University of Milan Bicocca

School of Science

Department of Computer Science, Systems and Communication

Bachelor of Science in Computer Science

DATA ANALYTICS

Project Report

Luca Poli 852027 l.poli6@campus.unimib.it

Summary

Introduction

For this Data Analytics project, I will address the problem expressed in Track 7: The Analysis of a Transportation Network, and inspired by the papers suggested in the track, I will attempt to analyze the graph of a city's public transportation network in order to assess its robustness and vulnerability to different types of attacks.

In detail, starting from the provided dataset containing download links to several GTFS (General Transit Feed Specification), I will choose one on which to focus the analysis; then, I will evaluate the structure and robustness of the network; next, I will try to simulate different types of attacks on the graph, and evaluate their impact on the network; and finally, I will build a demo that allows visualizing the previous analysis and simulating attacks on the graph.

The paper will be divided into four main chapters: in the first I will give a brief presentation on GTFS and the one chosen for the analysis; in the second I will analyze the structure of the graph and evaluate its robustness using different statistics and metrics; in the third I will simulate different types of attacks on the graph and evaluate their impact; and finally, in the fourth, I will present a demo that allows one to visualize the previous analysis and dynamically interact with the graph in order to manually study the impact of the attacks.

Chapter 1: Brief presentation on GTFS

In this chapter I will briefly introduce the provided dataset and the GTFS contained in it, specifically: what they are, how they are structured, and which one I chose to analyze.

1.1 Provided Dataset and GTFS

The dataset provided contains nearly 2000 GTFS. For each one we have the download link and some additional information: such as the publishing agency, state, region, city, etc..

A GTFS (General Transit Feed Specification) is a standardized file format (in .zipper format) that contains data from a public transportation network. It was developed by Google to make it easier to create public transportation applications. A GTFS file contains several types of files: agency.txt, stops.txt, routes.txt, trips.txt, stop_times.txt, calendar.txt, calendar_dates.txt, shapes.txt. Each file contains specific information about the public transportation network, such as: the agencies involved, stops, routes, schedules, fares, transfer rules, etc..

In particular, for graph extraction, I used the files:

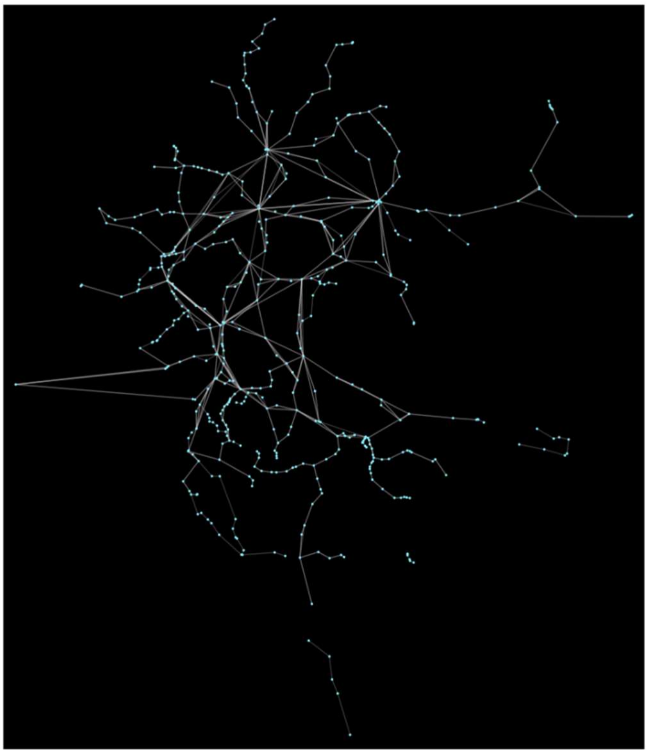
- Stops: which includes information regarding stations, for node extraction;
- Trips and stops_times: which include information regarding trips, routes and times, used for graph arc extraction.

1.2 Choice of GTFS

After a brief initial analysis of Italian and European GTFS, I chose the GTFS of Deutsche Bahn (BN), the German railway network. The main reason I chose it is its extent and complexity (475 stations and about 1200 lines); in fact, it spans the entire German territory and some neighboring states, and includes both local and regional lines and long-distance lines.

This makes the analysis on robustness and vulnerability more interesting, compared to a local network (such as a city subway); as any failures or attacks could impact a larger number of people, a larger territory, and have a higher probability of causing inconvenience and problems (because they are difficult to replace than other local lines).

Figure 1: Synthetic graph extracted from GTFS



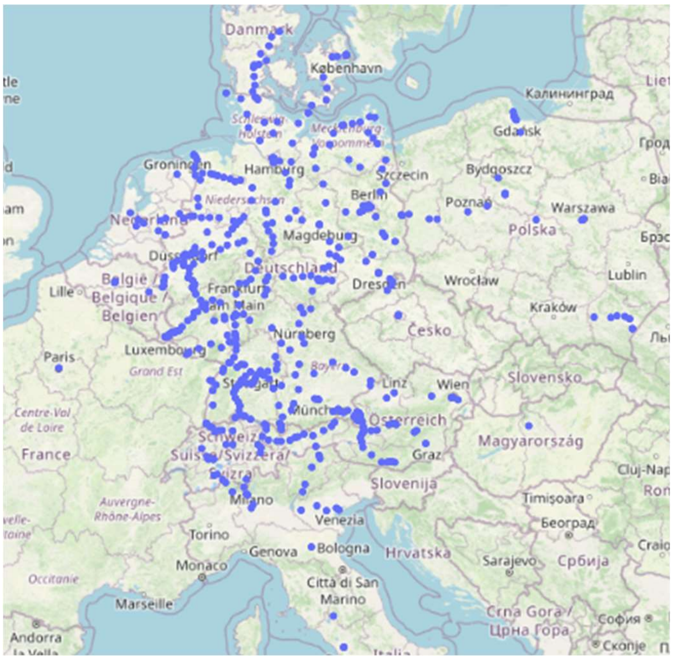


Figure 2: Simple map of stations

Chapter 2: Structural analysis of the graph

In this chapter, I will analyze the structure of the graph. Specifically, I will first expose the adopted metrics (such as degree, centrality, connectivity, etc..), then use them to extract the characteristics of the network; in order, to establish the importance of each node and assess network robustness and vulnerability.

2.1 Metrics adopted

The metrics adopted for graph analysis were chosen from those studied and based on the papers suggested in the outline, and are as follows:

- Degree: The number of arcs incident on a node.
- Centrality Degree (CD): Calculated as the normalized degree of a node relative to the maximum degree; measures a node's ability to influence the network.
- Centrality Betweenness (CB): Calculated as the number of times a node is crossed by the minimum paths between two nodes; measures the ability of a node to connect two parts of the network.

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

- Centrality Closeness (CC): Calculated as the inverse of the sum of the minimum distances between a node and all other nodes; measures the ability of a node to quickly reach all other nodes.

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

- Centrality Eigenvector (CE): Measures and assigns importance to nodes based on the importance of their neighbors; calculated via the adjacency matrix eigenvector.

$$\lambda x^T = x^T A,$$

- Centrality Clustering Coefficient (CCC): Calculated as the ratio of the number of arcs between a node's neighbors to the maximum number of possible arcs; measures the degree of interconnection between a node's neighbors.

$$c_u = \frac{1}{deg(u)(deg(u)-1))} \sum_{vw} (\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})^{1/3}.$$

- Centrality PageRank (CP): Calculated using the PageRank algorithm on each node; measures the importance of a node based on the importance of the nodes that connect to it.
- Density: Calculated as the ratio of the number of arcs to the maximum number of possible arcs; measures the degree of connectivity of the network.
- Giant Component: Calculated as the size of the largest connected component; measures the connectivity capacity of the network, in Strong (taking only direct arcs) or Weak variant.

