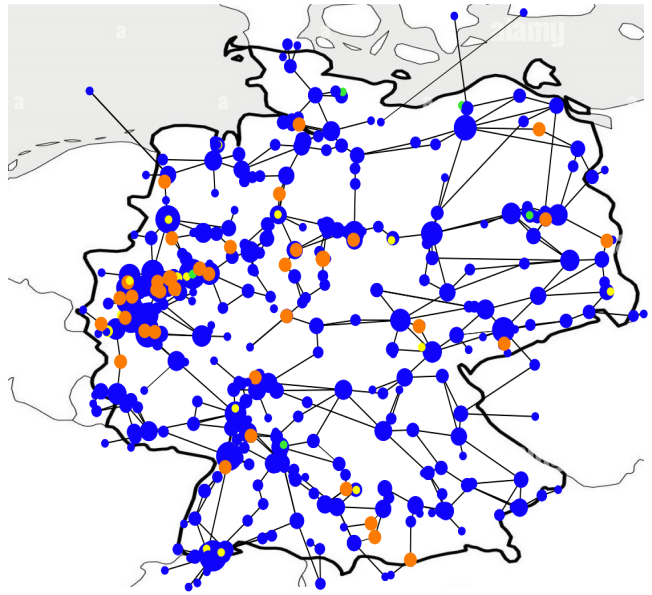


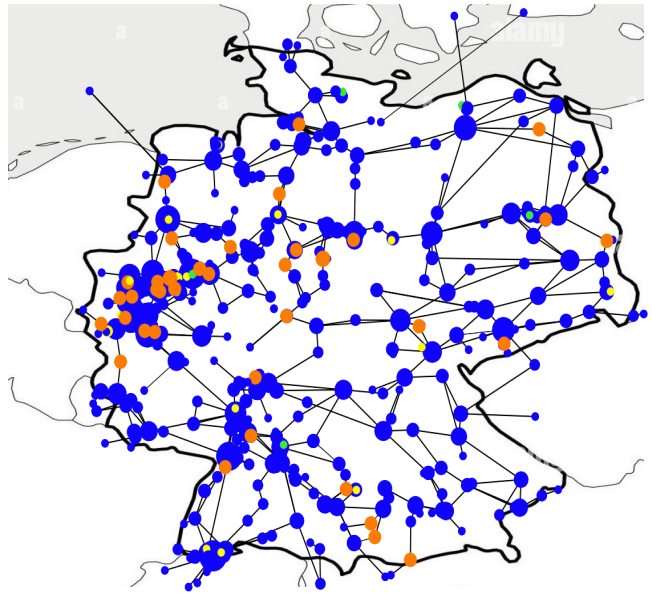
4.5 Attack using the type of the Node:

We are given a dataset of Power stations of Germany's power grid. This power grid is formed with 4 different types of power stations namely Substations, Plant, Generators and Auxiliary transmission node. Out of these 4 Substations and Auxiliary transmission nodes are the nodes which only works as intermediate nodes, they do not create or generate power. Whereas Plant and Generators are generators of the power and supply this power to the full network. Without these nodes there will be no power in the network.
Hence, these will be the nodes which will be more vulnerable to attack by an enemy state. The failure of these nodes will not only disconnect adjacent nodes(as in the case of first 4 attacks) but will be more than that. Since the network is receiving power from these nodes, the damage to these nodes will lead to power cut to the subsequently adjacent nodes to the first step adjacent nodes. For example. If Plant A is connected to 5 nodes then failure of Plant A will disconnect power to the 5 adjacent nodes and also the neighbors of these 5 nodes whose number can be 25 and this process will go on.Here we've taken minimalist approach and taken into consideration only 2 step damage while in reality it can very well be 3-4 step damage, given there are only 24 Power generating nodes behind network of 511 nodes.
We've used the data to first color the nodes of different type and then attack them. On attacking 24 nodes out of 511 almost 50 percent of the network was shut down.
The animation below shows the effect of the attack on the network.

