Robustness of the European Train Network

Matteo Bonacini

Abstract

In this work I investigate the robustness of the European train network, modeled using passenger route data from the Trainline online platform. Degree distribution analysis reveals a scale-free topology characterized by a degree distribution exponent of $\gamma = 2.23$, placing the network within the ultra-small world regime. This configuration supports high resilience to random failures, with connectivity preserved even when up to 95% of nodes are removed. However, the scale-free nature of the network also indicates vulnerability to targeted attacks: simulations show that removing just 20% of the highest-degree nodes can lead to network fragmentation, and the same is true for edge attacks. Community detection reveals the existence of four major communities which are locally more robust to targeted attacks and contribute to delay the effective destruction of the network. These findings highlight critical strengths and weaknesses within the European train network and suggest that more insight could be obtained by performing further studies.

CONTENTS

1 Introduction

The resilience and efficiency of transportation networks are critical to societal infrastructure, especially in interconnected systems like the European railway network, which facilitates daily commuting, economic trade, and tourism.

In this work, first I will construct a model by using the passenger train data available on the online platform *Trainline*. The resulting network can be thought of as the network of all the train routes that European train companies serve. Even though its nature is different from the real-world railway infrastructure network, studying its robustness can be proven useful nonetheless, as it could allow companies to identify possible areas of improvement in their train routes.

Then, by using this model, I will show how the network handles in case of failure of an arbitray number of its nodes. I will present both theoretical results and numerical simulations, and I will compare them with previous works on a similar network. In order to best introduce the theoretical results, I will explain some concepts regarding network degree distribution, small world and scale free networks, node degree correlation and community subdivision.

Finally, I will present a summary of the result obtained and give ideas for further research.

2 Extracting railway network data

Nowadays, most passenger train companies provide online platforms on which people can buy train tickets. These platforms usually contain sufficient data to reconstruct some form of the railway network:

- train numbers,
- station names.
- train routes between stations.
- travel time from one station to the next.
- network delays and failures.

One important piece of data that is *not* available to the public is the physical track layout and throughput. This means that, if one were to use data from this source, the resulting network would be made up of *train routes*, rather than actual *railways*. Even if some results will be similar, one must not forget that the two are very distinct entities. Train routes are managed by different companies; they can vary quiclky in time due to demand and cost requirements. Railways, on the other hand, are physical structures that cannot be altered, unless a significant investment of time and money is made.

2.1 Trainline

Trainline is a UK-based train-ticket company. They define themselves as:

> We are Europe's leading train and coach app. To put it simply, we are a one-stop-shop for train and coach travel. Every day, we gather routes, prices, and travel times from over 270 rail and coach operators in 40 countries, so that everyone can buy tickets quickly and save time, effort, and money.[1]

What their company does, in essence, is gather all the data from the major European train companies and then sell it. As a matter of fact, when a person buys a ticket through their platform, a small commision gets paid to Trainline, as an exchange for the ticket availability data[2]. The sale of the data can also happen at a larger scale, through their private-access *Global Travel API*[3].

2.2 Scraping the data

2.2.1 Legal Issues

I was not able to obtain access to the Trainline API, and thus I had to manually scrape their data from their website. This came with the advantage that it was completely free, but with the drawback that it is a legal gray area. Although at the moment of writing their Terms of Service[4] do not appear to explicitly prohibit scraping, I consider the study I made in the following sections to be valid only for my personal use (that is, for the purpose of taking the Complex Networks exam).

2.2.2 The scraping process

Trainline offers a portal that contains a list of all of the stations reached by their routes[5]. From this webpage, it is possible to extract a list containing the URL for each station page by looking at its source code. Once we have this list, we can start downloading each page individually. From every station page we can then extract the information of the most relevant destinations that can be reached from that station. This whole process is described in detail in Figure 1.

2.2.3 Data in numbers

Before moving on, I want to point out again one subtlety in the data that I got. That is, Trainline offers information on two types of "station". Some are the *real*, physical stations and some other are *fictional aggregates* (which I will call *cities*) of some number of real stations, usually found withing the same city. This aggregate is generated by Trainline using some unknown algorithm. Even though I trust that this data is accurate, it might be difficult to track its source outside of Trainline. For my study, I constructed the network using only the fictional aggregates.

Now, let us have a brief look at the data I got. In total, I have downloaded $18,502$ webpages ($4.95GB$ of data). Out of these, 248 were cities. The total number of links reported between the cities was 1976.

1. https://www.thetrainline.com/about-us
2. https://www.trainlinegroup.com/what-we-do/business-model/
3. https://www.thetrainline.com/solutions/api
4. https://web.archive.org/web/20240716112036/https://www.thetrainline.com/terms
5. https://www.thetrainline.com/en/stations

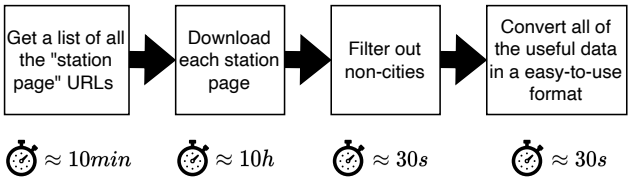


Figure 1: Overview of the scraping process. The first step is getting a list of URLs for all of the train stations. This is done by hand, by analyzing the code of the `/en/stations` page. The code contains one JavaScript array with the list of all URLs. With JavaScript it is easy to download each page programmatically. The maximum speed I was able to achieve was $\approx 20 pages/s$, but that would trigger the CloudFlare DoS protection. In the end I had to let the scraper run overnight, with a speed of about one page every two seconds. The last step is just a matter of runinng through each HTML file with a RegExp that filters out all the needed data. This is pretty straighforward, as all the data is contained inside a JavaScript object which can be easily parsed. The third step is explained in the following section (2.2.3).

2.3 Building the adjacency matrix

Each station on the Trainline website has a list of the most common stations that can be reached starting from it. With this piece of information, it is easy to construct a (directed) adjacency matrix A. Moreover, the data contains the average number of trains per day that reach said destinations, and the average time it takes to get there. We can use this information to construct a rate matrix W by multiplying each link by the ratio

$$\frac{\text{Number of daily trains through one route}}{\text{Average time it takes to complete the route}}.$$

The resulting matrix is plotted in Figure 15a and a zoomed in, cropped version of it is plotted in Figure 2. The rows and columns have been sorted in order of population of the respective cities. I find it nice to have a clear, human-readable plot of the (unweighted) data, and I provided it in Figure 16.

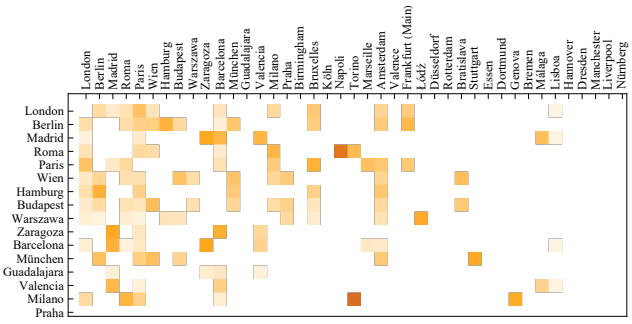


Figure 2: Detail of the adjacency matrix of the European train network, sorted by city population. Dense connections are evident among major hubs (left side of the matrix), indicating high connectivity, but they decrease with smaller cities.

The resulting matrix shows a gradient from left to right. This is a consequence of the fact that the Trainline data shows only the most common routes that start in any one city. It follows that

1) smaller cities will often display link information to bigger cities (like country or state Capitals), and
2) bigger cities will often display link information between themselves.

In the zoomed-in graph of Figure 2 it is easy to see this phenomenon: most links to the left of the matrix (where the big capital cities are) are two-ways and the matrix gets sparser and sparser the more we look to the right. This phenomenon indicates bias in the data, but it can easily corrected by making the assumption that *every train that leaves a destination must come back to it*. We can then make the rate matrix symmetric by mirroring it across the diagonal[6]. The resulting matrix is plotted in Figure 15b.

3 Degree-distribution properties

For each node i in a undirected network, we can define its *degree* k_i as the number of edges connected to it. Formally, if A is the adjacency matrix of the network, we define:

$$k_i = \sum_j A_{ij}.$$

Many useful properties of a network can be inferred by looking at its degree distribution. In this section, we will show some of these properties, and how they apply to the Trainline network.