Note: I used the weights of the arcs (distance between stops) as weights for the calculation of centralities, so as to take into account the actual distance between stops and not just their connectivity.

2.2 Grade analysis

The first analysis I performed was that of node degree. The degree of a node is the number of arcs incident on it (both direct and non-direct), and represents the number of connections a node has with other nodes in the network. In general, a node with a high degree is more important and influential than a node with a low degree, as it has more connections and influences more nodes.

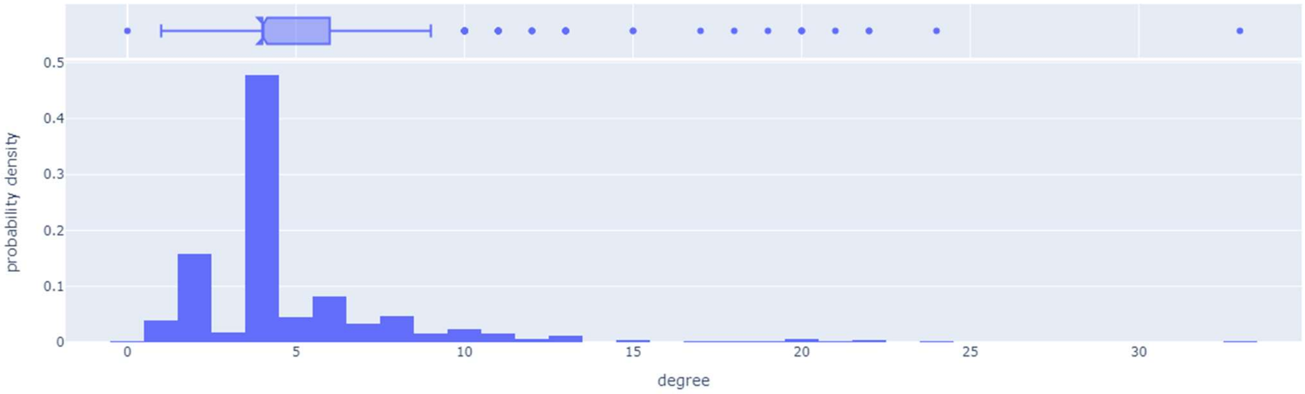


Figure 3: Degree distribution of nodes

As can be seen from the graph, most of the nodes have a low degree between 1 and 9, with the most common degree equal to 4; in addition, there are few nodes with a high degree (>= 10) and even fewer with a very high degree (>= 20), also the maximum degree is 33. We can also note a special case of a node with grade 0, which probably represents a stop registered in the GTFS but not yet activated by the agency.

So, we can say that the distribution reflects expectations, in that most stops are local and have few connections, while few stops are interchange and have many connections.

2.3 Analysis of centrality

The centrality of a node is a measure of its importance within the network, and there are several types, each with a different meaning. In general, a node with high centrality is more important and influential than a node with low centrality, as it has more connections and influences more nodes. I have used different metrics for its calculation (as described in section 2.1) as they provide different and complementary interpretations of a node's centrality.

2.3.1 Centrality Degree

It measures a node's ability to influence the network, and is calculated as the normalized degree of a node relative to the maximum degree; thus, the results are almost identical to those of the degree, except that the values are between 0 and 1.

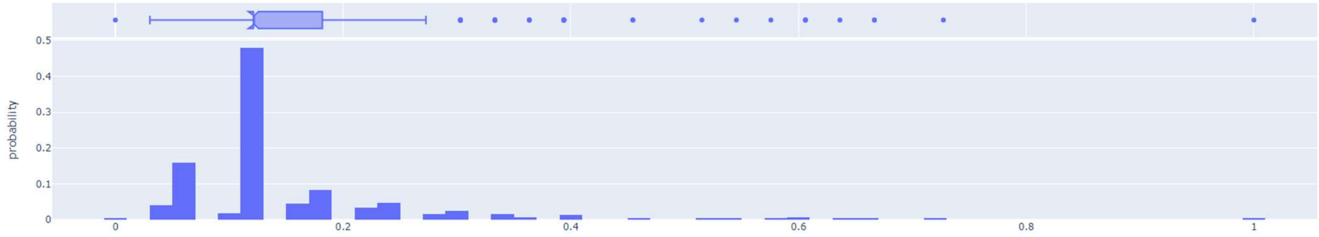


Figure 4: Centrality degree distribution

2.3.2 Centrality Betweenness

It measures the ability of a node to connect two parts of the network, and is calculated as the number of times a node is traversed by the minimum paths between two nodes.

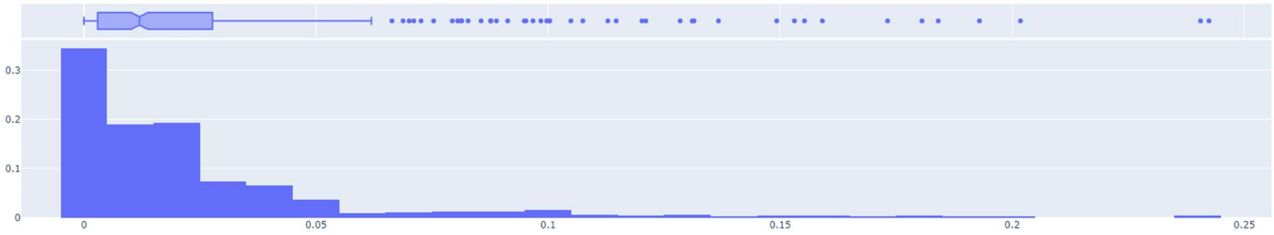


Figure 5: Centrality Betweenness distribution

The results show that most nodes have low centrality, with values between 0 and 0.05, while some nodes have much higher centrality. This allows us to distinguish between two types of nodes: hubs, which have high centrality and connect different parts of the network, and peripheral nodes, which have low centrality and do not connect different parts of the network.

2.3.3 Centrality Closeness

It measures the ability of a node to reach all other nodes quickly, and is calculated as the inverse of the sum of the minimum distances between a node and all other nodes.

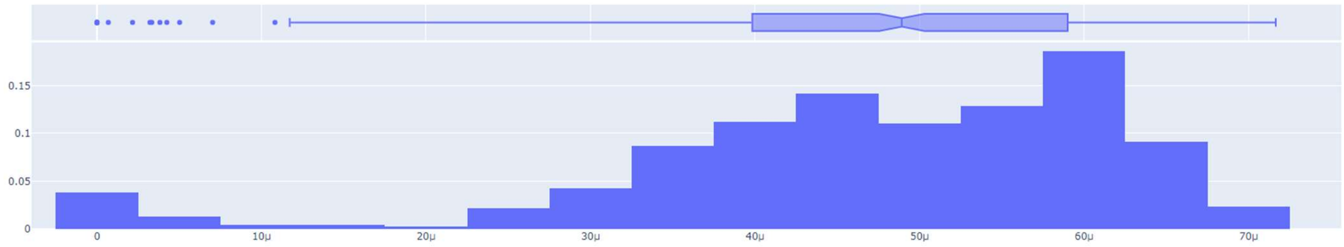


Figure 6: Centrality Closeness distribution

As can be seen from the graph, in general Closeness centrality is very low, with values ranging from 2.5e-5 to 7e-5. While this does not allow us to distinguish between hubs and peripheral nodes, it is characteristic of a high sprawl network such as the one under analysis, as the distance between stops is very high.

