Università degli Studi di Milano Bicocca

**Scuola di Scienze**

**Dipartimento di Informatica, Sistemistica e Comunicazione**

**Corso di laurea in Informatica**

# DATA ANALYTICS

Progetto d'esame

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

# Sommario

# Introduzione

Per questo progetto di Data Analytics affronterò il problema espresso nella traccia 7: L'analisi di una transportation network, e ispirandomi ai paper suggeriti nella traccia, cercherò di analizzare il grafo di una rete di trasporti pubblici di una città, al fine di valutarne la robustezza e la vulnerabilità a diversi tipi di attacchi.

Nel dettaglio, a partire dal dataset fornito, contenente i link di download a diversi GTFS (General Transit Feed Specification), ne sceglierò uno su cui concentrare l'analisi; quindi, valuterò la struttura e robustezza della rete; successivamente, cercherò di simulare diversi tipi di attacchi al grafo, e valutarne l'impatto sulla rete; e infine, costruirò una demo che permetta di visualizzare le precedenti analisi e simulare attacchi al grafo.

La relazione sarà divisa in quattro capitoli principali: nel primo farò una breve presentazione sui GTFS e sul quello scelto per l'analisi; nel secondo analizzerò la struttura del grafo e ne valuterò la robustezza usando diverse statistiche e metriche; nel terzo simulerò diversi tipi di attacchi al grafo e ne valuterò l'impatto; e infine, nel quarto, presenterò una demo che permetta di visualizzare le precedenti analisi e interagire dinamicamente con il grafo, al fine di studiare manualmente l'impatto degli attacchi.

# Capitolo 1: Breve presentazione sui GTFS

In questo capitolo presenterò brevemente il dataset fornito e i GTFS contenuti, di cui nello specifico: cosa sono, come sono strutturati, e quale ho scelto di analizzare.

## *1.1 Dataset fornito e GTFS*

Il dataset fornito contiene quasi 2000 GTFS. Per ognuno abbiamo il link di download e alcune informazioni aggiuntive: come l'agenzia pubblicante, lo stato, la regione, la città ecc..

Un GTFS (General Transit Feed Specification) è un formato di file standardizzato (in formato .zip) che contiene i dati di una rete di trasporti pubblici. È stato sviluppato da Google per rendere più facile la creazione di applicazioni di trasporto pubblico. Un file GTFS contiene diversi tipi di file: agency.txt, stops.txt, routes.txt, trips.txt, stop_times.txt, calendar.txt, calendar_dates.txt, shapes.txt. Ogni file contiene informazioni specifiche sulla rete di trasporti pubblici, come ad esempio: le agenzie coinvolte, le fermate, le linee, gli orari, le tariffe, le regole di trasferimento, ecc..

In particolare, per l'estrazione del grafo, ho usato i file:

- Stops: che include le informazioni riguardo alle stazioni, per l'estrazione dei nodi;
- Trips e stops_times: che includono informazioni riguardo a viaggi, linee ed orari, usati per l'estrazione degli archi del grafo.

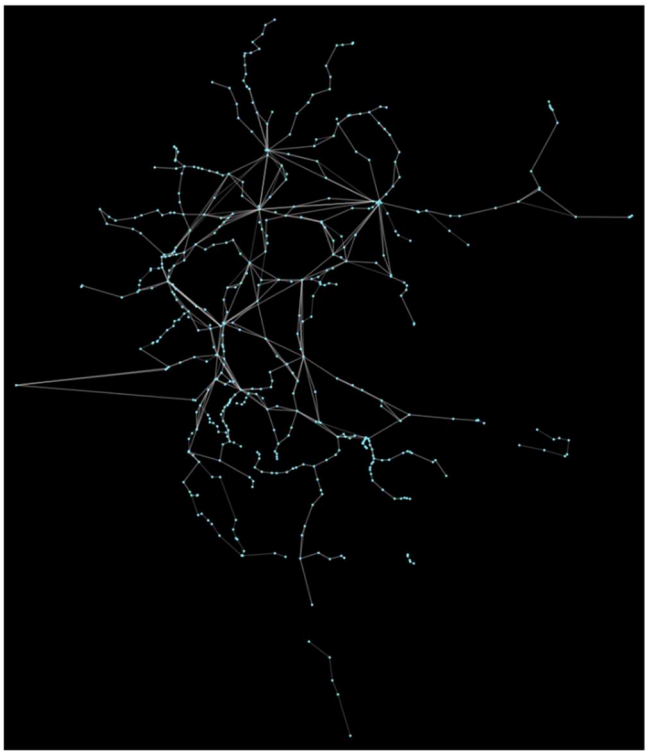
## *1.2 Scelta del GTFS*

Dopo una breve analisi iniziale su GTFS italiani ed europei, ho scelto il GTFS di Deutsche Bahn (BN), la rete ferroviaria tedesca. La ragione principale per cui l'ho scelta è la sua estensione e complessità (475 stazioni e circa 1200 linee); infatti, si estende su tutto il territorio tedesco e su alcuni stati limitrofi, e comprende sia linee locali e regionali, sia linee a lunga distanza.

Questo rende l'analisi sulla robustezza e vulnerabilità più interessante, rispetto a una rete locale (come la metropolitana di una città); in quanto eventuali fallimenti o attacchi potrebbero avere un impatto su un numero maggiore di persone, su un territorio più vasto

e hanno una maggiore probabilità di causare disagi e problemi (perché difficilmente sostituibili di altre linee locali).

*Figura 1: Grafo sintetico estratto dal GTFS*



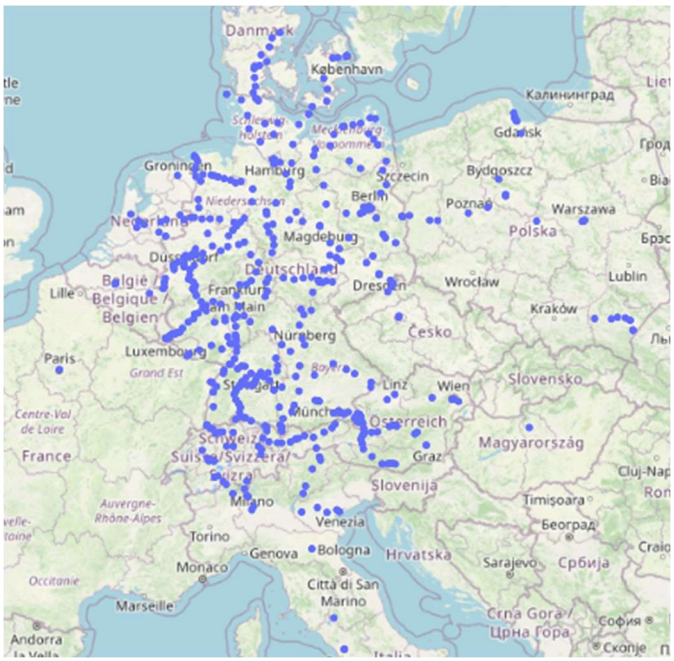



*Figura 2: Mappa semplici delle stazioni*

# Capitolo 2: Analisi strutturale del grafo

In questo capitolo analizzerò la struttura del grafo. In particolare, prima esporrò le metriche adottate (come il grado, la centralità, la connettività, ecc..), poi le userò per estrarre le caratteristiche della rete; al fine, di stabilire l'importanza di ogni nodo e valutare robustezza e vulnerabilità della rete.

