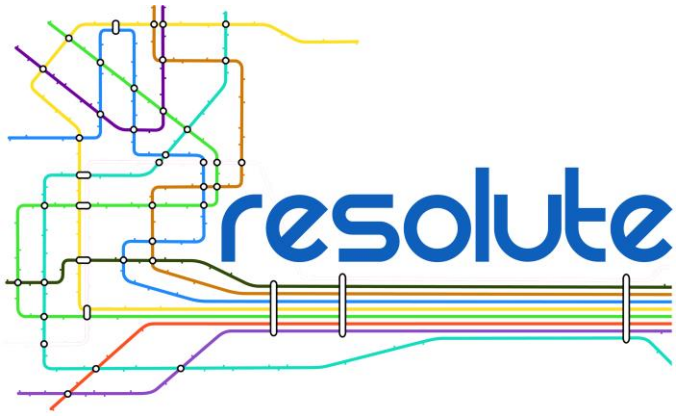


APPLICATION FRAMEWORK

Project Title	RESOLUTE
Project number	653460
Deliverable number	D4.3 – Application Framework
Version	4.1
State	Final
Confidentially Level	Public
WP contributing to the Deliverable	WP4 – Platform back-end
Contractual Date of Delivery	M1 (30/05/2015)
Finally approved by coordinator	14-04-2017
Actual Date of Delivery	28-February-2017
Authors	Francesco Archetti, Antonio Candelieri. Ilaria Giordani, Gaia Arosio
Email	archetti@milanoricerche.it, candelieri@milanoricerche.it, giordani@milanoricerche.it, arosio@milanoricerche.it
Affiliation	Consorzio Milano Ricerche (CMR)
Contributors	Anastasios Drosou (CERTH) Alexandros Zamichos (CERTH) Ilias Kalamaras (CERTH)

funded by the Horizon 2020
Framework Programme of the European Union

EXECUTIVE SUMMARY

This document provides a description of the Application Framework, the RESOLUTE component offering a set of analytical tools, based on user profiling and network science, to extract useful information from available sets of data in order to support a more sustainable resilience management of UTS.

The Application Framework addresses two specific issues that are the user profiling and the multi-risk and multi-layer network analysis of the UTS. While the former requires data relative to both RESOLUTE's and UTS's users, the latter is more focused on the modelling and analysis of the networked infrastructures which can be physical (i.e. road network) as well as service-related (i.e. public transport routes, not necessary mapped on the road network).

The contents of the deliverable are organized following a logic flow: links to the ERMG and the architecture of the RESOLUTE platform are initially presented; available sets of data are described, taking into account data sources "internal" to RESOLUTE as well as open-data initiatives and synthetic data; then user profiling and network analysis approaches for emergency management and resilience analysis, in particular in UTS, are summarized by carefully taking into account recent studies in the state-of-the art.

After that, a presentation of the tools used, integrated and expanded to implement data/network analysis functionalities are presented.

The document concludes with the report of results obtained from the design, development and preliminary validation of the implemented functionalities, while possible limitations and further improvements are summarized in the conclusion section.

PROJECT CONTEXT

Workpackage	WP4: Platform back-end
Task	T4.4: Application Framework
Dependencies	All the tasks in WP4, D2.2. and D3.5 and D3.7

Contributors and Reviewers

Contributors	Reviewers
Francesco Archetti (CMR)	
Antonio Candelieri (CMR)	
Ilaria Giordani (CMR)	
Gaia Arosio (CMR)	
Anastasios Drosou (CERTH)	
Alexandros Zamichos (CERTH)	
Ilias Kalamaras (CERTH)	

Version History

Version	Date	Authors	Sections Affected
0.0	29-07-2016	Antonio Candelieri (CMR)	Table of Contents
0.1	30-09-2016	Antonio Candelieri (CMR) Anastasios Drosou (CERTH), Alexandros Zamichos (CERTH), Ilias Kalamaras (CERTH)	Section 5
0.2	28-10-2016	Antonio Candelieri (CMR)	Section 4, Section 3
0.3	28-11-2016	Gaia Arosio (CMR)	Section 6
0.4	30-11-2016	Ilaria Giordani (CMR)	Section 2
1.0	19-12-2016	Ilaria Giordani (CMR), Anastasios Drosou (CERTH), Alexandros Zamichos (CERTH), Ilias Kalamaras (CERTH)	Section 3