2.3.4 Centrality Eigenvector

It measures and assigns importance to nodes based on the importance of their neighbors, and is calculated by the eigenvector of the adjacency matrix; it is very useful, as usually, it provides a slightly different view than the other metrics.

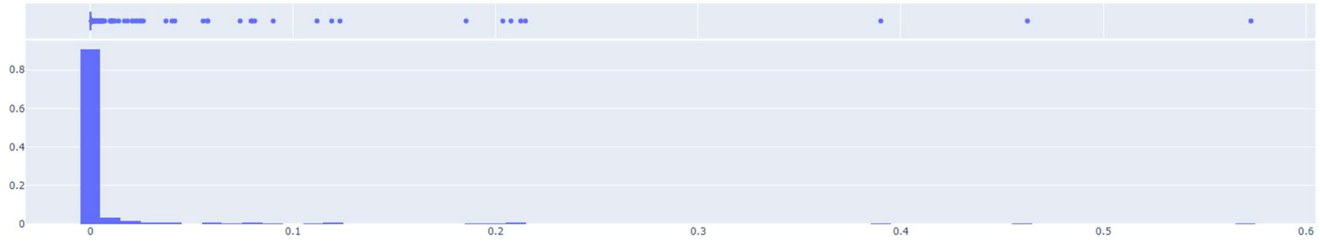


Figure 7: Centrality Eigenvector distribution

Centrality with Eigenvector is very similar to Betweenness, albeit much more skewed toward low values. This allows us to confirm that the network consists mainly of peripheral nodes, with few hubs.

2.3.5 Centrality Clustering Coefficient

It measures the degree of interconnection between a node's neighbors, and is calculated as the ratio of the number of arcs between a node's neighbors to the maximum number of possible arcs.

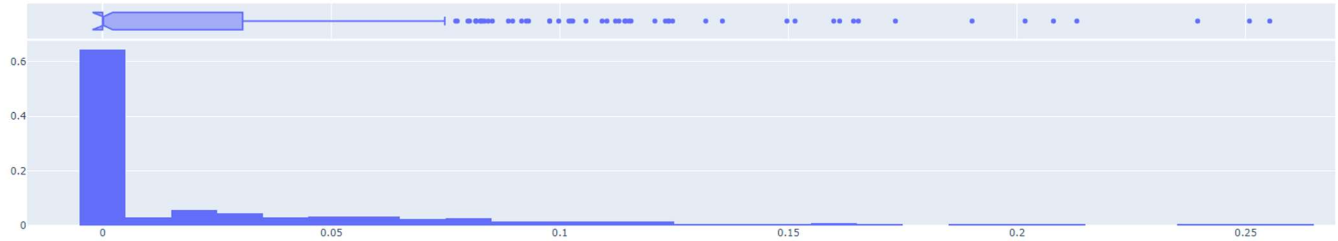


Figure 8: Centrality Clustering Coefficient distribution

This graph allows us to confirm that the network is mainly composed of peripheral nodes (a typical property of a high-extension network), as most nodes have a low clustering coefficient (60% have a value of about 0.005), however, there are some nodes with a much higher clustering coefficient, which could be hubs.

2.3.6 Centrality Page Rank

It measures the importance of a node based on the importance of its neighbors whose arcs are directed toward the node, and is calculated through the PageRank algorithm.

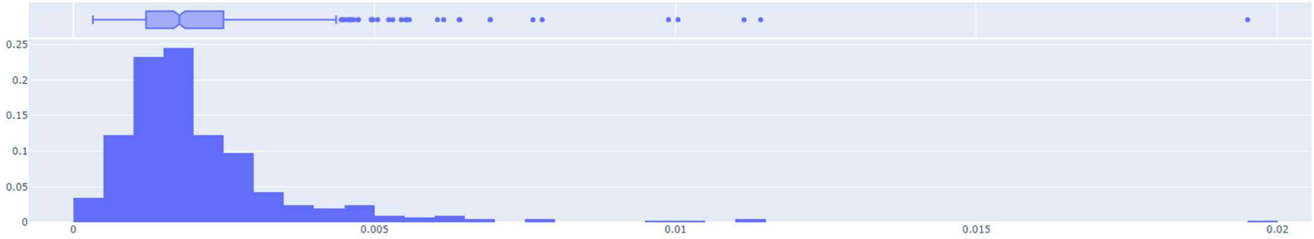


Figure 9: Centrality PageRank distribution

As can be seen from the graph, PageRank centrality has low average values (between 0.001 and 0.0025), with a few more central nodes (up to 0.005) and a few outliers reaching 0.02 at most. However, unlike the other metrics, PageRank centrality has a more even distribution.

2.3.7 Overall centrality

After considering each metric separately, I performed an overall analysis of centrality, first going to calculate the correlation between the metrics, so as to understand how well the information provided correlated with each other, and then going to calculate an overall index of centrality, so as to have an overall view of the centrality of a node. This was calculated on the relative metrics, i.e., scaled with their maximum, so as to have values on a uniform 0 and 1 scale (useful not for absolute assessment of centrality, but for relative assessment between nodes).

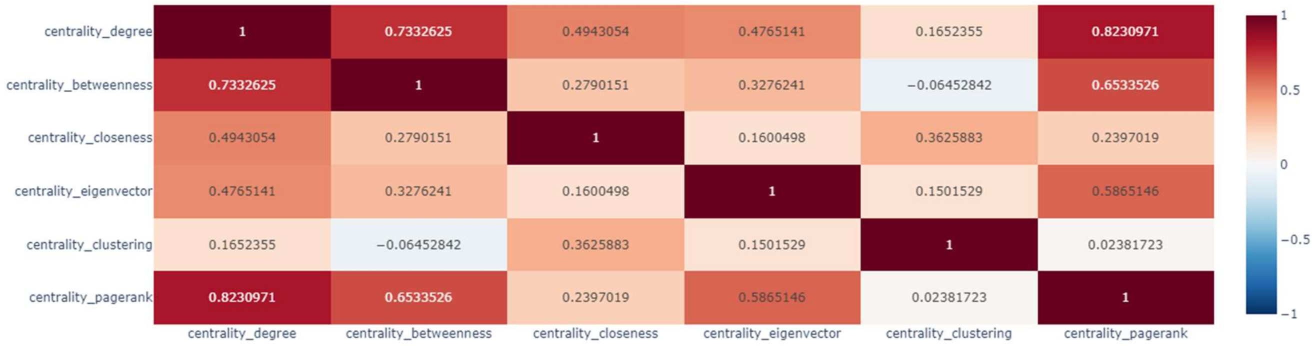
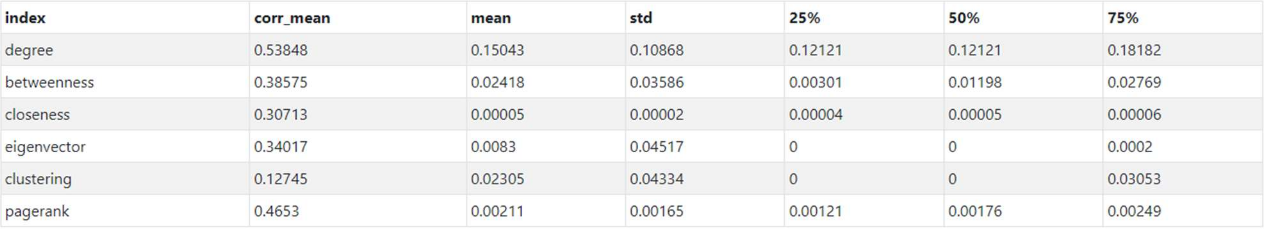


Figure 10: Correlation between centralities

In analyzing the correlation, I noticed that in general they are all correlated with each other (as expected), but there are some considerations that can be made:

- Degree, Betweenness and PageRank are the most related metrics, as they all depend on the number of arcs incident on a node.
- PageRank and Eigenvector are highly related to each other, as both depend on the importance of a node's neighbors.
- Clustering Coefficient is the metric least correlated with the others (even negatively with Betweenness), as it depends on the degree of interconnection between a node's neighbors.