## *2.1 Metriche adottate*

Le metriche adottate per l'analisi del grafo sono state scelte tra quelle studiate e in base ai paper suggeriti nella traccia, e sono le seguenti:

- Degree: Il numero di archi incidenti su un nodo.
- Centrality Degree (CD): Calcolato come il grado normalizzato di un nodo rispetto al grado massimo; misura la capacità di influenza di un nodo sulla rete.
- Centrality Betweenness (CB): Calcolato come il numero di volte che un nodo è attraversato dai cammini minimi tra due nodi; misura la capacità di un nodo di connettere due parti della rete.

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

- Centrality Closeness (CC): Calcolato come l'inverso della somma delle distanze minime tra un nodo e tutti gli altri nodi; misura la capacità di un nodo di raggiungere velocemente tutti gli altri nodi.

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

- Centrality Eigenvector (CE): Misura e attribuisce importanza ai nodi in base all'importanza dei loro vicini; calcolato tramite l'auto vettore della matrice di adiacenza.

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

- Centrality Clustering Coefficient (CCC): Calcolato come il rapporto tra il numero di archi tra i vicini di un nodo e il numero massimo di archi possibili; misura il grado di interconnessione tra i vicini di un nodo.

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

- Centrality PageRank (CP): Calcolato usando l'algoritmo di PageRank su ogni nodo; misura l'importanza di un nodo sulla base dell'importanza dei nodi che si connettono ad esso.
- Density: Calcolato come il rapporto tra il numero di archi e il numero massimo di archi possibili; misura il grado di connettività della rete.
- Giant Component: Calcolato come la dimensione della componente connessa più grande; misura la capacità di connettività della rete, in variante Strong (prendendo solo archi diretti) o Weak.

Nota: per il calcolo delle centralità ho utilizzato i pesi degli archi (distanza tra le fermate) come pesi, in modo da tener conto della distanza reale tra le fermate e non solo della loro connettività.

## 2.2 Analisi del grado

La prima analisi che ho effettuato è stata quella del grado dei nodi. Il grado di un nodo è il numero di archi incidenti su di esso (sia diretti che non), e rappresenta il numero di connessioni che un nodo ha con gli altri nodi della rete. In generale, un nodo con un grado elevato è più importante e influente di un nodo con un grado basso, in quanto ha più connessioni e influenza più nodi.

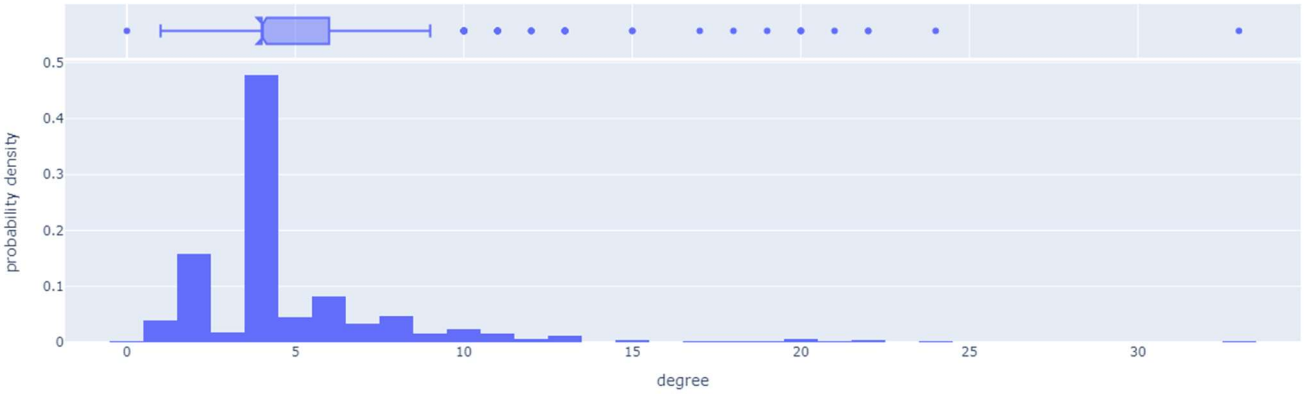



*Figura 3: Distribuzione del grado dei nodi*

Come si può notare dal grafico, la maggior parte dei nodi ha un grado basso tra 1 e 9, con il grado più comune uguale a 4; inoltre, ci sono pochi nodi con un grado elevato (>= 10) e ancora meno con un grado molto elevato (>= 20), inoltre il grado massimo è 33. Possiamo notare anche un caso particolare di un nodo con grado 0, che probabilmente rappresenta una fermata registrata nel GTFS ma non ancora attivata dall'agenzia.

Dunque, possiamo dire che la distribuzione rispecchia le aspettative, in quanto la maggior parte delle fermate sono locali e hanno poche connessioni, mentre poche fermate sono di interscambio e hanno molte connessioni.

## 2.3 Analisi della centralità

La centralità di un nodo è una misura della sua importanza all'interno della rete, e ne esistono diverse tipologie, ognuna con un significato diverso. In generale, un nodo con una centralità elevata è più importante e influente di un nodo con una centralità bassa, in quanto ha più connessioni e influenza più nodi. Per il suo calcolo ho utilizzato diverse metriche (come descritto nel paragrafo 2.1) in quanto forniscono interpretazioni diverse e complementari della centralità di un nodo.

### 2.3.1 Centrality Degree

Misura la capacità di influenza di un nodo sulla rete, ed è calcolato come il grado normalizzato di un nodo rispetto al grado massimo; dunque, i risultati sono pressoché identici a quelli del grado, con la differenza che i valori sono compresi tra 0 e 1.