1.1	16-01-2017	Ilaria Giordani (CMR)	Section 7
1.2	18-02-2017	Ilaria Giordani, Francesco Archetti (CMR)	All sections
2.0	28-02-2017	Ilaria Giordani, Francesco Archetti (CMR)	All sections (final release addressing QIR)
3.0	17-03-2017	Antonio Candelieri (CMR), Ilaria Giordani (CMR), Anastasios Drosou (CERTH), Alexandros Zamichos (CERTH), Ilias Kalamaras (CERTH)	Improved section 4 about user profiling (also results section)
3.1	24-03-2017	Antonio Candelieri (CMR), Ilaria Giordani (CMR), Anastasios Drosou (CERTH), Alexandros Zamichos (CERTH), Ilias Kalamaras (CERTH)	Improved section 5 about (a) "weather severity monitoring and associated flood hazard" and (b) cascading effects (also results section)
3.2	31-03-2017	Antonio Candelieri (CMR), Ilaria Giordani (CMR), Anastasios Drosou (CERTH), Alexandros Zamichos (CERTH), Ilias Kalamaras (CERTH)	Improved section 2 about addressed ERMG functions and technical details about computational modules
4.0	07-04-2017	Antonio Candelieri (CMR), Ilaria Giordani (CMR)	Improvement of conclusions section and finalization of the document

Copyright Statement – Restricted Content

Table of Content

List of Figures

1 INTRODUCTION

The Application Framework, which is part of the RESOLUTE platform back-end, consists of two computational modules enabling data analysis functionalities on the available data. These two computational modules mainly focus on: (i) managing user profiles – with respect both to the analysis of mobility habits of the UTS's users and the skill-based profiling of RESOLUTE's users to support team/skills management during emergency, and (ii) modelling and analysing, in a completely dynamic way, UTS networks through network analysis algorithms, and modelling and simulating cascading effects in UTS in order to predict the evolution of the impact on the network itself while computing resilience metrics with respect to the physical degradation and reduction of service.

The document is organized as follows: in chapter 2, ERMG functions are addressed and functionalities provided through the Application Framework are presented; in chapter 3 the available sets of data are also described: these can be divided in (i) data already available for the design, development and experimentation of the Application Framework functionalities, (ii) other open-data sources which can be further used to tune the Application Framework's algorithms and tools and (iii) other relevant data which could be acquired, integrated and used after RESOLUTE. Then, chapters 4 and 5 present functionalities for user's profiling and network analysis, respectively. In chapter 6 a detailed description of the design and development of the Application Framework is provided, including open-source software and libraries which have been extended and integrated in order to effectively implement the algorithms and models. Chapter 7 is devoted to the presentation of results obtained on the available datasets during the development of the Application Framework, while chapter 8 presents the most relevant conclusions about potentialities and limitations.

1.1 Relation with the project

The outcomes of the Task 4.4 "Application Framework" are strictly connected with the backend implementation and integration (WP4 Task 4.1, Task 4.3 and Task 4.5) and with the available data identified in Task 4.2, as well as with the pilots' definition and execution (WP6),

Design and development of algorithms and models for user profiling and network analysis have been driven by the conceptual model defined in D2.2 and the ERMG (D3.5 and D3.7). A relevant contribution has been provided by the activities of the T4.2 since data availability is one of the most critical aspects to be addressed in order to apply analytical functionalities to support decision making in UTS resilience management.

2 ADDRESSING THE ERMG

A cross-disciplinary and holistic approach is desirable in order to define practical strategies, guidelines and tools to strengthen the resilience of Urban Transport Systems (UTS). This chapter describes how algorithms and models for the analysis of relevant data can be used to implement specific functionalities addressing the ERMG.