Specifically, we can summarize the different metrics (and their correlations) with the following table:



index	corr_mean	mean	std	25%	50%	75%
degree	0.53848	0.15043	0.10868	0.12121	0.12121	0.18182
betweenness	0.38575	0.02418	0.03586	0.00301	0.01198	0.02769
closeness	0.30713	0.00005	0.00002	0.00004	0.00005	0.00006
eigenvector	0.34017	0.0083	0.04517	0	0	0.0002
clustering	0.12745	0.02305	0.04334	0	0	0.03053
pagerank	0.4653	0.00211	0.00165	0.00121	0.00176	0.00249

Figure 11: Summary table of metrics (absolute).

From which we can confirm the deductions made earlier, namely that: the network is composed mainly of peripheral nodes, and few hubs; the Degree and PageRank centralities are the most uniform, while the others are more skewed toward low values; and that the Clustering Coefficient centrality is the least correlated with the other metrics.

Finally, I calculated the overall relative centrality index as the average of the relative centralities weighted by the inverse of the correlation with the other metrics, so as to give more importance to the metrics least correlated with the others (which therefore have more information content).

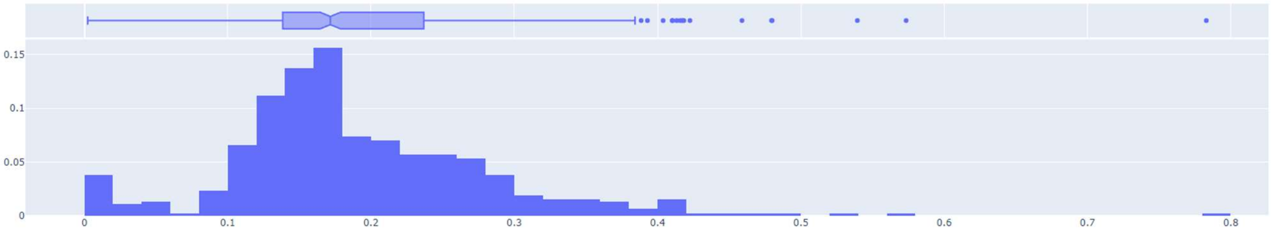
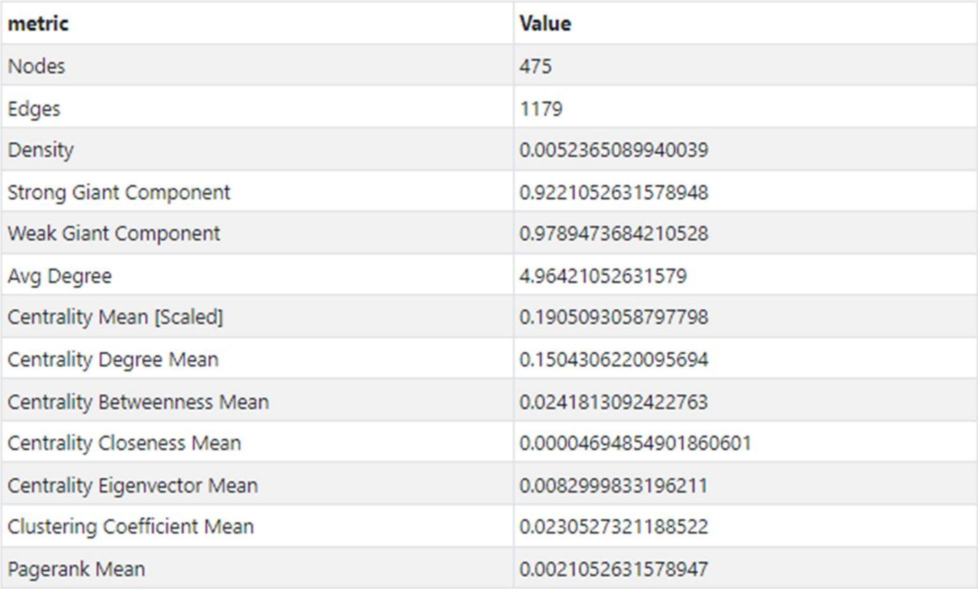


Figure 12: Centrality index

2.4 Overall network analysis

After analyzing the centrality of each node, it is useful to perform an overall network analysis using more general metrics. Specifically, in addition to considering the average of the centrality and degree metrics, I calculated the density of the network, and the size of the largest connected component (in the Weak and Strong variants).

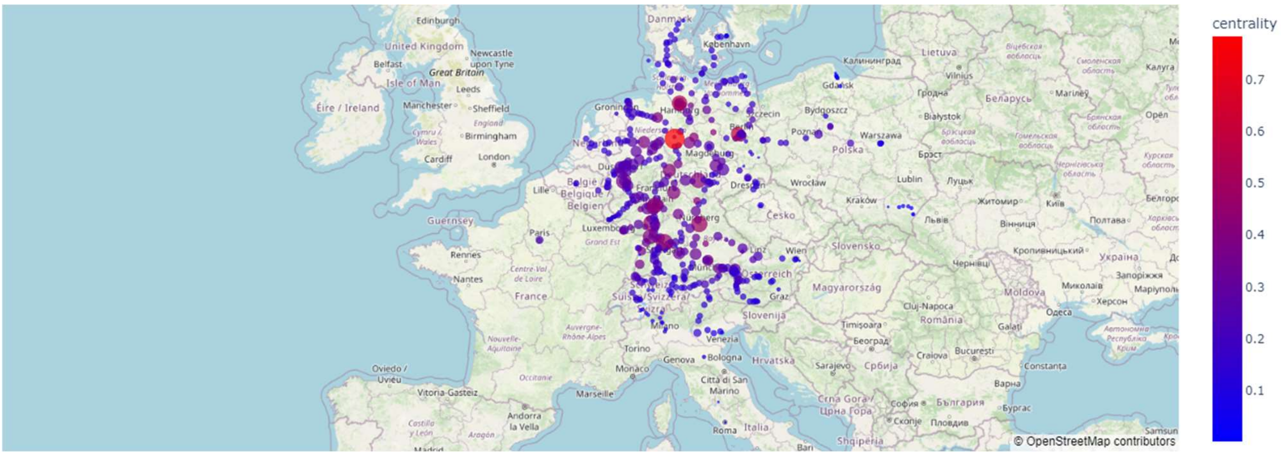


metric	Value
Nodes	475
Edges	1179
Density	0.0052365089940039
Strong Giant Component	0.9221052631578948
Weak Giant Component	0.9789473684210528
Avg Degree	4.96421052631579
Centrality Mean [Scaled]	0.1905093058797798
Centrality Degree Mean	0.1504306220095694
Centrality Betweenness Mean	0.0241813092422763
Centrality Closeness Mean	0.00004694854901860601
Centrality Eigenvector Mean	0.0082999833196211
Clustering Coefficient Mean	0.0230527321188522
Pagerank Mean	0.0021052631578947

Figure 13: Overall network table

From which we can see that the network has an average degree of 5, a very high size of the largest connected component (98% Weak and 92% Strong), an average centrality of 0.2 and low density (0.005). Thus, we can confirm that we have a very distributed network, not very compact, but very connected, with few hubs and many peripheral nodes. We can already infer that the network is very vulnerable to attacks targeting the most central nodes, but very robust to random attacks.