3.1 Degree distribution

There are two possible degree distributions that concern us:

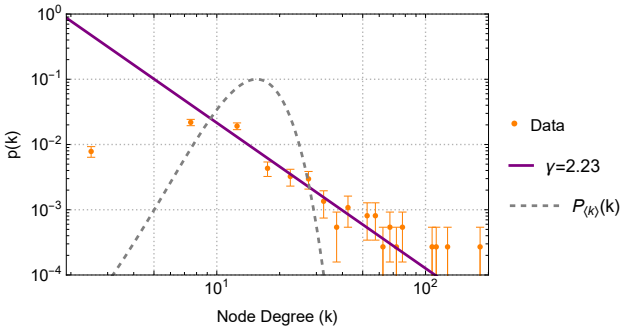
1) Poisson distribution and
2) Power-law distribution.

A network whose node degrees follow a power-law distribution is said to be a *scale-free* network [1]. Scale-free networks are prominent in nature and they exhibit some peculiar properties that we will briefly discuss below. The Poisson distribution, on the other hand, is characteristic of random networks [2], [3] and it constitutes the null hypothesis. If we were to find that the Trainline network followed a Poisson degree distribution, it would probably mean that there is something wrong with the data.

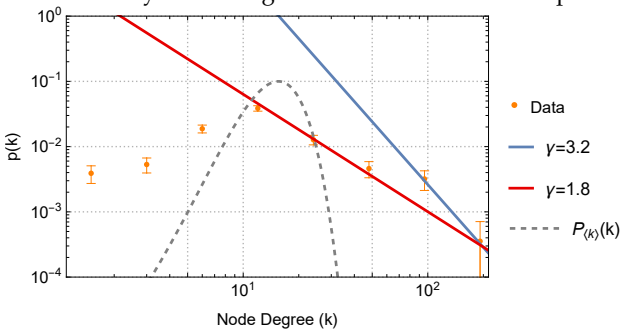
The best way to look at the degree distribution is by plotting it on a double-logarithmic plot; if the data points show a linear behavior for several orders of magnitude, then it means that the network is scale-free[7]. Given that the Trainline network has a relatively small number of nodes (for comparison, the WWW has

6. If a cell W_{nm} is empty, and the transposed cell W_{mn} is not, we set $W_{nm} = W_{mn}$. If both cells contain a value, we set each one of them to the average: $W_{nm} = W_{mn} = \frac{1}{2}(W_{nm} + W_{mn})$.

over $300,000$ nodes [4], while the network I am using only has 248), I have had some trouble in finding a fit for the data points. The plot, the fit result(s) and the problems I encountered are all explained in Figure 3. In the end, I concluded that the network shows a scale-free behavior $p(k) = Ck^{-\gamma}$ with parameters $C = 3.66$ and $\gamma = 2.23$. These results are consistent with other results obtained in this paper and their meaning is discussed in the following chapter.



(a) Data aggregation using linear binning. We show the best fit obtained by discarding the first two and last four points.



(b) Data aggregation using logarithic binning. Two possible fits are shown: one which takes into consideration only the last two points (blue) and one which takes into consideration the last five points (red). The latter has a value of $\gamma = 1.8$ which is anomalous (see section 3.2)

Figure 3: Node degree distribution and fitting. The first plot (a) used linear binning to aggregate the data, while the second plot (b) used logarithmic binning. The results are different and, moreover, the values of γ obtained in (b) are anomalous. When fitting these distribution, it is advised to use logarithmically-spaced bins; in this scenario, I believe the results which I obtained by using linearly-spaced bins are more reliable. In both plots, the dashed gray line is the best fit for a Poisson distribution (null hypothesis).

3.2 Scale-free property

By definition [5], a scale-free network is a network whose degree distribution follows a power law; that is:

$$p(k) = Ck^{-\gamma}. \tag{1}$$

7. A goodness-of-fit test would be in order to determine if the distribution is, in fact, a power-law.

In Nature we find that scale-free networks emerge in a wide variety of systems, such as: the internet [6], protein networks [7], metabolic networks [8] and e-mail networks [9]. The importance of scale-free networks arises from the fact that they exhibit some universal properties, that can be characterized solely by the exponent γ. The main difference between a scale-free and a random network lies in the high-k tail of the distribution. This difference is illustrated in Figure 3: here, we can see that the probability of finding a high-degree node in a scale-free networks is by orders of magnitude higher than in a random network. These high-degree nodes are often called *hubs* and their existence plays a crucial role in determining both the robustness of a network and its small-world properties. In Figure 4 I have shown how the node degrees are spatially distributed for the Trainline network. From this plot, it is easy to see that there are many, high-degree hubs spread troughout the whole network (they coincide with the busiest cities, as intuitively one would expect).

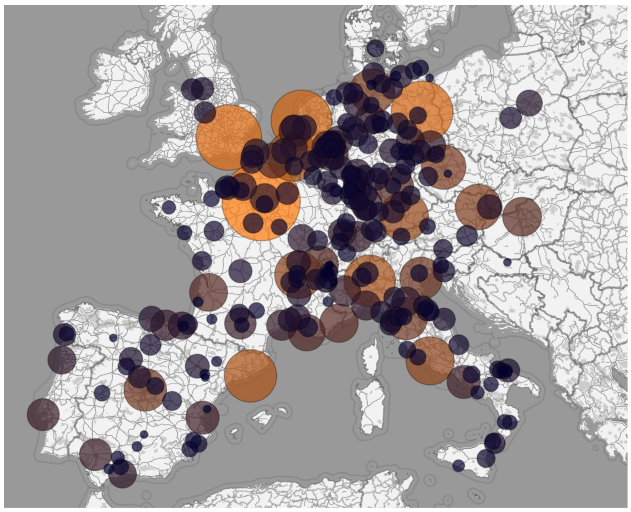


Figure 4: Visualization of the node degree distribution. The high-degree hubs correspond to European capitals and major cities. Each circle's color and size scale with the degree of its respective node.

3.2.1 *Meaning of scale-free*

The origin of the term *scale-free* comes from the fact that, for certain values of γ, the moments $\langle p(k)^n \rangle$ of the distribution (1) diverge if $n > 1$. Since the expectation value for a random variable k exhibits fluctuations on the order of $\sqrt{\langle k^2 \rangle}$, having this value diverge would mean that, if we were to choose a node at random, we could expect it to be both arbitrarily small or large. This is in contrast to what we find for random networks, where the second moment is $\langle k^2 \rangle = \langle k \rangle^2$ and thus, the fluctuations are on the order of $\sqrt{\langle k \rangle}$. While in the latter case the average degree quantifies a scale for the fluctuations, in the former we cannot really define a scale due to the divergence of the second moment.

3.2.2 (Ultra) small world properties

Many of the properties of a scale-free network depend on the value of the exponent γ. In this section, I will briefly discuss how γ influences the *small-world* property of a network; that is, the comparatively slower scaling of the average diameter $\langle d \rangle$ of a network with respect to its number of nodes. In other words, a network is said to exhibit small-world properties if the average distance between two nodes is relatively small, even if the number of nodes is very large. In the next sections, I will also show how the robustness properties of a scale-free network depend on γ. A more in-depth explaination of these results can be found in [5], and all of them can be obtained analytically [10], [11].

- $\gamma < 2$: **anomalous regime (I)**. Real, large networks cannot really follow this scaling law, as for $\gamma > 2$ we have that $\langle k \rangle$ diverges for large N. This would imply that the degree of the largest hub grows faster than the size of the network, which is not possible by definition.
- $\gamma = 2$: **anomalous regime (II)**. In this regime, we have that $\langle d \rangle \sim$ constant. This means that the degree of the largest hub is approximately equal to the number of nodes in the network. This forces the network into a *hub-and-spoke* configuration, in which all nodes are connected to a single, centralized hub.
- $2 < \gamma < 3$: **ultra-small world/scale free regime**. Here, we have that $\langle d \rangle \sim \log \log N$; moreover, the first moment of k converges, but all of the other moments diverge. In this regime, we find the presence of many high-degree hubs that are connected to small-degree nodes. This phenomenon effectively shrinks the distance between all the nodes of the network.
- $\gamma = 3$: **critical point**. Here $\langle d \rangle \sim \frac{\log N}{\log \log N}$, which is slower than the ultra-small wold regime but faster than the small-world regime.
- $\gamma > 3$: **small world/random network regime**. Here, $\langle d \rangle \sim \log N$ and $\langle k^2 \rangle$ converges. Hubs continue to exist, but they are not sufficiently large to make a significant difference with respect to a random network of the same size.