2.1 Mapping ERMG functions to Application Framework's functionalities and services

According to *"Continuity of passenger mobility following disruption of the transport system"*[1] that is a working document of the European commission, the response to a disruption should not only aim to restore the material elements of the transport system (infrastructure, equipment and facilities and information systems) but should also focus on citizens and passengers who are most directly affected by the disruption.

Appropriate strategies to prevent and respond to unexpected disruption affecting cross-border passenger transport could therefore involve actions at several levels:

- **Prevention**: competent authorities and operators need to strengthen their risk assessments and consider the need to improve the resistance and resilience of their transport infrastructure and information management networks;
- **Preparedness**: competent authorities and operators need to take the appropriate emergency planning measures (procedural provisions for crisis handling, availability of rerouting scenarios, emergency exercises, etc.) necessary to ensure a rapid response to major transport disruption. Exchange of information and coordination of preparatory measures increase the effectiveness of crisis prevention and the reaction to disruptive events;
- **Information**: carriers and terminal managers, as far as their competence allows, should inform passengers of developments in the transport situation, alternative routes, changed schedules, etc. If passengers are provided with comprehensive information, they can play an active role in the crisis response, for example by choosing alternative routes or postponing a planned trip, thus mitigating the negative effects of the transport disruption;
- **Assistance**: carriers should provide appropriate care for stranded passengers (food, refreshment and accommodation if necessary);
- **Alternative transport arrangements**: carriers should reroute affected passengers to their final destinations in the shortest time and under the most convenient conditions possible (or return them to their point of departure if this is more convenient for the passengers);
- **Re-establishing the flow of passenger traffic**: competent authorities and operators need to cooperate with each other to re-establish the functioning of the transport system following a disruption;
- **Respect of passenger rights**: there should be an effective complaint-handling mechanism to ensure that passenger rights are respected also in the event of major disruptions (e.g. clear procedures and deadlines, and alternative dispute resolution options to help passengers enforce their rights).

[1] SWD(2014)155 - Continuity of passenger mobility following disruption of the transport system" a working document of the European commission

All the above points will be positively impacted by the deployment of technical solutions based on the modelling and algorithmic strategies considered in this deliverable.

A major positive development underway is that mobile, personalized and ubiquitous computing along with the "pervasivity" of social networks are now allowing mathematically based models to use online data addressing specific needs and drive a 2-way communication platform in order to address effectively the issues "Information" and "Respect of passengers rights" .(Candelieri et al, 2015a; Candelieri et al. 2014b).

When assessing infrastructure performance, the level of physical damage of infrastructure components, usually predicted through fragility curves, is less important than the quantification of functionality losses, while both are essential for evaluating resilience and defining restoration strategies.

In (Gehl et al., 2016) a component-based approach for the derivation of fragility functions for infrastructure elements has been proposed. In particular, the effect of the **multi-hazard fragility** of bridges on the performance of road networks is investigated. This approach is important as it enables:

a) the evaluation of the damage in terms of functional losses and downtime duration;
b) the harmonization of the fragility models between several hazard types, while accounting for potential accumulation of damages.

Although focus of the quoted study is on bridges, the proposed methodological framework is valid for analysing other network elements, such as tunnels or road segments. Considerations provided by the paper are general and address the four cornerstones of resilience:

- ***knowing what to do***,
- ***knowing what to look for***,
- ***knowing what to expect***,
- ***and knowing what has happened***;

thus defining a broad socio-technical perspective focusing on how the anticipation provided by vulnerability analysis must interact with the monitoring (*knowing what to look for*), responding (*knowing what to do*) and learning (*knowing what has happened*) abilities in order to contribute to the overall goal of enhancing the resilience of the transport system.

The four cornerstones are also addressed by the first release of ERMG (deliverable D3.5); the present deliverable is aimed at creating a set of back-end functionalities, in particular algorithms and models for user profiling and network analysis, to support the implementation of ERMG functions for the resilience management.

With respect to user profiling the aim is to provide an innovative method to match the skills of the users (of the RESOLUTE platform) with the operating needs during the emergency. Furthermore, (mobility) data mining approaches are also proposed to characterise the behaviour of UTS's users and support the evaluation of response to disruptions, such as delays and interruptions.