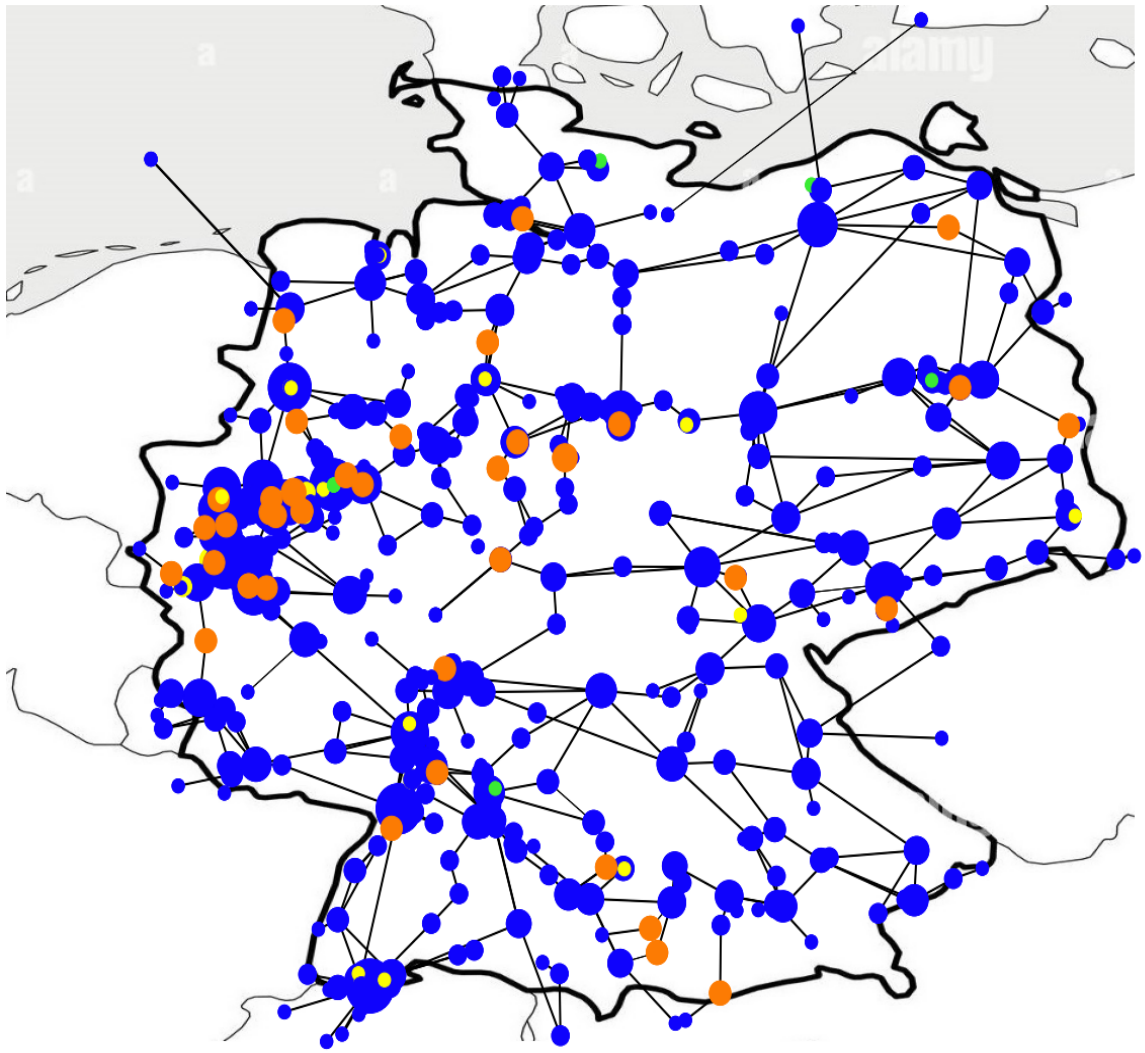
4.6 Attacking a link

It is always not feasible to make large scale attack on country. Instead some small scale sabotage is easily possible. In the case of such attack, a particular link between two nodes is disconnected which is easily possible compared to attacking a Power station facility which is heavily guarded. Attacking such a link between two nodes will disconnect the further links to the network and can make a small scale but effective impact.
For identifying such link we use a link between two such nodes with highest betweenness centrality. We achieved a damage of 3.5 percent when a single link between node number 212 and 68 was destroyed
Additionally, if that link is from a power station to a network then it will be more devastating. We identified a link between Plant 166 and node 165, which can lead to shut down of 4.5 percent of the network with the minimalist 2 step approach.
The animation below shows the effect of the attack on the network.





Finally, we collect results from all the above algorithms and suggest an attack from West direction of Germany attacking in a small region on 35 selected nodes to damame almost 40 percent of the network.



4.7 Defence strategy

1. Decentralization: The Centrality distribution graphs shows that the network is Centralised around few nodes. Out of 511 nodes only 20-30 nodes have very high centrality (this is true for all 3 centrality except closeness centrality). Hence, In order to prevent failure the network show be decentralized and majority nodes should have equal contribution in the network. By this although if a node is attacked, the Grid will remain intact.

2. Micro grids: Micro grids would have local power generation and allow smaller grid areas to be separated from the rest of the grid in case of failure. Also, In case of failure the micro grid system can power the local system and keep the power supply on[4].

3. Protection of Highly vulnerable nodes: Although if we decentralized the systems, the power generating nodes will remain to be the most vulnerable to an attack. These power stations must be protected heavily both physically and in cyber space.

Summary and Future possibilities