3.3 Discussion of the fit results

The proportionality constant C of (1) can be obtained by imposing the normalization condition:

$$\sum_{k=1}^{+\infty} p(k) = 1.$$

Node degrees can only assume discrete values but, if we consider the limit in which the number of nodes is large, we can switch to the continuum formalism and compute C more easily:

$$\begin{aligned} C &= \frac{1}{\int_{k_{\min}}^{+\infty} k^{-\gamma} dk} \\ &= (\gamma - 1) k_{\min}^{\gamma - 1}, \end{aligned} \tag{2}$$

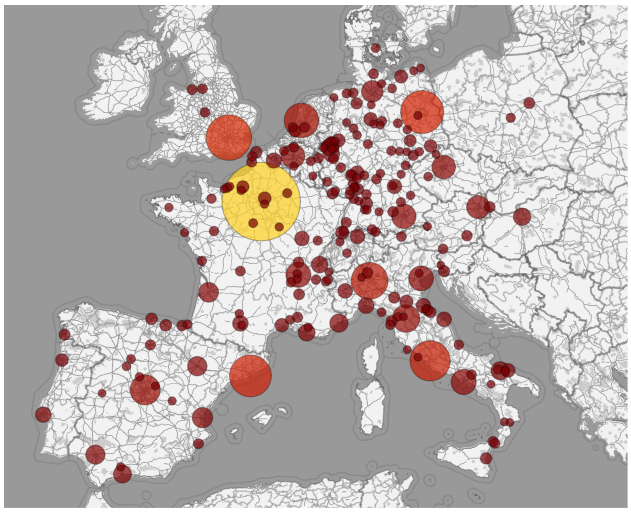


Figure 5: Visualization of the node betweenness centrality distribution. The nodes with highest centrality correspond to European capitals and major cities. Each circle's color and size scale with the betweenness centrality of its respective node.

where $k_{\min}$ is the smallest degree for which we expect the power law (1) to hold. The values of $k_{\min}$ and γ can then be found by performing a fit on the data; the proper methodology can be found on [5].

By plugging the fit result obtained in the previous section into (2), we obtain the estimate: $k_{\min} = 2.42$ and $\gamma = 2.23$, which corresponds to the ultra-small world regime discussed in section 3.2.2. This result is exactly what we would expect to obtain from a good transport network: the ultra-small world property implies that any node (city) can be reached in a short time by any other node, even if the two nodes considered are small, scarcely-connected cities. Moreover, if we compute the mean graph distance of the Trainline network we obtain $\langle d \rangle = 2.28$, which is comparable to $\log \log N = 1.70$, as discussed in section 3.2.2. Graphically, we can get an idea of how the ultra-small world effect comes into play by looking at Figure 5. There, I have plotted the *betweennes centrality* $g(v)$ of each node v, which is a property related to the number of shortest paths that pass through it:

$$g(v) = \sum_{i \neq v \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}},$$

where $\sigma_{ij}(v)$ is the number of shortest paths from i to j that pass through v and σ_{ij} is the number of all shortest paths from i to j. The nodes with very high betweenness centrality represent the hubs which are responsible for shortening the distances across the network.

At least one previous work [12] has run the same analysis on a similar European train network [13], but they obtained a different result of $\gamma = 4.11$. The discrepancy of these results most likely comes from the different natures of the networks considered.

4 DEGREE CORRELATIONS

Degree correlations quantify the likelihood of a high-degree node to connect to a low-degree node and vice-versa. They appear in many real-world networks (such as, the Internet, biological networks and social networks [5]), but not in all of them (e.g. the power grid does not exhibit degree correlations [5]). The study of degree correlation can help shed light on some network properties; for the scope of this work, what concerns us is the relation between degree correlations and network robustness.

4.1 Possible cases

We can distinguish three possible types of networks:

1) **Neutral networks**. If there is no degree correlation, we say that a network is neutral. This is true, for example, in random networks.
2) **Assortative networks**. This is the case if there is positive correlation between node degrees. In these types of networks, high-degree nodes tent to link to other high-degree nodes, and small-degree nodes to other small-degree nodes.
3) **Disassortative networks**. This is the case if there is negative correlation between node degrees. Here, high-degree nodes tent to connect to low-degree nodes and vice-versa.

If the nodes of a network start randomly failing, we expect to see a phase transition at around some value of $\langle k \rangle$ for which the size of the giant component of the network falls to zero. Assortativity delays the phase transition in the following way:

- In assortative networks, the phase transition happens at a lower value of $\langle k \rangle$, meaning that a higher number of nodes must fail in order to lose the giant component.
- In disassortative network, the opposite is true, meaning that the network will break up after a smaller number of nodes is removed.

4.2 Measuring correlations

Mathematically, all the information about degree correlation is contained in the *degree correlation matrix* e_{ij}, which is defined as:

$$e_{ij} = \text{Probability of finding nodes with degrees } i \text{ and } j \text{ at the end of a randomly selected link.}$$

One can compute [5] e_{ij} explicitly for random networks. The result is:

$$e_{ij} = q_i q_j,$$

where q_k is the probability of finding a node of degree-k at the end of a randomly selected link, given by

$$q_k = \frac{k\, p(k)}{\langle k \rangle}.$$

From an operative standpoint, working using e_{ij} directly is quite cumbersome and thus, it is better to introduce the *degree correlation function*:

$$k_{nn}(k) = \sum_{k'} k' P(k'|k).$$

$P(k'|k)$ is the probability to reach a degree-k' node by following a link from a degree-k node and thus $k_{nn}(k)$is the average degree of the neighbors of all degree-k nodes.

The assortativity of a network depends on $k_{nn}(k)$ in the following way:

- For neutral netowrks, we expect it to be a constant: $k_{nn}(k) = \langle k^2 \rangle / \langle k \rangle$.
- For assortative networks, we expect it to follow a positive trend: $\frac{d}{dk} k_{nn}(k) > 0$.
- For disassortative networks, we expect the opposite: $\frac{d}{dk} k_{nn}(k) < 0$.

4.3 Results from the data

I have plotted the degree-correlation matrix in Figure 6a and, more importantly, the trend of the degree correlation function in Figure 6b. From the latter, it is immediate to see that the data follows a downward trend; this means that the network is disassortative and thus more vulnerable to failures. Again, this result is not compatible with the one obtained in [12], in which their network was observed to be assortative.

5 COMMUNITY SUBDIVISION

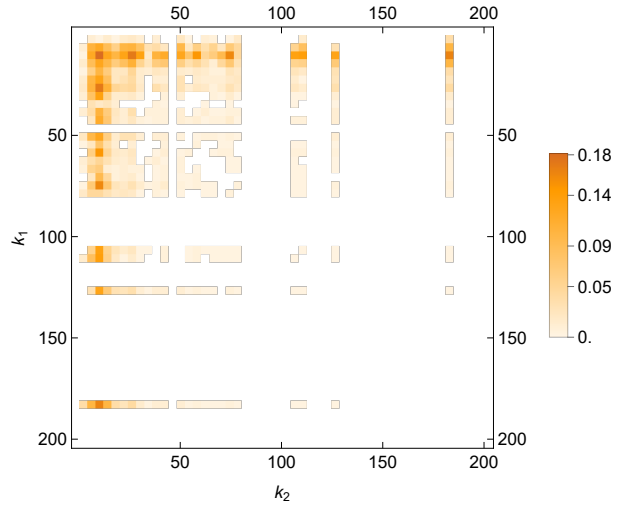
Real-world network tend to be subdivided in many, smaller communities. This phenomenon is found in social networks [14], biological network [15] and phone-call networks [16], and it has been of great interested for researchers since the early 2000s. One very good (and famous) example of the effect that the community subdivision can have on a network is the case known as *Zachary's Karate Club* [17]. In this paper, the authors applied an autonomous community detection algorithm to the karate club participants' personal-interactions network which was able to predict the exact way in which the network fell apart after one major conflict struck the club.