Figure 14: Map of stations. The size of the markers represents degree, while the color represents centrality.



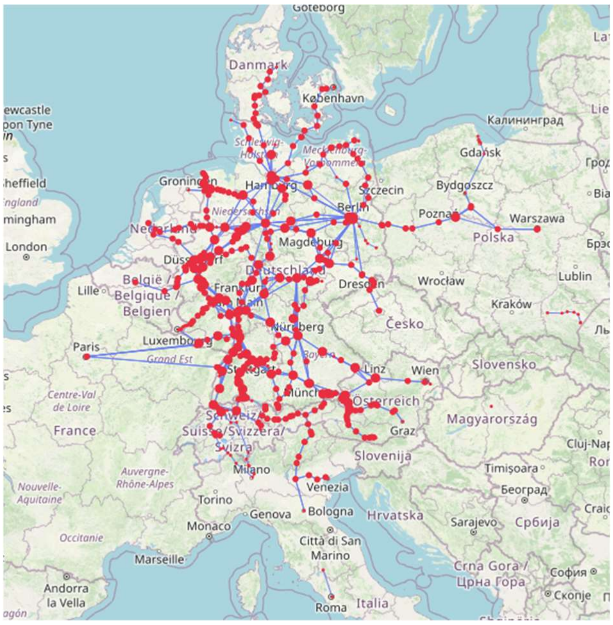


Figure 15: Map of stations and lines

As can be deduced from the two maps, most of the nodes are interconnected (there are only a few separate segments in Poland and Italy), and most of the Hubs are located in Germany (the main Hub is in Hannover).

Chapter 3: Network attack simulations

In this chapter I will simulate different types of attacks on the graph, and evaluate their impact on the network. Specifically, after describing the various types of attacks, I will run the simulations and evaluate the results.

3.1 Definition of attack and criteria

An attack is defined as an iterative process in which, at each step, one or more nodes (and their arcs) are removed from the network, and the impact of this removal on the network is evaluated, considering different metrics, until 10% of the remaining nodes are reached. These metrics will be displayed separately, each in its own graph, and overall, in a single graph in a relative manner, to assess the impact of the attack on the network.

The metrics considered to assess the impact of the attack are as follows:

- Edges: Number of arcs remaining in the network.
- Density: Density of the network.
- Strong Giant Component: Magnitude of the largest connected component, considering the direction of the arcs.
- Weak Giant Component: Magnitude of the largest connected component, not considering the direction of the arcs.
- Centrality mean: Average of the (relative) centralities of the remaining nodes.
- Avg Degree: Average degree of the remaining nodes.

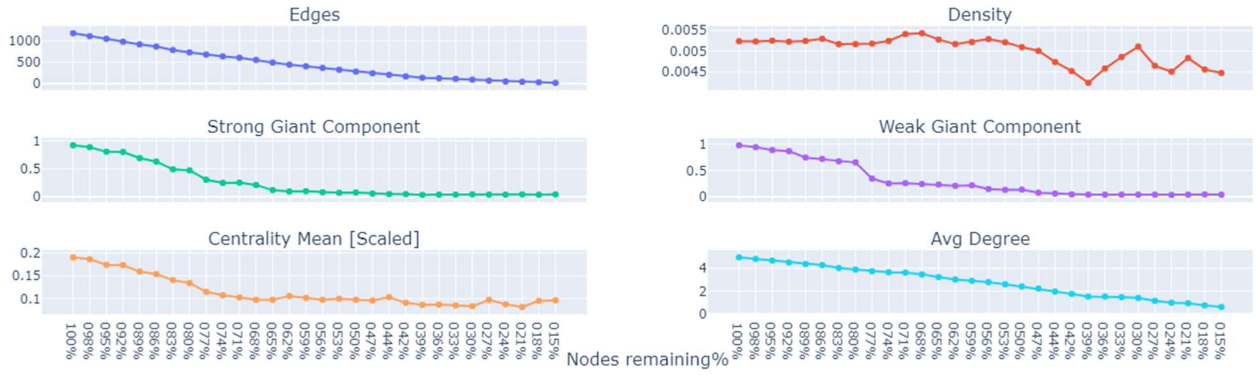
Also, in each attack the nodes to be removed will be chosen randomly (Random attack type); or, by taking the nodes with the highest centrality (from which we have, for each centrality, a different attack type).

3.2 Results of simulations

I simulated both Random attacks and targeted attacks on nodes with the highest centrality, for each centrality metric. The results of the simulations can be browsed through the demo, here I will report and analyze the most interesting ones and finally the overall ones.

3.2.1 Random Attack

Figure 16: Random attack simulation, graph with specific metrics.



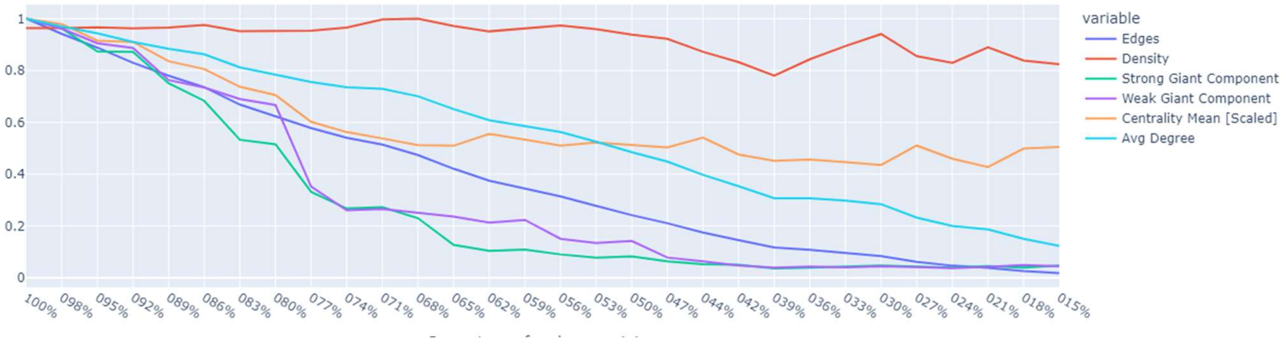
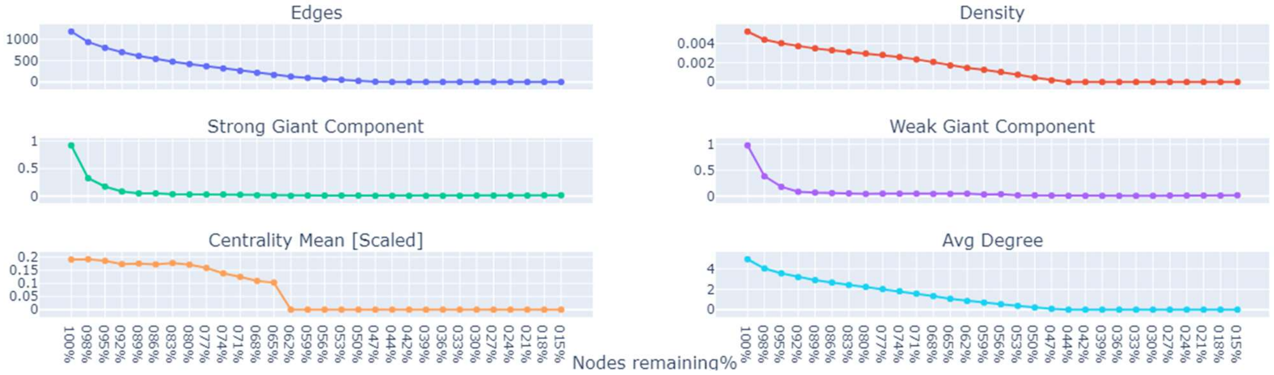


Figure 17: Random attack simulation, summary graph.