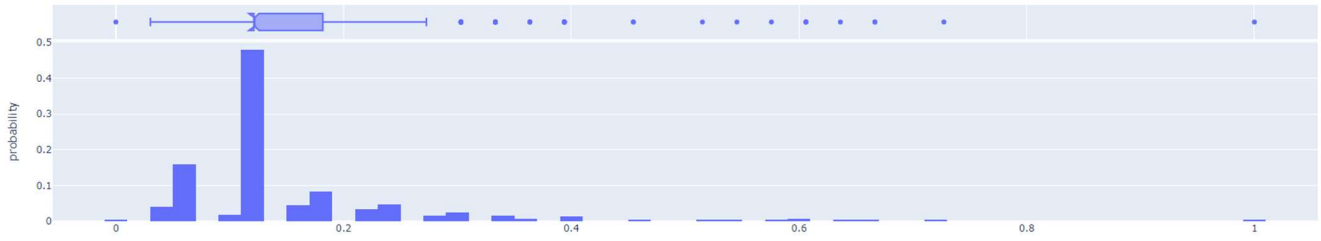



*Figura 4: Centrality degree distribution*

### 2.3.2 Centrality Betweenness

Misura la capacità di un nodo di connettere due parti della rete, ed è calcolato come il numero di volte che un nodo è attraversato dai cammini minimi tra due nodi.

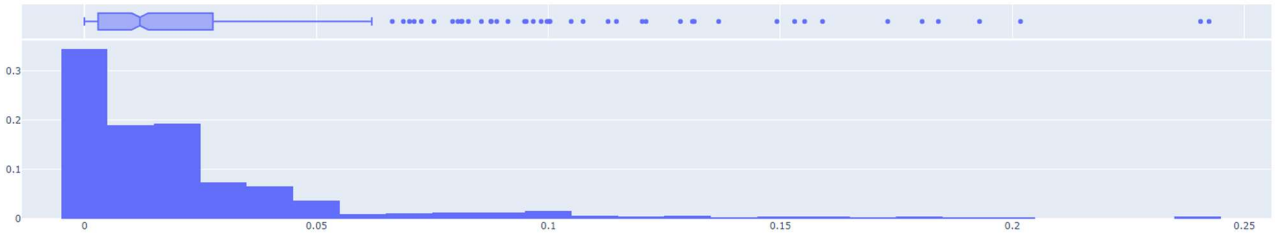



*Figura 5: Centrality Betweenness distribution*

I risultati mostrano che la maggior parte dei nodi ha una centralità bassa, con valori compresi tra 0 e 0.05, mentre alcuni nodi hanno una centralità molto più elevata. Questo ci permette di distinguere tra due tipi di nodi: gli hub, che hanno una centralità elevata e connettono diverse parti della rete, e i nodi periferici, che hanno una centralità bassa e non connettono parti diverse della rete.

### 2.3.3 Centrality Closeness

Misura la capacità di un nodo di raggiungere velocemente tutti gli altri nodi, ed è calcolato come l'inverso della somma delle distanze minime tra un nodo e tutti gli altri nodi.

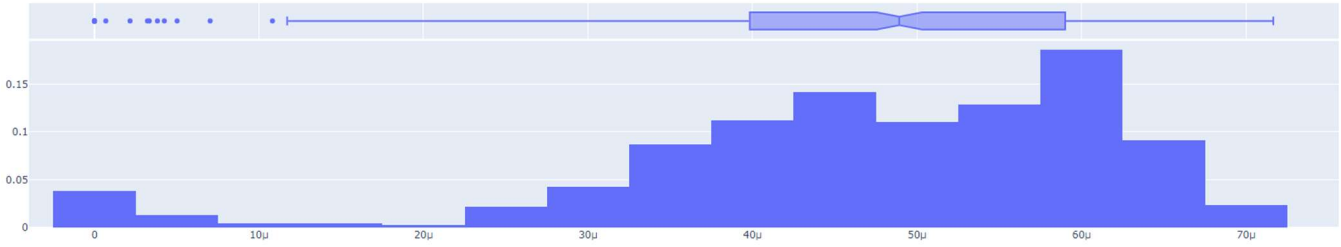



*Figura 6: Centrality Closeness distribution*

Come si può notare dal grafico, in generale la centralità di Closeness è molto bassa, con valori compresi tra 2,5e-5 e 7e-5. Questo seppur non ci permette di distinguere tra hub e nodi periferici, è caratteristico di una rete ad alta estensione come quella in analisi, in quanto la distanza tra le fermate è molto elevata.

### 2.3.4 Centrality Eigenvector

Misura e attribuisce importanza ai nodi in base all'importanza dei loro vicini, ed è calcolato tramite l'autovettore della matrice di adiacenza; è molto utile, in quanto solitamente, fornisce una visione leggermente diversa rispetto alle altre metriche.

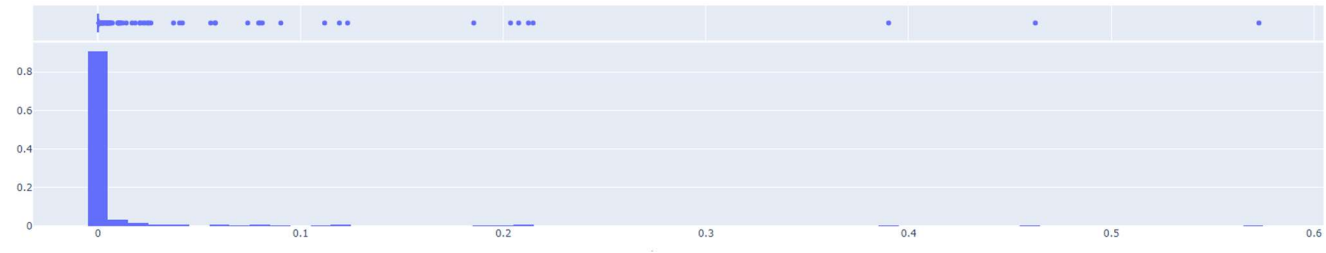



*Figura 7: Centrality Eigenvector distribution*

La Centrality con Eigenvector è molto simile a quella di Betweenness, seppur molto più skewed verso valori bassi. Questo ci permette di confermare che la rete è composta principalmente da nodi periferici, con pochi hub.

### 2.3.5 Centrality Clustering Coefficient

Misura il grado di interconnessione tra i vicini di un nodo, ed è calcolato come il rapporto tra il numero di archi tra i vicini di un nodo e il numero massimo di archi possibili.