This example illustrates how community detection can be a useful tool to quantify the robustness of a given network and to potentially predict its modes of failure under stress. In this chapter, I will briefly explain how communities can be defined and I will show how the Trainline network is subdivided.

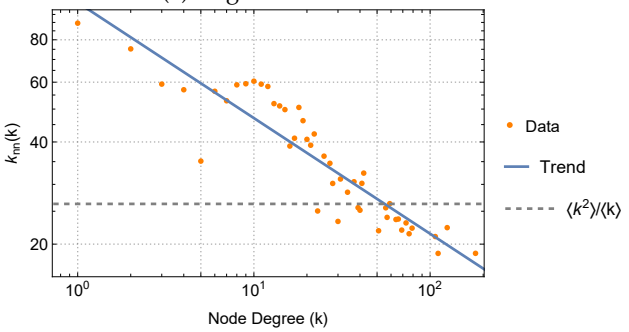
5.1 Defining communities

Currently, there is not any widely-accepted definition of *community*. Moreover, the mere existence of communities currently lacks any deep mathematical proofs and is based only upon empirical observations. Barabasi [5] defines communities by using a series of hypotheses:

1) Communities are structures uniquely encoded in a network's wiring diagram. The *ground truth*



(a) Degree correlation matrix



(b) Trend for the degree correlation function. The fit indicates a downwards trend that follows the power law $\sim k^{-.34}$.

Figure 6: Results of the degree correlation analysis: degree correlation matrix (a) and fit result for the degree correlation function (b).

of the community structure is to be discovered with algorithms.

2) A community corresponds to a connected subgraph, whose connections are denser than with the rest of the network.
3) Random networks do not have communities.

Algorithms for community detection are constantly being developed [18]; the main challenge for new algorithm is the difficulty in obtaining good results while maintaining a low computational complexity. Given that the number of possible partition in a graph scales exponentially with its number of nodes [5], a simple brute-force approach would not work.

Currently, the best scaling algorithm is the Louvain method [16], which scales linearly with the number of links ($\mathcal{O}(L)$). There exist other algorithms that scale polinomially with the size of the network, up to the clique percolation algorithm [19] which scales exponentially with the number of nodes ($\mathcal{O}(e^N)$). Ultimately, there does not exist a definitive *best choice*; each algorithm has its own strength and weaknesses and should be used only within its intended use cases.

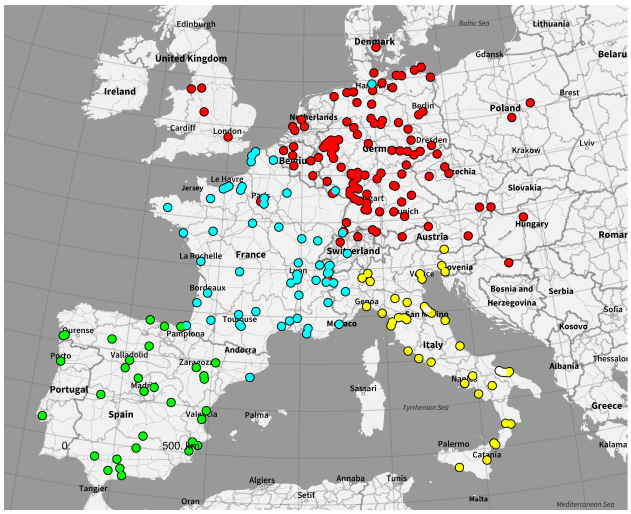


Figure 7: Community structure of the Trainline network. Four distinct communities are detected, roughly aligning with the geographic regions of Italy, France, Spain, and Germany. These regional clusters suggest that disruptions may cause localized isolation rather than complete network fragmentation.

5.2 Community detection results

I have run a community detection algorithm on the Trainline network and the results are displayed in Figure 7. I have used the Mathematica function `FindGraphCommunities` [20] with `Methods->"Hierarchical"` settings. The official documentation does not state which exact algorithm the function implements, but it is most likely one of the hierarchical algorithms discussed in [18].

Perhaps unsurprisingly, The algorithm managed to find four different communities which, for the most part, group the nodes which are situated in Italy, France, Spain and Germany. Given that, by definition, the link density is between community members is greater than with members of other communities, we can speculate that if the network were to face an attack, it would be less likely for a single community to immediately far apart, but more likely for different communities to remain isolated.

This is corroborated by the graph in Figure 8, which shows that the degree of a certain node and its betweenness centrality are positively correlated. This, in turn, means that the more important a node is (i.e. the more links a certain city has), the higher is the number of shortest paths that pass through it. If an attack were to target the most important nodes first, it would make the travel time between nodes larger, up to infinity; if we assume that nodes in the same community are on average closer together, it is not unreasonable to assume that the shortest path between them will diverge slower than between nodes that are part of different communities. The behavior of the network under an attack is discussed thoroughly in the next section.

We can also compare different centrality measures between each other (Figure 17). The result is that they

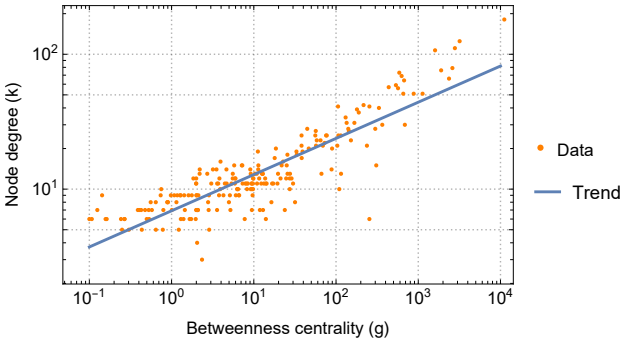


Figure 8: Relationship between node degree and betweenness centrality. A positive correlation between the two is found, indicating that highly connected hubs also serve as primary paths in the network. This relationship shows how hubs are crucial in keeping the different communities connected together.

are all positively correlated and, more importantly, they positively correlate with the population of each node's respective city, as one could expect.

6 Network Robustness

The goal of studying network robustness is to see how a network behaves under some kind of failure. The most common form of failure that is studied is the removal of one node from a network. If the node is chosen at random, we call this a *random failure* but if the node is chosen according to some heuristics, then we are talking about a *targeted attack*. After the removal many properties of the network will change, but the one that concerns us the most is the probability P_∞ with which a randomly selected node is part of the giant component. In this section we will discuss some theoretical results and compare them with failure simulations on the Trainline network.

6.1 Random failures

When we gradually remove nodes from a network P_∞ gets lower, up until it reaches zero. Under some condition, the network may undergo a phase transition and lead to P_∞ reaching zero with a sharp drop, way before all of the nodes have been removed. This phase-transition behavior has been studied thoroughly within the framework offered by *percolation theory* [21]. The results show that in a random network, the order parameter P_∞ follows

$$P_\infty(f) \propto (f_c - f)^{\beta_f},$$

where f is the fraction of nodes removed from the network, f_c is a value dependent on the lattice type and β_f is a *critical exponent*. For reference, the value of f_c for a two-dimensional square lattice is $f_c \approx .407$ and the value of β_f is $\beta_f = 5/36$ (which is universal for lattices of dimension 2).

When dealing with networks with an arbitrary degree distribution, however, we observe a different behavior. Relevant to us is the case of scale-free network, in which we do not see a sharp phase transition (for random failures, that is) but instead we find a steady, slow decline of P_∞. Actually it can be proven that, for a network with an arbitrary degree distribution, we have a critical threshold f_c on the fraction of nodes that we can remove, after which the network will completely lose its giant component. f_c is given by [22], [23]:

$$f_c = -\frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}. \tag{3}$$

Equation (3) predicts two different possible behaviors for scale-free network, depending on the value of γ:

- For $\gamma < 3$, $\langle k^2 \rangle$ diverges and thus we get $f_c \to 1$ in the $N \to +\infty$ limit. This means that, in order to fragment the network, we need to remove all of its nodes.
- For $\gamma > 3$ we have that f_c is independent of network size N, but depends only on γ and $k_{\min}$. This behavior, proven in [5], is the same as what we see in a random network: the network falls apart once a certain fraction of its nodes is removed.