As expected, the random attack does not have a significant impact on the network as it is highly connected; in fact, even by removing 10% of the nodes (about 47) the network remains highly connected (the size of the largest connected component remains at 85%), and the average centrality only drops to 85% of the initial value (0.16); however, by removing 20% of the nodes the network reaches 50% connectivity, and the average centrality drops to 70% of the initial value (0.133); finally by removing 50% of the nodes, we notice that connectivity drops below 10%.

3.2.2 Attack Centrality Degree

Figure 18: Degree attack simulation, graph with specific metrics.



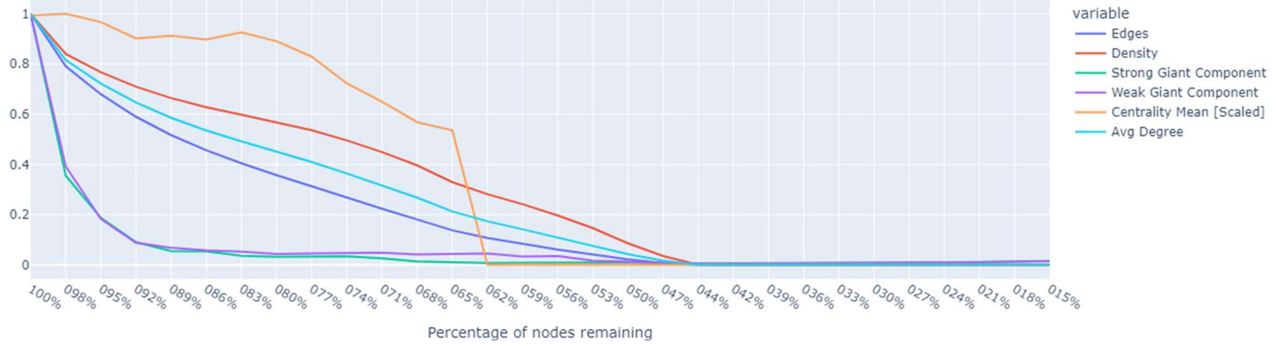
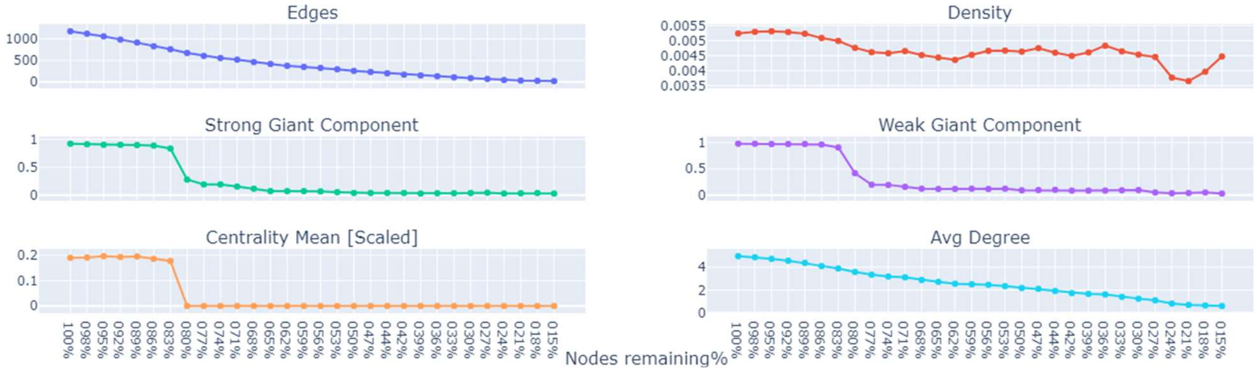


Figure 19: Degree attack simulation, summary graph

Analyzing the attack on nodes with the highest degree centrality, we notice that the attack has a much more significant impact on the network. In fact, removing only 2% (28 nodes) brings the network up to 35% connectivity, even though centrality remains almost unchanged; Furthermore, while the other statistics go down incrementally, we can see that centrality drops sharply between 30% and 40% of the removed nodes, from 65% to almost 0.

3.2.3 Attack Centrality Clustering Coefficient

Figure 20: Simulation attack Clustering Coefficient, graph with specific metrics.



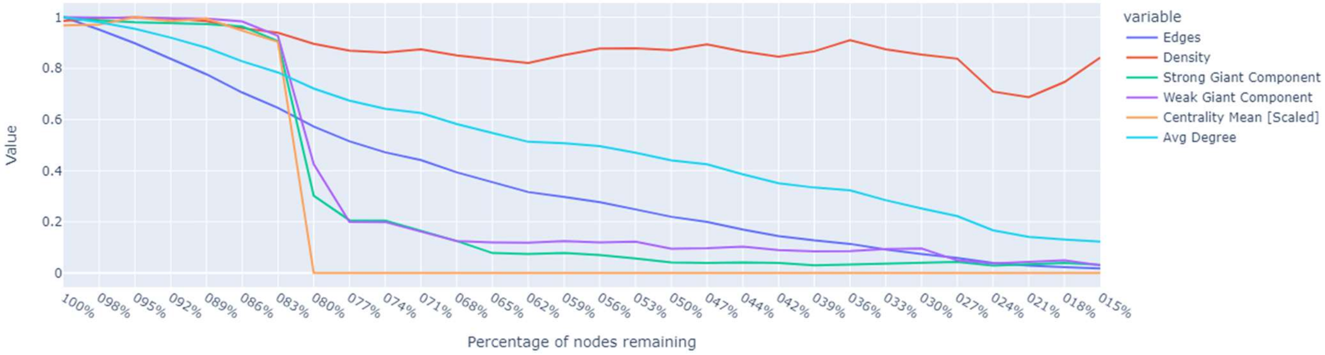
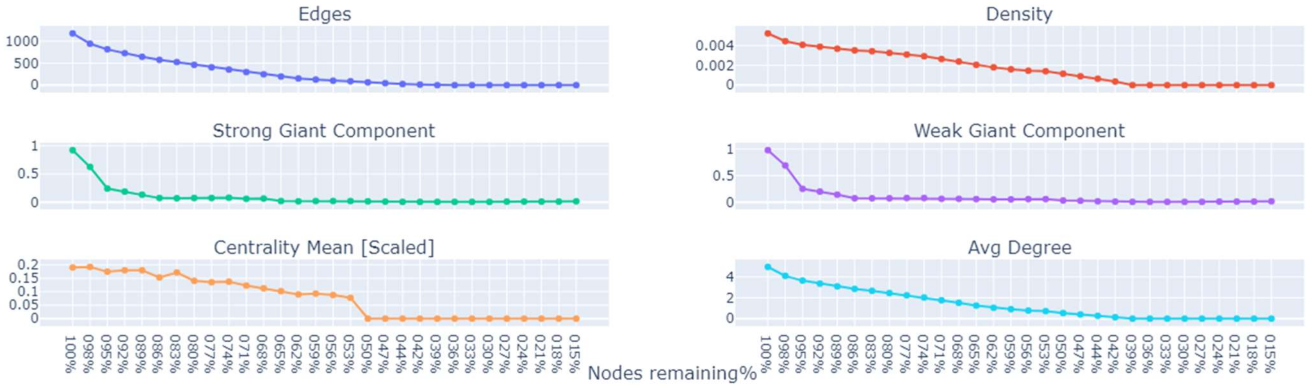


Figure 21: Simulation attack Clustering Coefficient, summary graph.