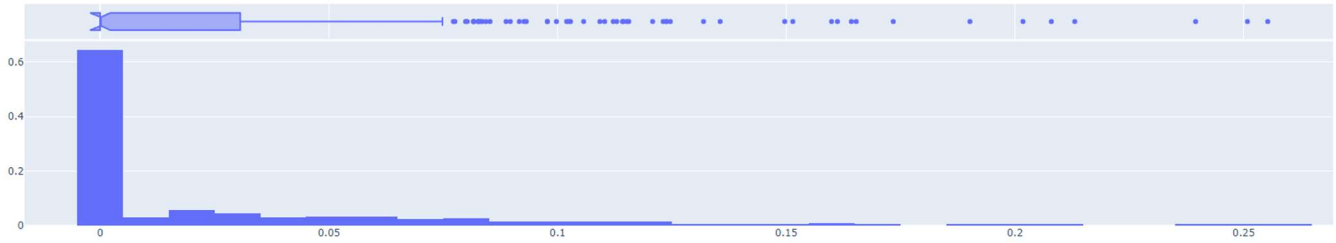



*Figura 8: Centrality Clustering Coefficient distribution*

Questo grafico ci permette di confermare che la rete è composta principalmente da nodi periferici (proprietà tipica di una rete ad alta estensione), in quanto la maggior parte dei nodi ha un coefficiente di clustering basso (il 60% ha un valore di circa 0.005), tuttavia, ci sono alcuni nodi con un coefficiente di clustering molto più elevato, che potrebbero essere hub.

### *2.3.6 Centrality Page Rank*

Misura l'importanza di un nodo in base all'importanza dei vicini i cui archi sono diretti verso il nodo, ed è calcolato tramite l'algoritmo di PageRank.

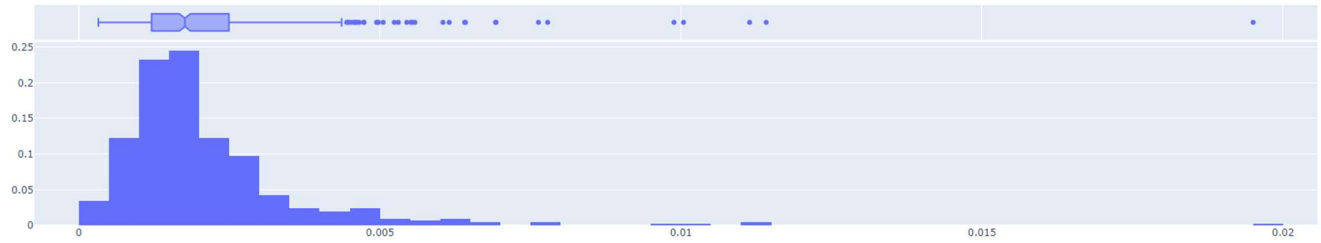



*Figura 9: Centrality PageRank distribution*

Come si può notare dal grafico, la centralità di PageRank presenta valori mediamente bassi (tra lo 0.001 e lo 0.0025), con pochi nodi più centrali (fino a 0.005) e alcuni outliers che raggiungono al massimo 0.02. Tuttavia, a differenza delle altre metriche, la centralità di PageRank presenta una distribuzione più uniforme.

### *2.3.7 Centrality complessiva*

Dopo aver considerato ogni metrica a parte, ho effettuato un'analisi complessiva della centralità, andando prima a calcolare la correlazione tra le metriche, in modo da capire quanto le informazioni fornite siano correlate tra loro, e poi andando a calcolare un indice complessivo di centralità, in modo da avere una visione complessiva della centralità di un nodo. Ciò è stato calcolato sulle metriche relative, ovvero scalate con il loro massimo, in

modo da avere valori su una scala uniforme 0 e 1 (utile non per la valutazione assoluta della centralità, ma per la valutazione relativa tra i nodi).

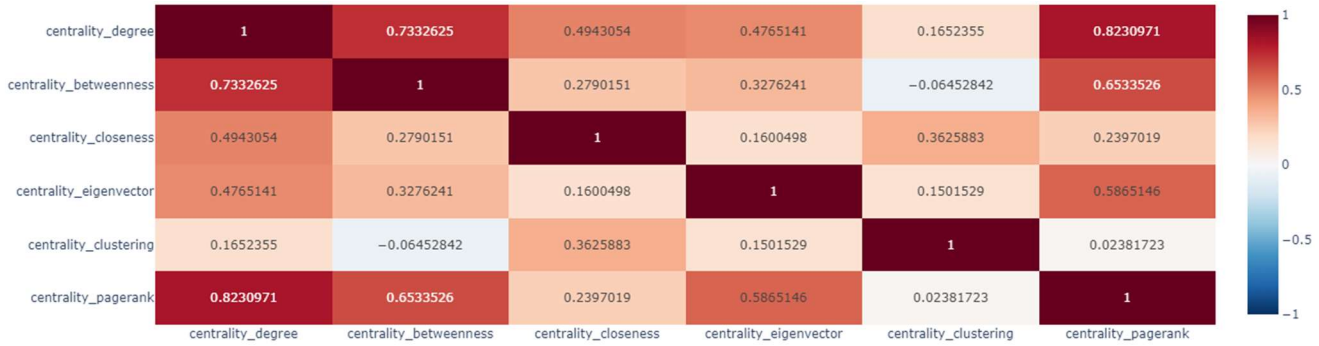



*Figura 10: Correlazione fra le centralità*

Analizzando la correlazione, ho notato che in generale esse sono tutte correlate tra loro (come atteso), ma ci sono alcune considerazioni che si possono fare:

- Degree, Betweenness e PageRank sono le metriche più correlate tra loro, in quanto tutte dipendono dal numero di archi incidenti su un nodo.
- PageRank ed Eigenvector sono molto correlate tra loro, in quanto entrambe dipendono dall'importanza dei vicini di un nodo.
- Clustering Coefficient è la metrica meno correlata con le altre (addirittura negativamente con Betweenness), in quanto dipende dal grado di interconnessione tra i vicini di un nodo.

In particolare, possiamo riassumere le diverse metriche (e le loro correlazioni) con la seguente tabella:

| 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 |

*Figura 11: Tabella riassuntiva delle metriche (assolute)*

Da cui possiamo confermare le deduzioni fatte in precedenza, ovvero che: la rete è composta principalmente da nodi periferici, e pochi hub; le centralità di Degree e PageRank sono quelle più uniformi, mentre le altre sono più skewed verso valori bassi; e che la centralità di Clustering Coefficient è la meno correlata con le altre metriche.

Infine, ho calcolato l'indice di centralità relativa complessivo, come la media delle centralità relative pesata dall'inverso della correlazione con le altre metriche, in modo da dare più importanza alle metriche meno correlate con le altre (che perciò hanno un maggior contenuto informativo).