With the help of this dataset we've tried to analyze a hypothetical possibility of an attack on the power grid. We used various tools of network analysis and performed analysis based on centrality concepts and time stamp concept (2 step attack on power plant).Unlike conventional power grid failures due to cascading effect and vulnerability due to interconnectivity, load imbalance, we tried to have solution independent of it.
One further activity one can perform is combine the cascading effect power grid failure with the above algorithms and we'll get a better way to analyse the system.
One immediate activity one can perform is by using the above analysis on similar type of dataset available on SciGrid website for European Union power grid.

Author Contribution

1. Introduction and Power grid failure theory and literature : Aman Dhillon, Completed and written by Vinay Bharambe
2. Theoretic of the Graph : Performed by Roshni Jharbade, Completed and written by Vinay Bharambe
3. Network visualisation : Anil Kandel, Written and formatted by Anil Kandel
4. Network visualisation with Plotly : Vinay Bharambe
5. Attack strategy and Algorithms : Vinay Bharambe
6. Code writing : Anil Kandel and Vinay Bharambe
7. Analysis of results : Anil Kandel and Vinay Bharambe
8. Production of figures : Vinay Bharambe
9. Report writing : Anil Kandel and Vinay Bharambe

Acknowledgments

We would like to thank the Department of Physics and Astrophysics, Delhi University for giving us the opportunity to work on this Project. We would to thank Ms.Yashika Sethi for her assistance in the project work. We are thankful for the constant guidance from ***Prof. Sanjay Jain*** throughout the work.

References

1. Mark Newmann, Networks: An Introduction(1^{st} edition)

2. Francesc Comellas and Llorenç Sánchez, Graph Centralities and Cascading Failures in Networks, 2019

3. Jennifer Golbeck, Network Structures and Measures in Analyzing the social web, 2013

4. Power grid failure - Sourabh Kothari

Appendix

1. Attack algorithm 1 (Degree centrality)
2. Attack algorithm 2 (Betweenness centrality)
3. Attack algorithm 3 (Closeness centrality)
4. Attack algorithm 4 (Eigenvector centrality)
5. Attack on power generating nodes
6. Attack algorithm 6 (Attack on link)