I run a simulation on the Trainline network (Figure 9). Each data point in the plot is obtained by removing a certain fraction of nodes from the network and counting how many of the nodes are part of the giant component. This process is repeated 16 times for each data point and the graph reports the standard deviation of all the results. The behavior we see is exactly what we expect from a scale free network with $\gamma < 3$. This result is great, as it indicates that the Trainline network is resilient to random failures of its node. If, for example, a city were to be excluded from the network due to economical or environmental issues, the rest of the network would be able to keep working without any issues.

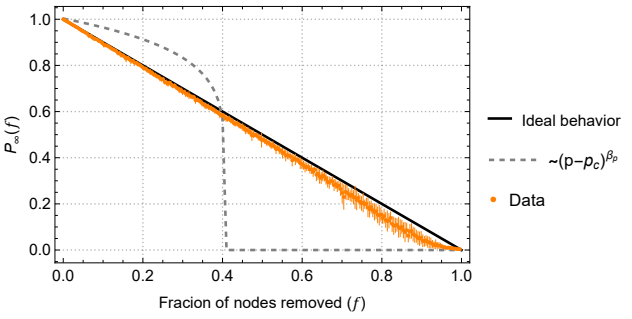


Figure 9: Simulation of resilience of the network to random failures. Multiple runs were performed in which a certain number of nodes chosen at random was removed from the network. The graph shows the fraction of nodes that are still part of the giant component after the removal. The dashed, gray line is what we would expect to see if the network was wired like a 2-dimensional square lattice (null hypothesis).

6.2 Targeted attacks

A scale-free network behaves differently if, instead of failing randomly, the nodes are attacked in order of importance.

6.2.1 *Attacking nodes*

We consider the case in which a potential attacker manages to disable the nodes in order of importance; that is, targeting the highest-degree node first. When this is the case, scale-free network do fail, and after only a small fraction of the nodes are removed [24]. It is possible to derive [25] the following equation, whose solutions are the critical fraction of nodes that need to be removed by an attacker in order for the network to break apart completely:

$$f_c^{\frac{2-\gamma}{1-\gamma}} = 2 + \frac{(2-\gamma)}{3-\gamma} k_{\min} \left(f_c^{\frac{3-\gamma}{1-\gamma}} - 1 \right). \qquad (4)$$

In our case, using the values of $k_{\min}$ and γ that we obtained above, we get that $f_c = .235$ is the only acceptable solution.

Again, we can confirm this behavior by running a simulation in which we remove the highest degree nodes, one at a time, and compute P_∞ at each step. The result (Figure 10) shows a fast, steady decline up to $f = .2$, after which we have a steep drop. At $f = .3$, more than 95% of the network is now unreachable. This behavior matches what the theory predicts and highlights one very big weakness of the Trainline network.

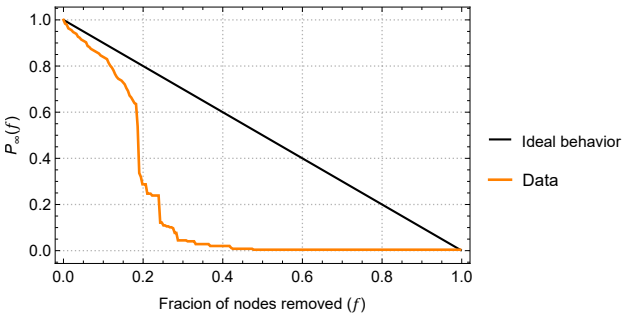


Figure 10: Simulation of resilience of the network to targeted attacks. One simulation was performed in which the highest-degree nodes were removed sequentially. The graph shows the fraction of nodes that are still part of the giant component after the removal.

6.2.2 *Surviving communities*

We can make one interesting observation by plotting the community structure of the network just after the big drop at $f = .2$ (Figure 11). From this we can see that, even if the network is now fragmented, the individual communities more or less managed to survive. A network in this state would still be able to carry out some of its work; this means that the current attack strategy is a little less effective than what the calculations predict.

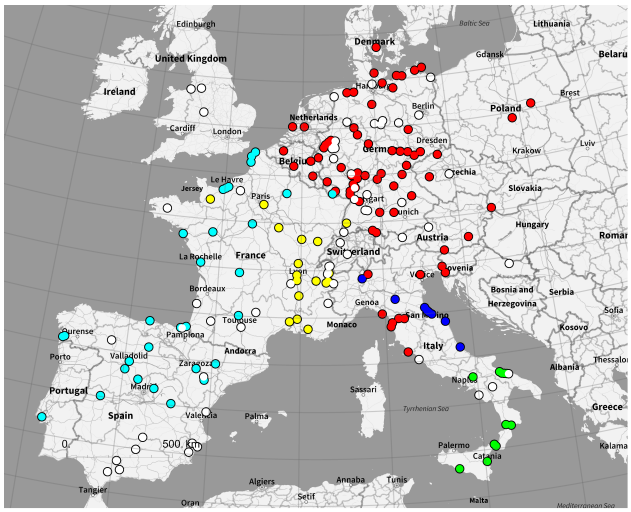
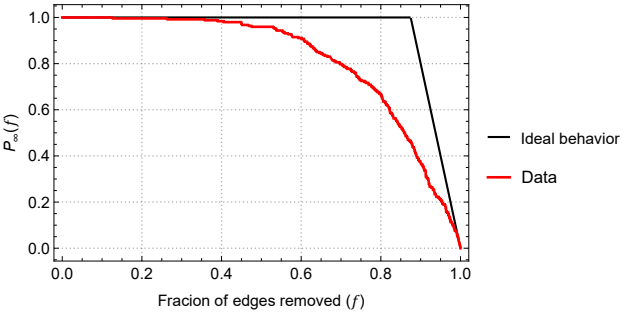
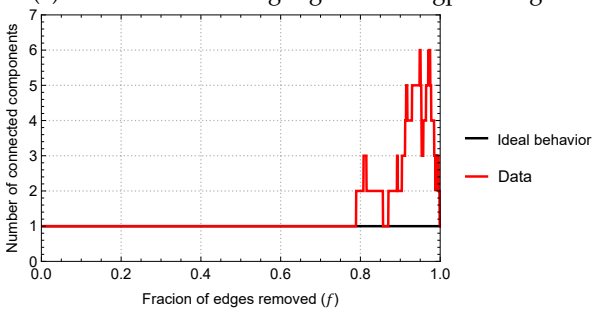


Figure 11: Survivor communities after the removal of the top 50 highest-degree nodes. We can see that the community detection algorithm still manages to pick up some communities, meaning that some areas of the network manage to stay locally connected. The white dots group all the nodes that do not belong to any of the biggest communities.

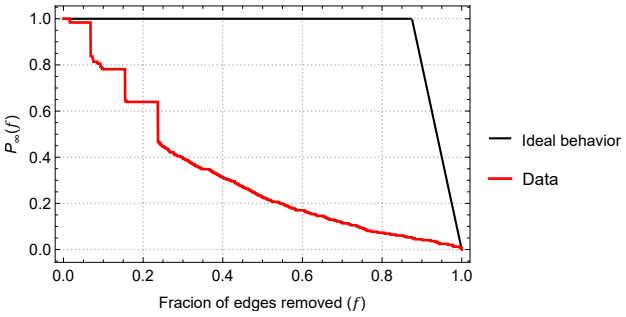


(a) Results of removing highest-througphut edges.

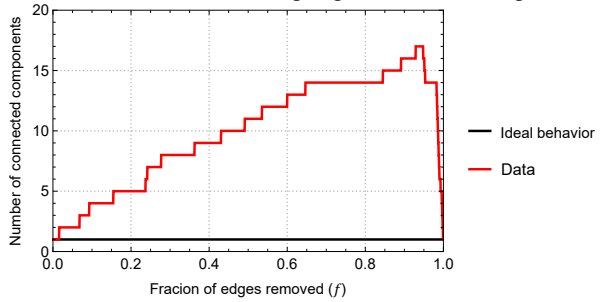


(b) Number of connected components after the highest-throughput edges are removed. The value of 1 indicates that the whole network stays connected, up until the top 80% of the edges are removed.