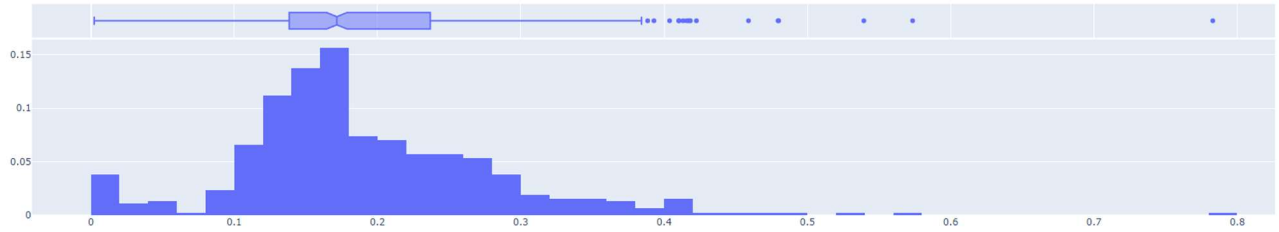



*Figura 12: Centrality index*

## 2.4 Analisi complessiva della rete

Dopo aver analizzato la centralità di ogni nodo, è utile effettuare un'analisi complessiva della rete, mediante metriche più generali. In particolare, oltre a considerare la media delle metriche di centralità e del grado, ho calcolato la densità della rete, e la grandezza della componente connessa più grande (nelle varianti Weak e Strong).

| 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 |

*Figura 13: Tabella complessiva della rete*

Da cui possiamo notare che la rete ha un grado medio di 5, una grandezza della componente connessa più grande molto elevata (98% Weak e 92% Strong), una centralità media di 0.2 e densità bassa (0.005). Possiamo quindi confermare che abbiamo una rete molto distribuita, poco compatta, ma molto connessa, con pochi hub e molti nodi periferici. Possiamo già dedurre che la rete è molto vulnerabile ad attacchi mirati ai nodi più centrali, ma molto robusta a attacchi casuali.

*Figura 14: Mappa delle stazioni. La dimensione dei marker rappresenta il grado, mentre il colore rappresenta la centrality.*

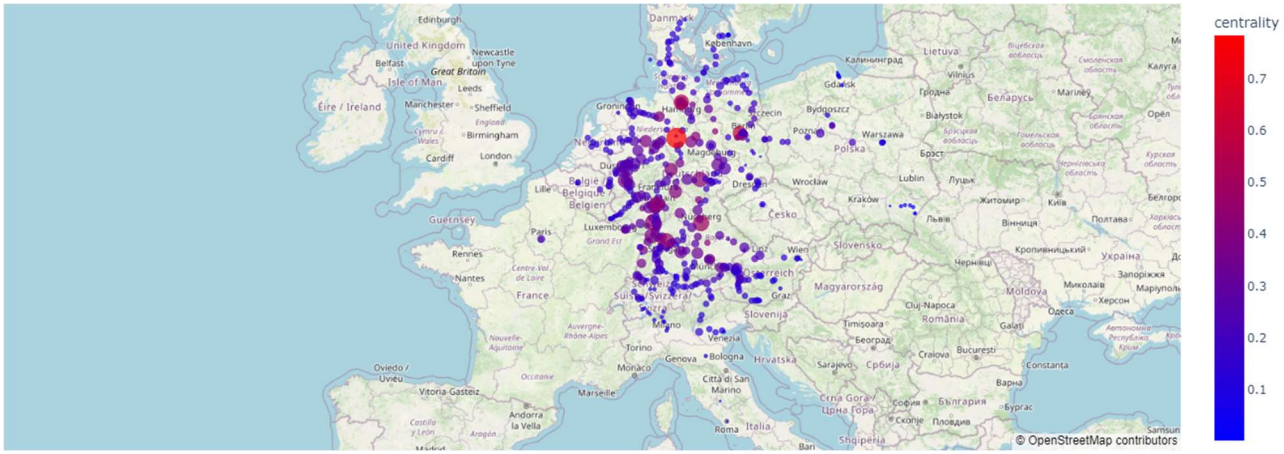




*Figura 15: Mappa delle stazioni e linee*

Come si può dedurre dalle due mappe, la maggior parte dei nodi sono collegati tra loro (ci sono solo alcuni segmenti separati in Polonia ed Italia), e la maggior parte degli Hub si trova in Germania (l'Hub principale è ad Hannover).

# Capitolo 3: Simulazioni di attacco alla rete

In questo capitolo simulerò diversi tipi di attacchi al grafo, e ne valuterò l'impatto sulla rete. In particolare, dopo aver descritto i vari tipi di attacchi, eseguirò le simulazioni e ne valuterò i risultati.

## *3.1 Definizione di attacco e criteri*

Si definisce un attacco come un processo iterativo in cui, ad ogni step, si rimuove uno o più nodi (e i relativi archi) dalla rete, e si valuta l'impatto di questa rimozione sulla rete, considerando diverse metriche, fino a raggiungere il 10% dei nodi rimanenti. Tale metriche saranno visualizzate separatamente, ognuna nel proprio grafico, e complessivamente, in un unico grafico in modo relativo, per valutare l'impatto dell'attacco sulla rete.

Le metriche considerate per valutare l'impatto dell'attacco sono le seguenti:

- Edges: Numero di archi rimanenti nella rete.
- Density: Densità della rete.
- Strong Giant Component: Grandezza della componente connessa più grande, considerando la direzione degli archi.
- Weak Giant Component: Grandezza della componente connessa più grande, non considerando la direzione degli archi.
- Centrality mean: Media delle centralità (relativa) dei nodi rimanenti.
- Avg Degree: Grado medio dei nodi rimanenti.

Inoltre, in ogni attacco i nodi da rimuovere saranno scelti in modo casuale (tipo di attacco Random); oppure, prendendo i nodi con la centralità più elevata (da cui abbiamo, per ogni centralità, un tipo di attacco diverso).

## 3.2 Risultati delle simulazioni

Ho simulato sia attacchi Random, sia attacchi mirati ai nodi con la centralità più elevata, per ogni metrica di centralità. I risultati delle simulazioni sono navigabili tramite la demo, qui riporterò ed analizzerò i più interessanti e infine quelli complessivi.