As we note from the graphs, the Clustering Coefficient type attack has a very particular impact (as expected, given the low correlation of the metric with respect to the others). In fact, we note that up to 20% of the removed nodes, both connectivity and centrality remain almost unchanged (92% and 90% of the initial value, respectively); on the other hand, after removing another 5% of the nodes (reaching 25% of the removed nodes), connectivity drops to 20%, while centrality drops to 0.

3.2.4 Attack Centrality Page Rank

Figure 22: Simulation Page Rank attack, graph with specific metrics.



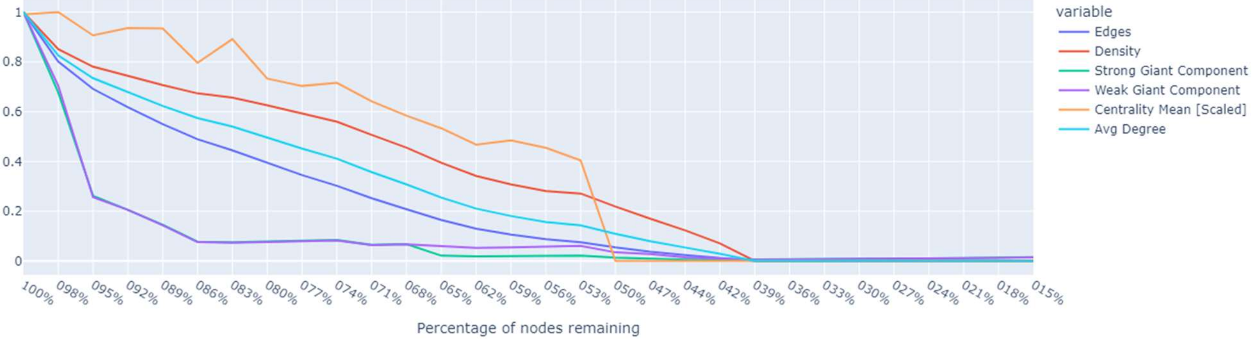


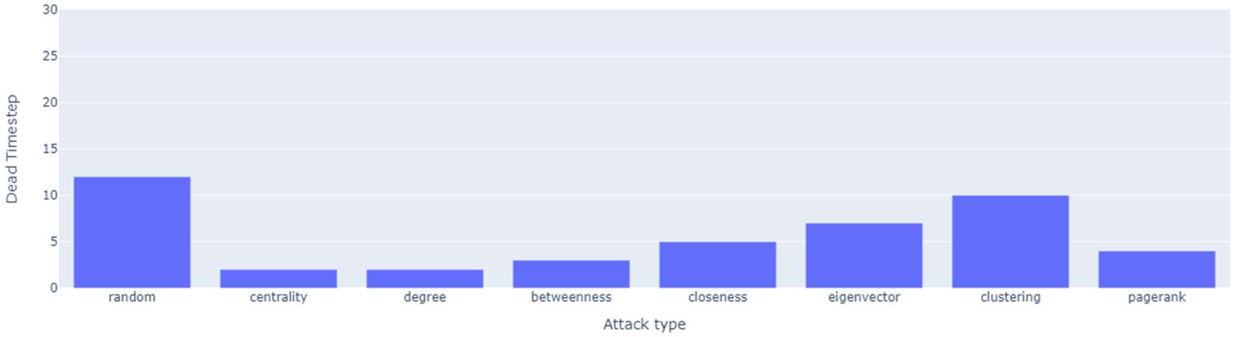
Figure 23: Simulation Page Rank attack, summary graph.

The PageRank attack has a very similar impact to the Degree attack, in that both metrics are highly correlated with each other. In fact, we note that the attack has a very significant impact on the network (albeit minor), and that connectivity drops to 20 percent after removing 5 percent of the nodes (70 nodes), while centrality drops incrementally up to 50 percent of the removed nodes where it then drops sharply to 0.

3.3 Overall analysis of attacks

To assess the impact of attacks on the network overall, I calculated the 'Dead Timestep,' which is the number of steps after which the network was significantly impacted (defined as a set change in a metric from the initial value). Metrics and variation are settable in the demo, but for this report I considered connectivity (Strong G.C.) and average centrality at 20 percent of the initial value.

Figure 24: Dead timestep for each type of attack, with Strong G.C at 20%.



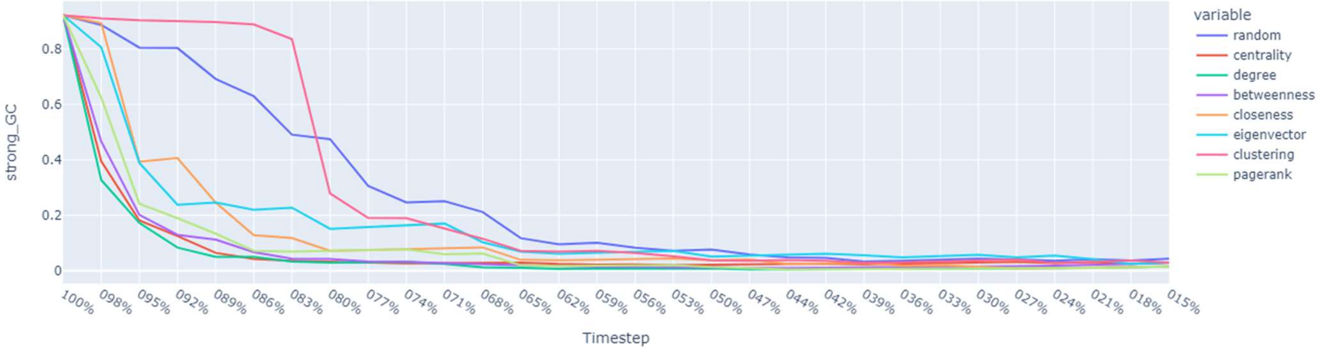
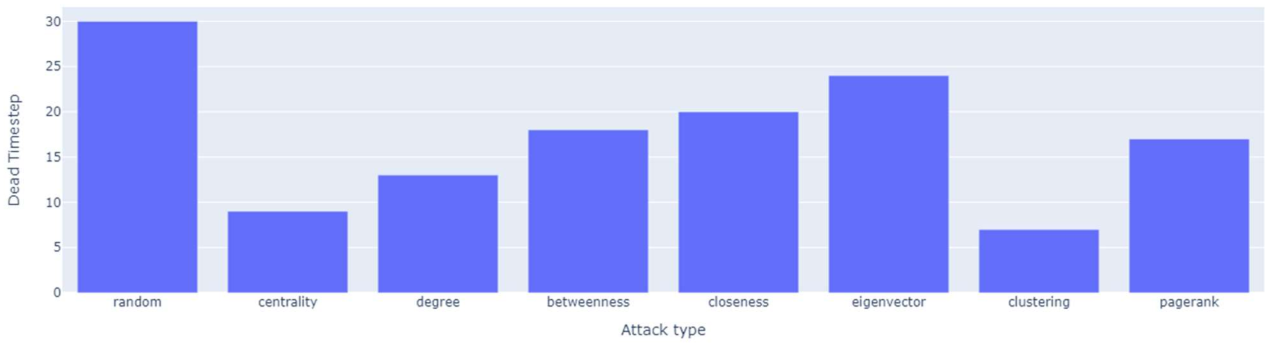


Figure 25: Strong G.C. for each type of attack at each timestep

We can see from this graph that each attack with at most 12 steps brings the network to the terminal state. In particular, we observe that:

- The Centrality, Degree, Betweenness, Closeness and PageRank attacks are very similar.
- Eigenvector, Clustering, and Random attacks have much less impact on the network, with random (as expected) having the least impact.