Results from graph theory and network science offer the possibility to develop and use tools for supporting decisions in UTS resilience management, including preparedness, response, recovery and adaptation. Most research addresses vulnerability – usually related to decision making during the pre-hazard phase – and it has been reaching a relevant level of maturity and sophistication. Regardless its maturity level, the adoption of vulnerability research findings by planners, practitioners and operators is still limited.

RESOLUTE aims at supporting the operationalisation of resilience concepts by offering tools based on graph theory and network science research, for more consolidated vulnerability analysis and more recent resilience evaluation.

These tools are only methodological components of a more holistic and cross-disciplinary overall approach going beyond the topological analysis of the infrastructure and the analysis of data related to its usage.

2.2 Mapping ERMG functions to Application Framework's functionalities and services

In Figure 1, the RESOLUTE's architecture is reported, with user profiling and network analysis modules (i.e. application framework) highlighted. They are both placed in the "Mission critic layer" and are part of the back-end systems providing functionalities to deliver decision support

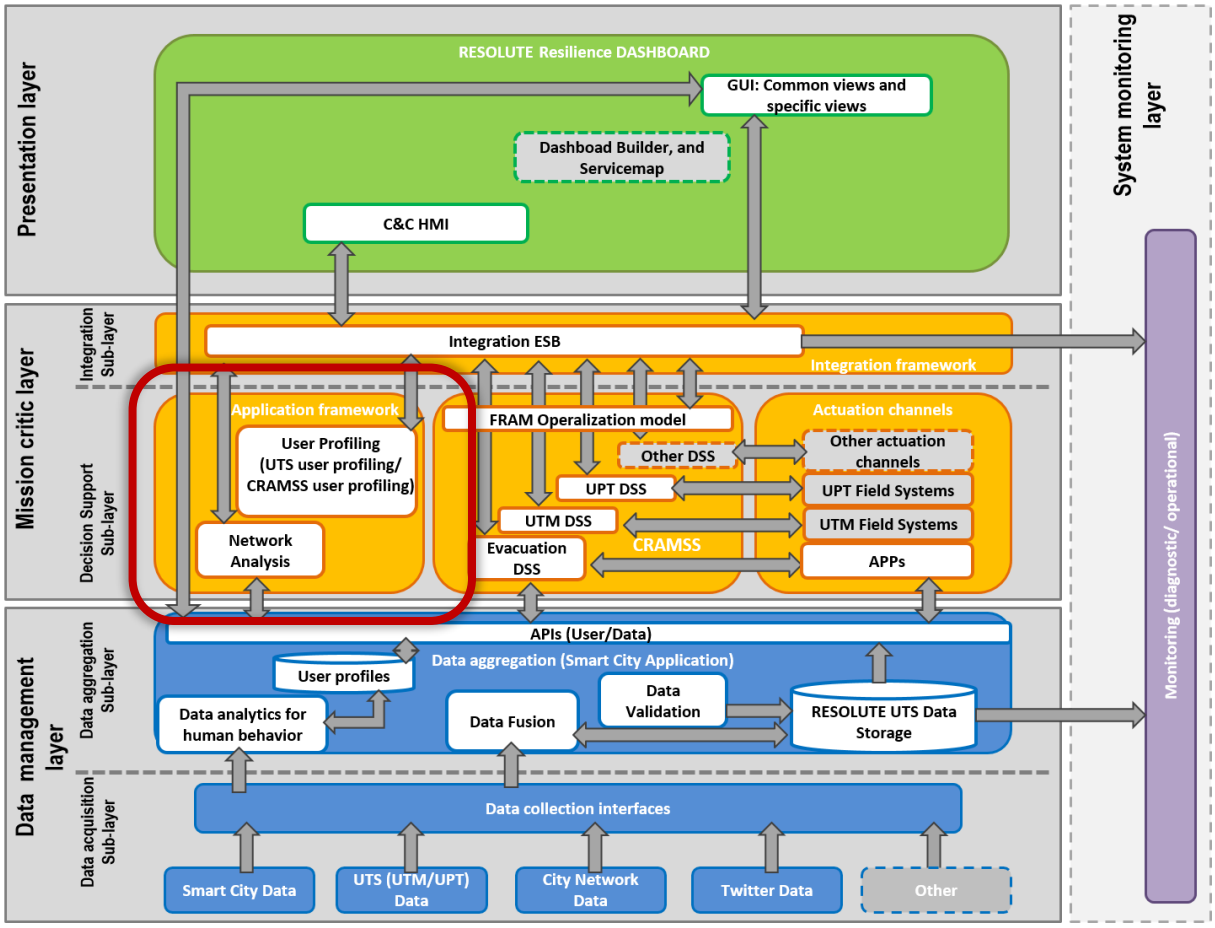


Figure 1 – RESOLUTE Architecture: Application Framework highlighted

With respect to the ERMG defined in D3.5, the application framework will offer algorithms and models to support the following functions:

ANTICIPATE

- ***Perform risk assessment*** – with respect to possible causes and associated impact to/response of the physical transportation network and/or service level, analysed through network science. Moreover, based on the kind of the processed data (e.g. real-time location of users), their results could also help to perform risk assessment. Huge concentration of people in the same place may constitute a risk factor.
- ***Manage awareness and user behaviour*** – improving them through algorithms for the characterization, simulation and analysis of different behaviours and skill-based management.

- ***Maintain physical/cyber infrastructure*** – through the analysis of the networked infrastructure, the identification of critical components (nodes/edges), the possibility to model cascading effects, etc. This will allow to better understand how to manage financial resources for improving asset management at both physical and cyber level.