### 3.2.1 Attacco Random

*Figura 16: Simulazione attacco Random, grafico con metriche specifiche*

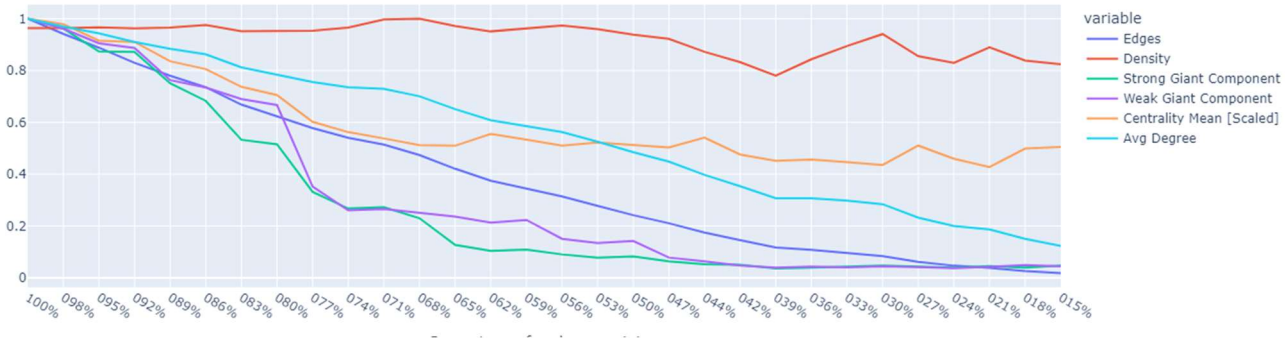




*Figura 17: Simulazione attacco Random, grafico riassuntivo*

Come previsto, l'attacco random non ha un impatto significativo sulla rete in quanto essa è molto connessa; infatti, anche rimuovendo il 10% dei nodi (circa 47) la rete rimane molto connessa (la grandezza della componente connessa più grande rimane all'85%), e la centralità media scende solo al 85% del valore iniziale (0.16); tuttavia, rimuovendo il 20% dei nodi la rete arriva al 50% di connettività, e la centralità media scende al 70% del valore iniziale (0.133); infine rimuovendo il 50% dei nodi, notiamo che la connettività scende sotto il 10%.

### 3.2.2 Attacco Centrality Degree

*Figura 18: Simulazione attacco Degree, grafico con metriche specifiche*

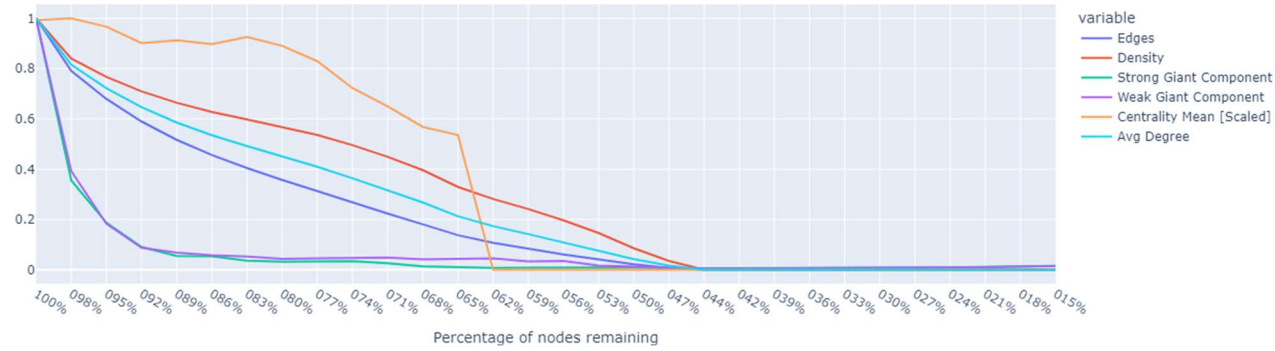




*Figura 19: Simulazione attacco Degree, grafico riassuntivo*

Analizzando l'attacco ai nodi con la centralità di degree più elevata, notiamo che l'attacco ha un impatto molto più significativo sulla rete. Infatti, rimuovendo solo il 2% (28 nodi) la rete arriva al 35% connettività, anche se la centralità rimane pressoché invariata; Inoltre, mentre le altre statistiche scendono in modo progressivo, possiamo notare che la centralità scende bruscamente tra il 30% e il 40% dei nodi rimossi, passando dal 65% a pressoché 0.

### 3.2.3 Attacco Centrality Clustering Coefficient

*Figura 20: Simulazione attacco Clustering Coefficient, grafico con metriche specifiche*

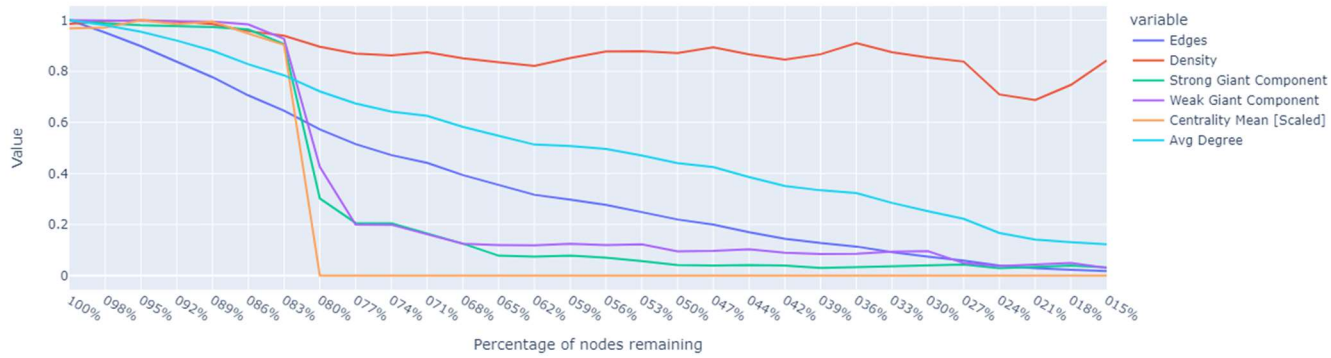



*Figura 21: Simulazione attacco Clustering Coefficient, grafico riassuntivo*

Come notiamo dai grafici, l'attacco di tipo Clustering Coefficient ha un impatto molto particolare (come atteso, vista la bassa correlazione della metrica rispetto alle altre). Notiamo infatti che fino al 20% dei nodi rimossi, sia la connettività che la centralità rimangono pressoché invariate (rispettivamente 92% e 90% del valore iniziale); invece, dopo aver rimosso altri 5% dei nodi (arrivando al 25% dei nodi rimossi), la connettività scende al 20%, mentre la centralità scende a 0.