Figure 12: Simulation of resilience of the network to targeted edge attacks. One simulation was performed in which the highest-throughput edges were removed sequentially. The graph (a) shows the fraction of nodes that are still part of the giant component after the removal, while the graph (b) shows the number of components in which the network is split after the removal.



(a) Results of removing highest-salience edges.



(b) Number of connected components after the highest-salience edges are removed. The fact that this value is growing indicates that the network is being fragmented in small (larger than a single node) pieces.

Figure 13: Simulation of resilience of the network to targeted edge attacks. One simulation was performed in which the highest-salience edges were removed sequentially. The graph (a) shows the fraction of nodes that are still part of the giant component after the removal, while the graph (b) shows the number of components in which the network is split after the removal.

6.2.3 *Attacking edges*

We can also simulate how the network would behave if an attack were to be performed on the links.s The rationale behind this is that train tracks, contrary to train stations, cannot be monitored throughout their whole length and thus could be more easily targeted by a destructive attack[8]. There is some research [26] on what is the best edge attack strategy, but it is mostly based on empirical results and lacks a deep theoretical study.

I run two simulations with two different attack strategies. In the first one, I removed the links according to their throughput (magnitude in the rate matrix W), starting from the highest to the lowest. In the second, I removed the links according to their *edge salience* [27] value (which is a measure related to the number of shortest paths that pass through a given edge).

The two strategies lead to very different results. When using the former strategy (Figure 12), the network remains mostly untouched. As a matter of fact, if we confront the two graphs in Figure 12a and Figure 12b we can see that, even though P_∞ starts to decay after $f = .4$, the number of connected components of the network stays constant throughout, up until $f = .8$. This means that the removal of edges would, at most, only leave out singular nodes without fragmenting the network. When using the latter strategy, however, we find a different behavior (Figure 13). Here, the attack almost immediately fragments the network. Each sharp drop the graph of Figure 13a mean that a subgraph (whose size is given by the magnitude of the drop) has been isolated from the rest of the network.

As we did for the nodes, we can plot the correlations between different edge centrality measures (Figure 18). One interesting result is the positive correlation between edge salience and its length. This makes the result of the simulation just above even more alarming, as we can assume that attacking a longer (and thus more salient) train track would be easier than a shorter one.

6.3 Summary of the results

Equations (3) and (4) can be plotted together to show the predicted breakdown threshold of a scale-free network under the two possible moedes of failure. The result (Figure 14) shows that for low values of γ, scale-free networks are susceptible to targeted attack and resilient to random failures. If the value of γ gets too high, the value of f_c tends to converge for both cases.

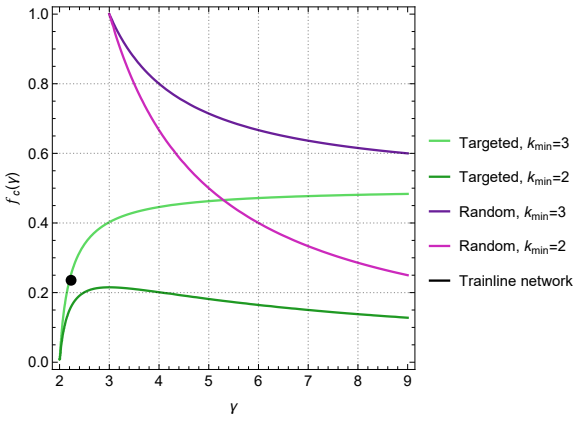


Figure 14: Plot of equations (3) and (4), which summarizes the resilience of a network as a function of the parameters γ and $k_{\min}$. We can see that for low values of γ a scale-free network is resilient to random failures but vulnerable to attacks. Over a certain threshold, the values fo f_c tend to converge for both types of failures.

7 Conclusions and Future Works

7.1 Conclusions

In this paper, we have shown how the Trainline network is scale-free. This fact alone dictates most of its robustness properties; mainly, we had predicted:

- Very high tolerance to random node failures, and
- Low tolerance to targeted attacks.

The theoretical results matched perfectly the numerical simulations on the real-world data. We can summarise them with the following points.

8. Again, I would like to stress out that the Trainline network is very different from the European train track network. Nonetheless, in order for a link to exist on the Trainline network, there need to exist a physical track link as well.

- The Trainline network is scale-free with degree exponent $\gamma = 2.23$. It exhibits ultra-small world properties.
- If some number of randomly-chosen stations were to fail, the vast majority of all the other stations in the network would still be able to function normally.
- In the (unlikely) scenario in which the highest-degree stations were targeted by an attack, we would see the network fragment only after the top 20% of nodes is taken out. Some large communities would still be standing up until the top 30% of nodes is taken out.
- In the (more likely) scenario in which the highest-throughput links were targeted by an attack, the majority of the network would still be able to function normally, up until 80% of the links are removed. Before that, only a few isolated stations would be cut off from the network.
- However, if a clever attacker were to target the most salient links, the network would fragment immediately.

My results differ from those obtained in [12], but this might be due to the different nature of the networks.

7.2 Future works

This work highlighted some strengths and weaknesses of the Trainline network. While this network may be correlated with the physical train track network, in reality it is of a very different nature. The *failing* of a node (or edge) in this network does not imply that the respective station has been shut off: it can also mean that one specific train company has decided to disable one of its routes due to economical reasons. This robustness study is useful per se, but it might also be interesting to study the robustness properties of the *physical* train tracks network, in which each link corresponds to some stretch of railroad. This network would allow multi-links and its properties could be different than those observed in this paper. This kind of study could help predict and prevent possible interruptions in the case of terrorist attacks or wars, for example.

Another possible line of work could be an analysis of the discrepancies between the results from this paper and the ones from [12]. Even though the network appears scale-free in both studies, the results in [12] suggest that the actual passenger transport network may be less robust than what the result in the present work suggest (due to a higher value of γ). A thorough study would need to compare different strategies for aggregating train timetable data into a network in order to find out which is the one that is most representative of the real world. Such network would probably contain both time-dependent links and multi-links.

References

[1] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[2] P. Erdös and A. Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[3] E. N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[4] Reka Albert, Hawoong Jeong, and Albert-Lazlo Arabasi. Diameter of the world-wide web. 1999.

[5] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016.

[6] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, August 1999.

[7] H. Jeong, S.P. Mason, A.-L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411, 2001.

[8] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[9] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale-free topology of e-mail networks. *Phys. Rev. E*, 66(3):035103, September 2002.

[10] Reuven Cohen and Shlomo Havlin. Scale-free networks are ultrasmall. *Physical Review Letters*, 90(5), February 2003.

[11] Béla Bollobás* and Oliver Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, jan 2004.

[12] L. Calzada-Infante, B. Adenso-Díaz, and S. García Carbajal. Analysis of the european international railway network and passenger transfers. *Chaos, Solitons and Fractals*, 141:110357, 2020.

[13] *European rail timetable summer 2018*. European Rail Timetable, Peterborough, England, June 2018.

[14] G.C. Homans. *The Human Group*. International library of sociology and social reconstruction. Harcourt, Brace, 1950.

[15] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, August 2002.

[16] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[17] Wayne W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[18] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[19] Imre Derényi, Gergely Palla, and Tamás Vicsek. Clique percolation in random networks. *Phys. Rev. Lett.*, 94:160202, Apr 2005.

[20] Wolfram Research. FindGraphCommunities. https://reference.wolfram.com/language/ref/FindGraphCommunities.html, 2015.

[21] A. Bunde and S. Havlin. *Fractals and Disordered Systems*. Springer Berlin Heidelberg, 2012.

[22] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85(21):4626–4628, November 2000.

[23] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 1995.