- ***Develop Strategic Plan*** – the derived knowledge from the users profiles (e.g. type of users, amount, special needs, etc.), can help to the development of strategic plans that are adapted to the needs of all kind of user. The users profiling can give a better understanding to the emergency responsibles considering the number, the characteristics and the special needs of each group of people, in order strategic plans that manage each one group to be developed.

MONITOR

- ***Monitor Safety & Security*** – During emergencies the module can be used in order to monitor the movements of the ESSMA users and their condition.

RESPOND

- ***Restore/repair operations*** – modelling and analysing the UTS as a graph will enable a number of analyses which can support the definition of the most suitable restore/repair operations for recovery, as well as improve both asset management and quality of services. This includes, for instance, dealing with the bus bridging problem.
- ***Coordinate emergency actions*** – During an emergency, by using the described module, the efficiency of the operator is significantly increased for handling large amounts of users collectively. Thus the operators can choose certain actions/guidelines to be sent to users with same/common behavior, while special treatment/guidelines can be sent to unique or "anomalous" users.

LEARN

- ***Provide adaptation and improvement insights*** – through the analysis of UTS users mobility data, to better understand and characterize their behaviour in normal conditions as well as during disruptions, emergencies and disasters and through network analysis, in order to study how to improve both the physical infrastructure and the quality of service.

Regarding the RESOLUTE's architecture, displayed in Figure 1 and described in detail in the deliverable D4.1, the user profiling module is placed within the "*Mission critic layer"* and more specifically in the *"Decision Support Sub-layer"* and the *"Application framework*". The module communicates with the *"Data management layer"* for retrieving input data and distributes its results to the CRAMSS through the *"Integration ESB"*.

The inputs of the user profiling module are retrieved from the collected data of all the Emergency Support Smart Mobile Application (ESSMA) users. The data is composed by historical data, stored in the ESSMA Database, as well as by real time data retrieved direct from the online ESSMA users and are sent to the user profiling module through the evacuation decision support system (eDSS). The results of the profiling procedure are pushed to the integration ESB and can be retrieved from each one of the interested CRAMSS's components. The results can be

used for computation purpose as well as for illustrative. For instance, the eDSS presents the formatted clusters in a visual way, offering to the evacuation responsible an interactive way for handling huge numbers of users collectively.