### *3.2.4 Attacco Centrality Page Rank*

*Figura 22: Simulazione attacco Page Rank, grafico con metriche specifiche*

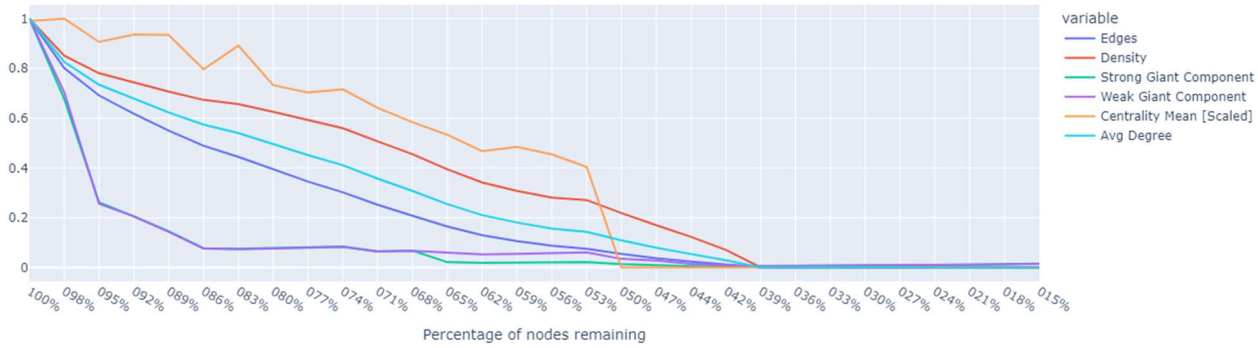



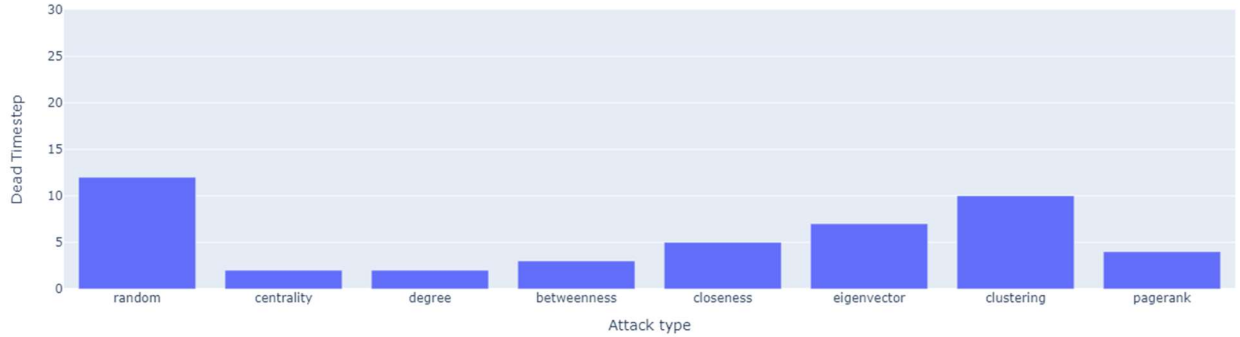

*Figura 23: Simulazione attacco Page Rank, grafico riassuntivo*

L'attacco di tipo PageRank ha un impatto molto simile a quello di tipo Degree, in quanto entrambe le metriche sono molto correlate tra loro. Infatti, notiamo che l'attacco ha un impatto molto significativo sulla rete (seppur minore), e che la connettività scende al 20% dopo aver rimosso il 5% dei nodi (70 nodi), mentre la centralità scende in modo progressivo fino al 50% dei nodi rimossi dove poi scende bruscamente a 0.

## 3.3 Analisi complessiva degli attacchi

Per valutare complessivamente l'impatto degli attacchi sulla rete, ho calcolato il 'Dead Timestep', ovvero il numero di step dopo cui la rete ha subito un impatto significativo (definito come una variazione impostata di una metrica rispetto al valore iniziale). Metrica e variazione sono impostabili nella demo, ma per questa relazione ho considerato connettività (Strong G.C.) e centralità media al 20% del valore iniziale.

*Figura 24: Dead timestep per ogni tipo di attacco, con Strong G.C al 20%*



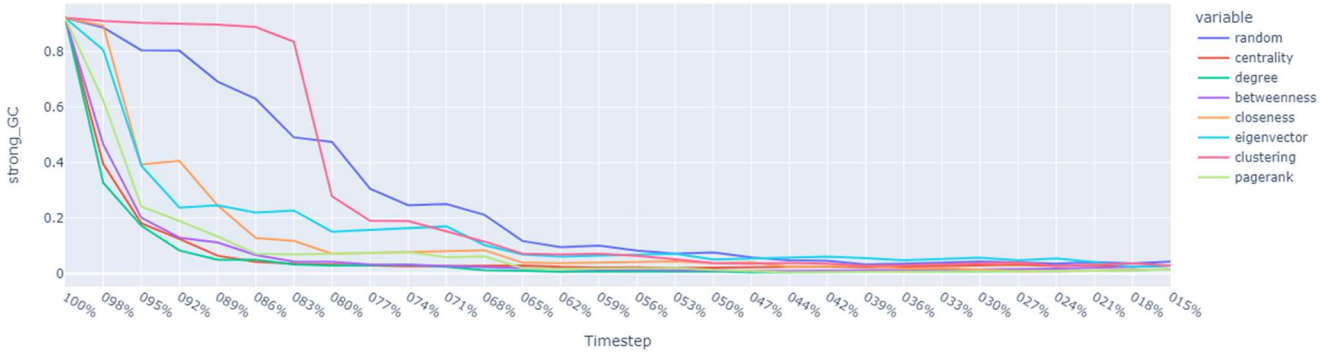




*Figura 25: Strong G.C. per ogni tipo di attacco ad ogni timestep*

Possiamo notare, da questo grafo, che ogni attacco con al più 12 step porta la rete allo stato terminale. In particolare, osserviamo che:

- Gli attacchi Centrality, Degree, Betweenness, Closeness e PageRank sono molto simili.
- Gli attacchi Eigenvector, Clustering e Random hanno un impatto molto minore sulla rete, con random (come atteso) che ha l'impatto minore.

*Figura 26: Dead timestep per ogni tipo di attacco, con Centrality al 20%*


*Figura 27: Centrality per ogni tipo di attacco ad ogni timestep*

Per la Centrality, osserviamo invece risultati alquanto diversi:

- Gli attacchi Centrality e Clustering ora hanno un impatto molto simile, con addirittura clustering che ha un impatto maggiore.
- Gli attacchi Degree, Betweenness, Closeness e PageRank hanno un impatto minore sulla rete rispetto a prima (ma comunque significativo).
- L'attacco Eigenvector ha l'impatto peggiore sulla rete (escludendo random), ma relativamente in linea con gli altri.
- L'attacco Random dopo tutti i timestep non ha ancora avuto un impatto significativo sulla rete.