[24] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

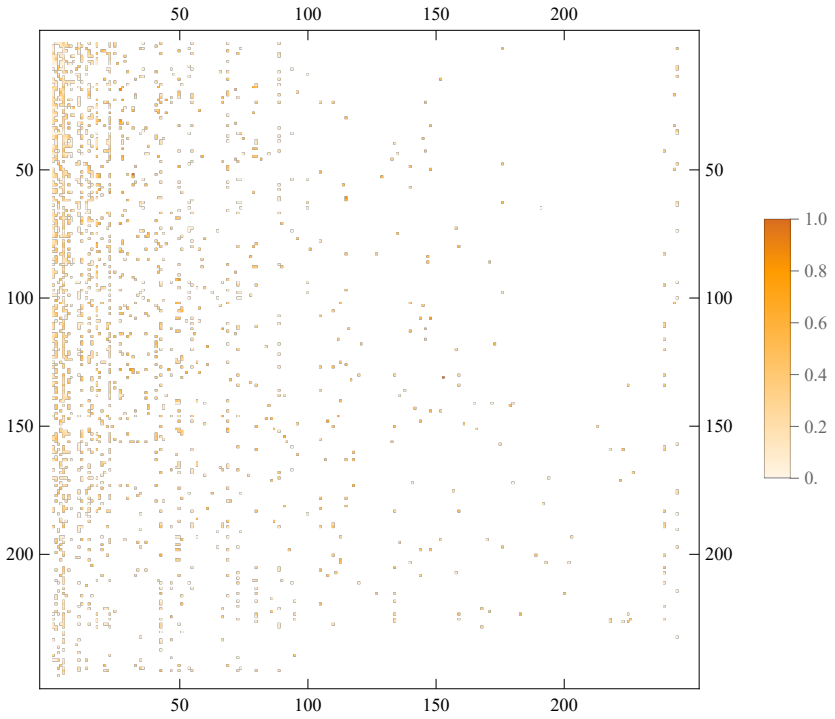
[25] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, 86:3682–3685, Apr 2001.

[26] M. Bellingeri, D. Bevacqua, F. Scotognella, R. Alfieri, and D. Cassi. A comparative analysis of link removal strategies in real complex weighted networks. *Scientific Reports*, 10(1), March 2020.

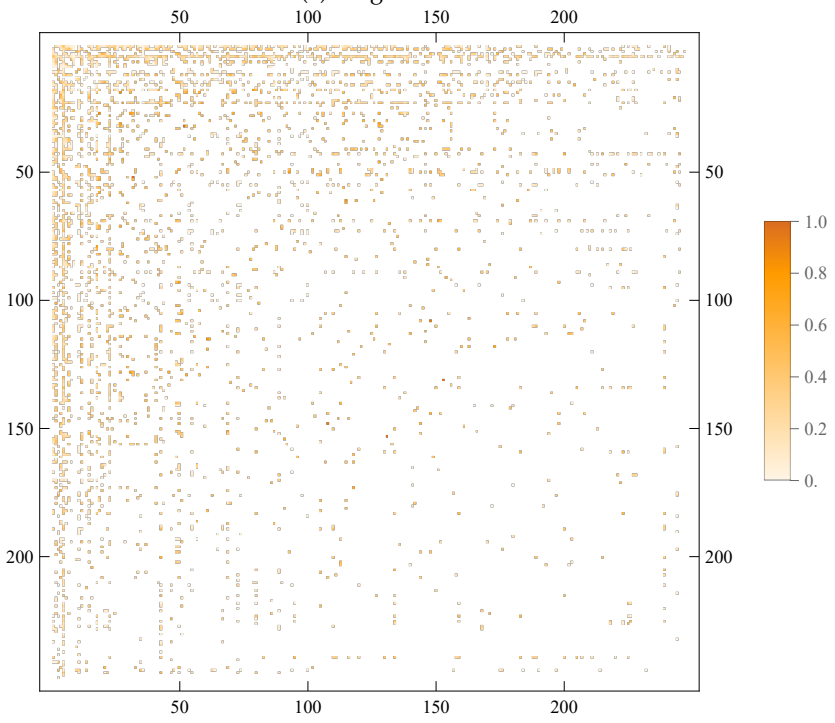
[27] Daniel Grady, Christian Thiemann, and Dirk Brockmann. Robust classification of salient links in complex networks. *Nature Communications*, 3(1), May 2012.

[28] Wolfram Research. Citydata source information-wolfram language documentation. https://reference.wolfram.com/language/note/CityDataSourceInformation.html.

Appendix A
Large pictures



(a) Original rate matrix



(b) Symmetrized rate matrix

Figure 15: Plot of the full, normalized rate matrix, sorted by city population.

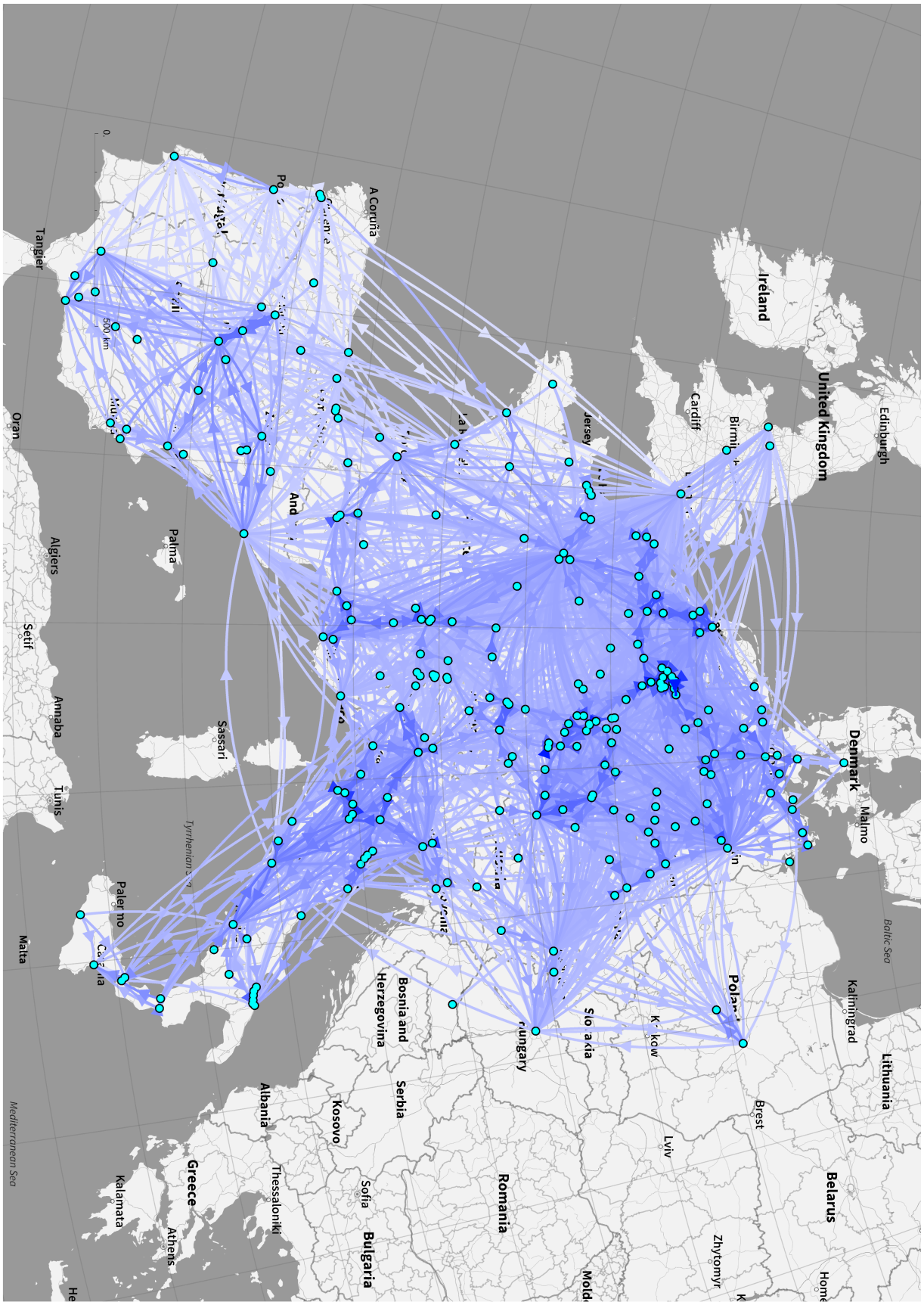


Figure 16: Big plot of the whole (unsymmetrized) network. The color of each arrow is proportional to its respective weight in the rate matrix W.