Figure 26: Dead timestep for each type of attack, with Centrality at 20%.



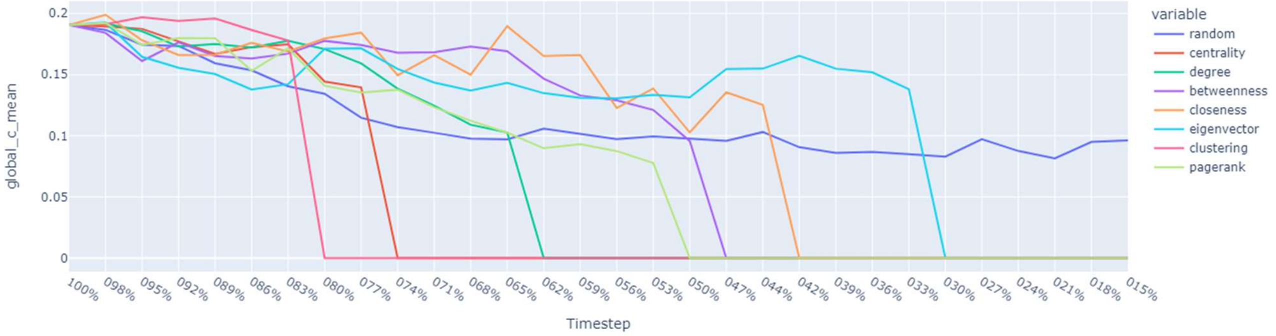


Figure 27: Centrality for each type of attack at each timestep.

For Centrality, however, we observe somewhat different results:

- Centrality and Clustering attacks now have a very similar impact, with even clustering having a greater impact.
- Degree, Betweenness, Closeness and PageRank attacks have less impact on the network than before (but still significant).
- The Eigenvector attack has the worst impact on the network (excluding random), but relatively in line with the others.
- The Random attack after all the timesteps has not yet had a significant impact on the network.

Note: Some centrality statistics (Degree, Closeness, Eigenvector) are not optimal for assessing the impact of attacks on the network; as, they are calculated and scaled with respect to the remaining nodes, and thus do not take into account the actual impact of the attack on the network. However, they were considered for completeness and comparison with other metrics.

3.4 Conclusions on attacks

From the simulations performed, we can deduce that:

The network is very vulnerable to attacks targeting the most central nodes; in contrast, random removal is much less effective.

- The best statistics for assessing impact are the Strong and Weak Giant Component, as they take into account the connectivity of the network, and the Average Centrality, as it takes into account the importance of the remaining nodes.
- The most impactful attacks are Degree, Betweenness, Closeness, and PageRank, as these are the metrics most correlated with each other and with network connectivity.
- During seizures, centrality tends to drop progressively to a certain point, where it then drops abruptly to 0.
- The removal of nodes according to the Clustering strategy has a particular impact, as it tends to hurt centrality more than connectivity (i.e., it is the inverse of the others).
- The Edges and Avg Degree metrics drop linearly, while they provide information, they are less meaningful for assessing the impact of attacks on the network.
- The Density metric in most attacks is not correlated with the number of nodes removed, and thus is not significant (except in the case of Degree and PageRank).

Conclusion

In this project, starting with a dataset containing nearly 2000 GTFSs, I briefly analyzed the GTFSs and chose that of Deuthesche Bahn (BN), the German railway network, for analysis; then, I analyzed the structure of the graph and evaluated its robustness and vulnerability to different types of attacks.

The GTFS of Deuthesche Bahn (BN) after a phase of conversion to DataFrame and then to graph, was analyzed in terms of structure and connectivity. In particular, I used and compared different centrality metrics to assess the importance of nodes and the connectivity of the network.

Next, I simulated different types of attacks on the graph, and evaluated their impact on the network. In particular, I simulated both Random attacks and targeted attacks on the nodes with the highest centrality, for each centrality metric. I then compared the results of the simulations and evaluated the damage caused to the network.

In conclusion, I deduced that the network is very large and connected, with few hubs and many peripheral nodes, and that it is very vulnerable to attacks targeting the most central nodes (particularly Degree, Betweenness, Closeness, and Pagerank attacks), but very robust to random attacks. Furthermore, I observed that the best statistics for assessing the impact of attacks are the Strong and Weak Giant Component, as they take into account the connectivity of the network, and the Average Centrality, as it takes into account the importance of the remaining nodes.

In the future, other elements of the arches (besides length) such as travel time, capacity, etc., could be considered for more precise centrality analysis. Also, other types of couplings could be considered, such as those based on the number of people carried or other selection criteria. Finally, one could extend the demo to work on other GTFS (since the structure is not strictly dependent on the GTFS chosen), and to display other metrics and simulate other types of attacks.

Appendix A: Some technical details.

The demo, in order to run, needs some data placed in the app/data folder; therefore, to run the code from scratch, you must first run the install.bat file (or install the libraries in the requirements.txt), which will install the libraries; and then, the main.py file (which expects the sources.csv in the ./data folder), which will take care of creating the missing folders, downloading the data, and preparing the files needed for the demo.

Among the various libraries adopted, I used one, named Peartree, to convert the GTFS format DataFrame to graph; however, I had to make some changes to the code to remove some bugs and to adapt it to my needs.

To start the demo, you need to run the index.py file, which will take care of starting the server and dashboard.

Appendix B: Dashboards and Demos

The Dashboard is divided into 4 pages:

- Home: Home page presenting the map of stations and their statistics; it is suggested to use that one to view only node statistics.
- Graph Evaluation: Page that exposes the structural analysis done to the graph.
- Attacks Analysis: Page presenting simulations of attacks made to the graph, with the ability to change metrics and variations to assess the impact of attacks on the network.
- Demo: Page presenting the demo, where you can view and interact with the graph, and simulate attacks manually.

Specifically, the demo, in addition to displaying the complete graph, allows you to:

- Visualize nodes and lines, with related information and statistics.
- Examine and compare nodes in detail, using the two side tables (one main and one comparison) that can be opened with the relevant arrows.
- Eliminate nodes and lines one by one, taking advantage of the comparison table to assess the impact on individual elements.
- Eliminate multiple items simultaneously in order to simulate attacks manually.
- Increase and decrease the size of nodes with higher degree.
- Reset or save the changes made to the graph (in the ./app/data/users folder).
- Use the table of states (below the map) to view graph statistics after each removal.
- Use the "Manual Attack Results" section (below the table) to view the trend in the various states of the graph statistics.