Nota: Alcune statistiche di centralità (Degree, Closeness, Eigenvector) non sono ottimali per valutare l'impatto degli attacchi sulla rete; in quanto, sono calcolate e scalate rispetto ai nodi rimanenti, e quindi non tengono conto dell'effettivo impatto dell'attacco sulla rete. Tuttavia, sono state considerate per completezza e confronto con le altre metriche.

## *3.4 Conclusioni sugli attacchi*

Dalle simulazioni effettuate possiamo dedurre che:

*La rete è molto vulnerabile ad attacchi mirati ai nodi più centrali; al contrario, la rimozione randomica è molto meno efficace.*

- Le migliori statistiche per valutare l'impatto sono la Strong e Weak Giant Component, in quanto tengono conto della connettività della rete, e la Centralità Media, in quanto tiene conto dell'importanza dei nodi rimanenti.
- Gli attacchi con maggiore impatto sono quelli di tipo Degree, Betweenness, Closeness e PageRank, in quanto sono le metriche più correlate tra loro e con la connettività della rete.
- Durante gli attacchi la centralità tende a scendere in modo progressivo fino a un certo punto, dove poi scende bruscamente a 0.
- La rimozione di nodi secondo la strategia Clustering ha un impatto particolare, in quanto tende a danneggiare maggiormente la centralità rispetto alla connettività (cioè, è l'inverso delle altre).
- Le metriche Edges e Avg Degree scendono in modo lineare, seppur forniscono informazioni, sono meno significative per la valutazione dell'impatto degli attacchi sulla rete.
- La metrica Density nella maggioranza degli attacchi non è correlata con il numero di nodi rimossi, e quindi non è significativa (tranne nel caso di Degree e PageRank).

# Conclusione

In questo progetto, partendo da un dataset contenente quasi 2000 GTFS, ho brevemente analizzato i GTFS e scelto quello di Deuthesche Bahn (BN), la rete ferroviaria tedesca, per l'analisi; successivamente, ho analizzato la struttura del grafo e ne ho valutato la robustezza e vulnerabilità a diversi tipi di attacchi.

Il GTFS di Deuthesche Bahn (BN) dopo una fase di conversione in DataFrame e poi in grafo, è stato analizzato in termini di struttura e connessione. In particolare, ho usato e confrontato diverse metriche di centralità per valutare l'importanza dei nodi e la connettività della rete.

Successivamente ho simulato diversi tipi di attacchi al grafo, e ne ho valutato l'impatto sulla rete. In particolare, ho simulato sia attacchi Random, sia attacchi mirati ai nodi con la centralità più elevata, per ogni metrica di centralità. Ho poi confrontato i risultati delle simulazioni, valutando i danni causati alla rete.

In conclusione, ho dedotto che la rete è molto estesa e connessa, con pochi hub e molti nodi periferici, e che è molto vulnerabile ad attacchi mirati ai nodi più centrali (in particolare quelli di tipo Degree, Betweenness, Closeness e Pagerank), ma molto robusta ad attacchi casuali. Inoltre, ho osservato che le migliori statistiche per valutare l'impatto degli attacchi sono la Strong e Weak Giant Component, in quanto tengono conto della connettività della rete, e la Centralità Media, in quanto tiene conto dell'importanza dei nodi rimanenti.

Nel futuro, per un'analisi delle centralità più precisa, si potrebbero prendere in considerazione altri elementi degli archi (oltre alla lunghezza) come il tempo di percorrenza, la capienza, ecc.. Inoltre, si potrebbero considerare altri tipi di attacchi, come quelli basati sul numero di persone trasportate o su altri criteri di selezione. Infine si potrebbe estendere la demo per lavorare su altri GTFS (in quanto la struttura non è strettamente dipendente dal GTFS scelto), e per visualizzare altre metriche e simulare altri tipi di attacchi.

# Appendice A: Alcuni dettagli tecnici.

La demo, per poter funzionare, necessita di alcuni dati posti nel folder app/data; quindi, per eseguire il codice da zero, è necessario avviare prima il file install.bat (o installare le librerie nel requirements.txt), che istallerà le librerie; e successivamente, il file main.py (che si aspetta il sources.csv nel folder ./data) che si occuperà di creare i folder mancanti, scaricare i dati e preparare i file necessari per la demo.

Tra le varie librerie adottate, ne ho utilizzata una, di nome Peartree, per convertire il GTFS formato DataFrame in grafo; tuttavia, ho dovuto apportare alcune modifiche al codice per rimuover alcuni bug e per adattarlo alle mie esigenze.

Per avviare la demo è necessario avviare il file index.py, che si occuperà di avviare il server e la dashboard.

# Appendice B: Dashboard e Demo

La Dashboard è suddivisa in 4 pagine:

- Home: Pagina iniziale che presenta la mappa delle stazioni e le relative statistiche; si suggerisce di usare quella per visualizzare le sole statistiche dei nodi.
- Graph Evaluation: Pagina che espone l'analisi strutturale fatta al grafico.
- Attacks Analysis: Pagina che presenta le simulazioni di attacchi fatte al grafo, con la possibilità di cambiare le metriche e le variazioni per valutare l'impatto degli attacchi sulla rete.
- Demo: Pagina che presenta la demo, in cui è possibile visualizzare ed interagire con il grafo, e simulare attacchi manualmente.

In particolare, la demo, oltre a visualizzare il grafo completo, permette di:

- Visualizzare nodi e linee, con relative informazioni e statistiche.
- Esaminare e confrontare nel dettaglio i nodi, usando le due tabelle laterali (una principale e una di confronto) apribili con le relative frecce.
- Eliminare nodi e linee uno per uno, sfruttando la tabella di confronto per valutare l'impatto sui singoli elementi.
- Eliminare più elementi simultaneamente, al fine di simulare attacchi manualmente.
- Aumentare e ridurre la dimensione dei nodi con grado maggiore.
- Resettare o salvare le modifiche fatte al grafo (nel folder ./app/data/users).
- Usare la tabella degli stati (sotto la mappa) per visualizzare le statistiche del grafo dopo ogni rimozione.
- Usare la sezione "Risultati attacco manuale" (sotto la tabella) per visualizzare l'andamento nei vari stati delle statistiche del grafo.