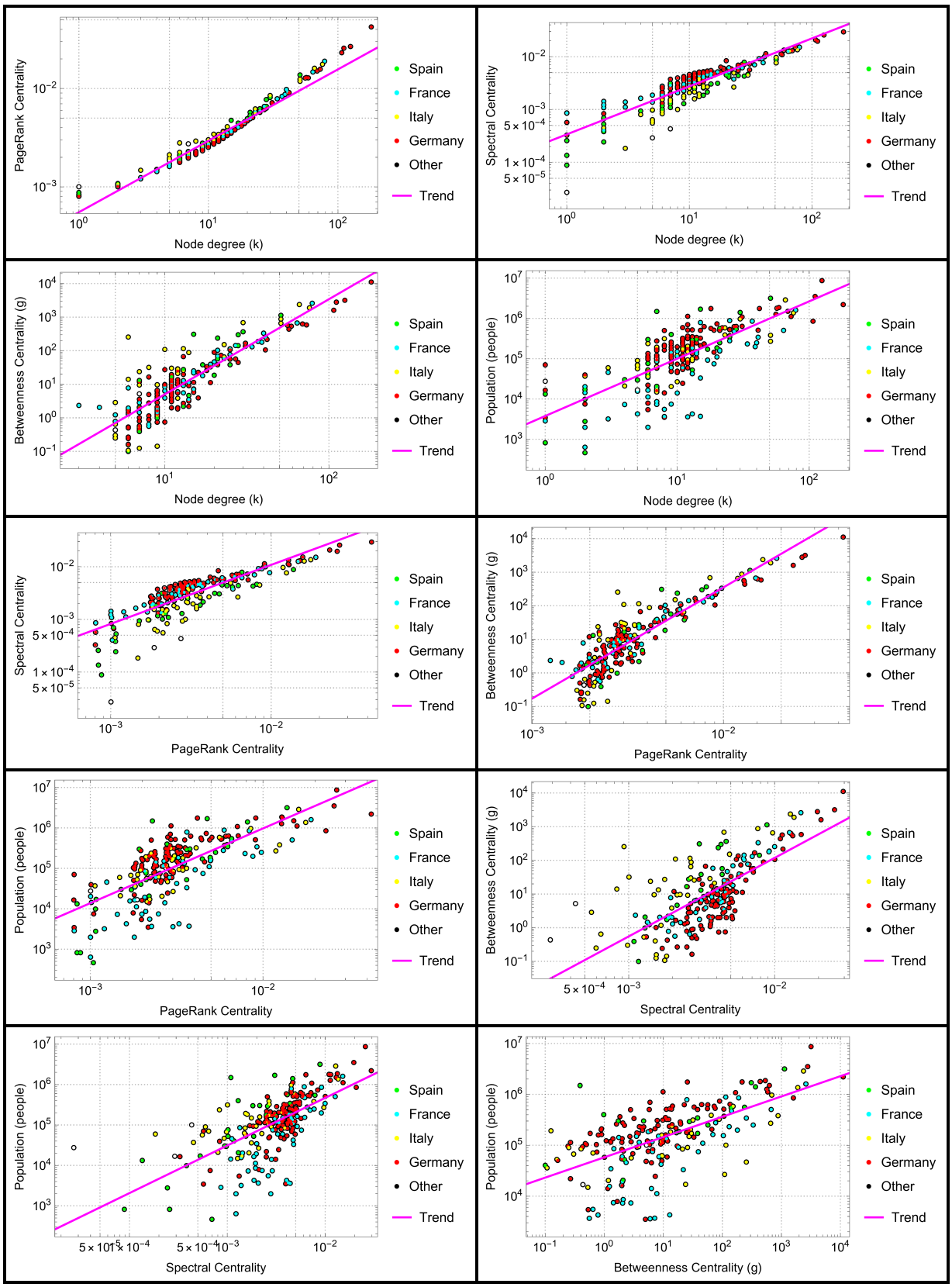


Figure 17: Scatter plots of different node centrality measures. The data points are colored according to the community subdivision; the colors appear to be uniformly distributed within the data points. The city of Pairs is erroneously classified as belonging to the community of Germany, and it is the rightmost point in every graph (except for population). The population data is taken from Wolfram's CityData repository [28], while the other values are computed from the Trainline network. All the plots show positive correlation between measures.

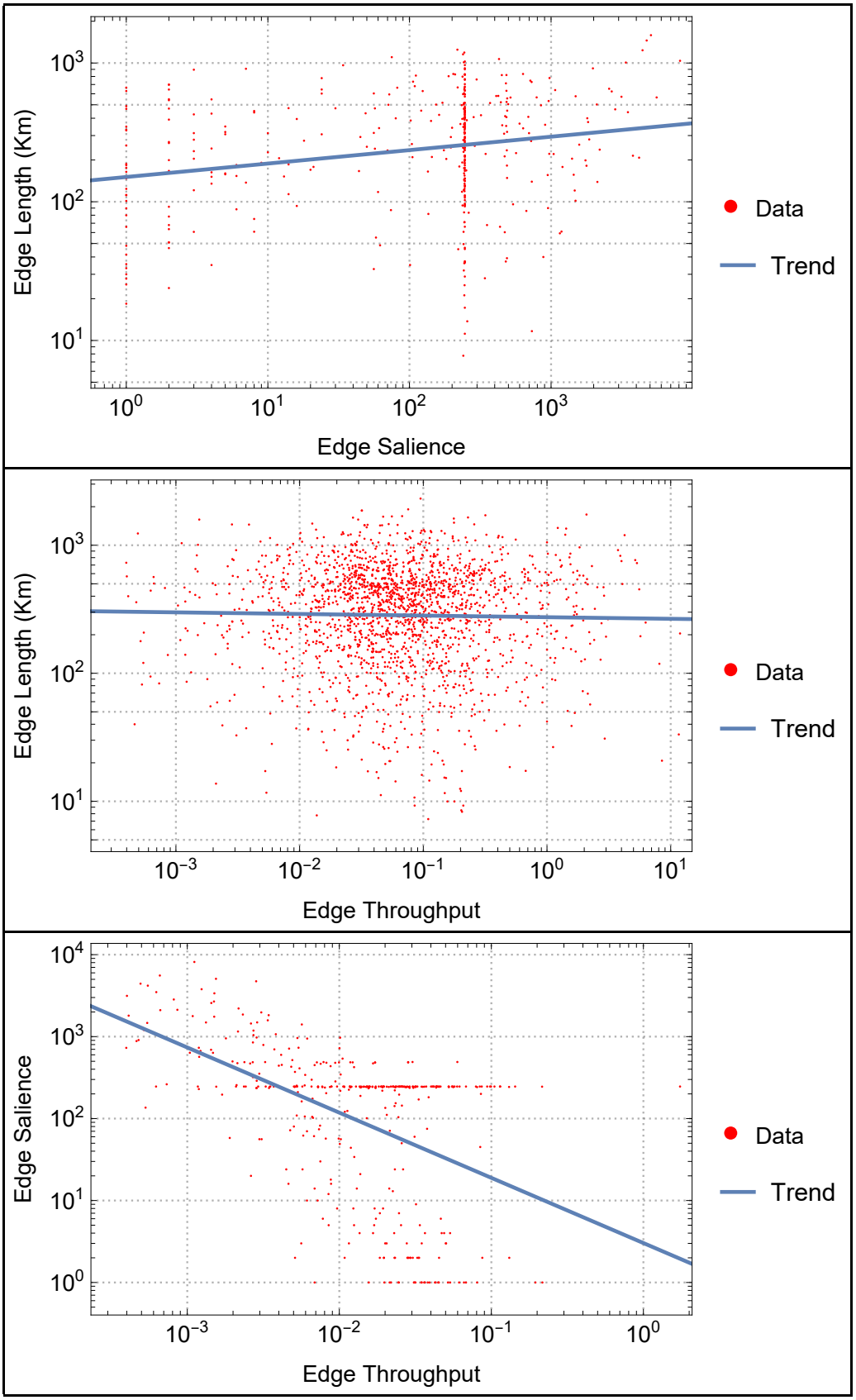


Figure 18: Scatter plots of different edge centrality measures. We can clearly see that there is positive correlation between the real-world edge length and its salience, negative correlation between edge salience and its throughput and no correlatiom between edge length and throughput.