The Network Analysis module, also placed within the "*Mission critic layer"* and more specifically in the *"Decision Support Sub-layer"* and the *"Application framework*", retrieves from *"Data management layer"* the basic input data, in particular structural and service information about the UTS, and also communicate with other modules through the *"Integration ESB"*. In particular, it retrieves from ESB messages related to modifications in the UTS (e.g. unavailability of one or more stations/stops, interruption of a line, etc.) and provides, on demand, the output of the analysis on the dynamically modified graph-based model of the UTS. Output of the analysis is a list of nodes and edges identified as "critical" with respect to a set of measures computed on the current status of the UTS. These outputs can be then interpreted by the module which performs the analysis request; this allows for using the network analysis module in different contexts and, more important, to make visualization independent on the analysis.

The following Figure 2 summarizes the computational modules within the Application Framework as well as the interactions with the Data Management Layer and the ESB.

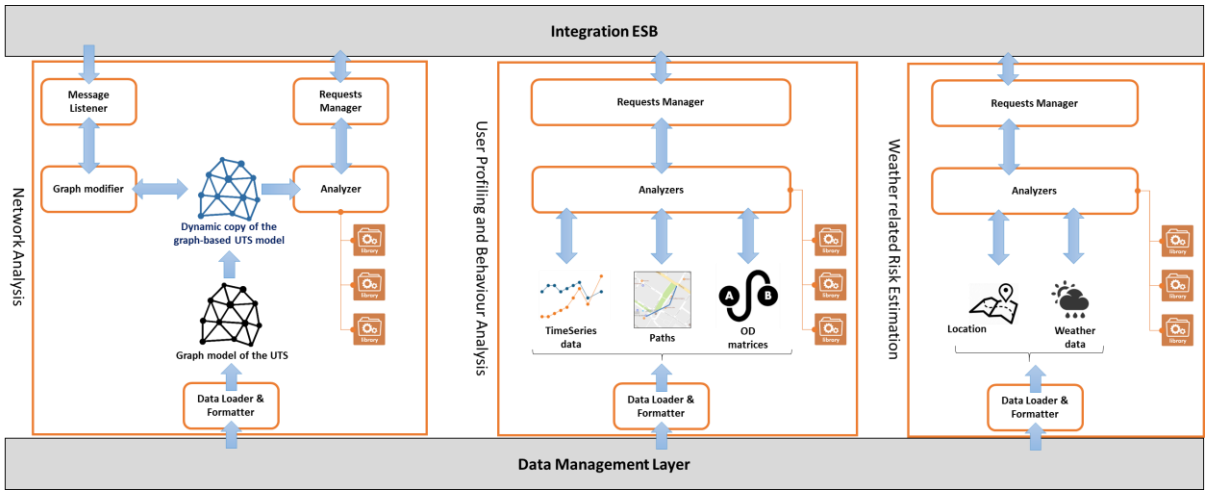


Figure 2 – Modules within the Application Framework

3 SUITABLE AND AVAILABLE DATA

This section summarizes available sets of data used to design, develop and preliminary test – offline – the Application Framework. The sets of data include a sample of those presented in D4.2 and some open-data repository.

3.1 Data available

The larger and more complete source of data is the Knowledge Base presented in D4.2, in particular with respect to the access to network data used for implementing network analysis algorithms and models. Some open-data initiatives related to urban transport networks topological data have also been taken into account.

With respect to the user profiling analysis, focused on RESOLUTE's users, a synthetic dataset of agent paths has been generated for development and testing purposes. An open-data repository, namely the passenger counts of the London Underground, has been used for testing algorithms to analyse mobility behaviours of UTS's users.

3.1.1 Data for User Identity and Profile Management

In order to investigate methods for automatic detection of different mobility behaviour of agents, a synthetic dataset of agent paths through a city was generated. A number of 20 agents were considered, split into two groups:

- One group of 5 agents with no mobility problems,
- One group of 15 agents with mobility problems.

The group of agents with mobility problems was further divided into 3 subgroups, one of older people, one of blind people and one of people using a wheelchair.

For each agent, a set of 100 paths was generated, simulating the routes that the agents follow within a city, forming a set of 2000 paths in total. In order to generate the paths, the city's road network was considered as an undirected graph, where the crossroads are the vertices and the roads connecting the crossroads are the edges connecting the vertices. Each vertex of the graph is associated with the geographical coordinates (longitude, latitude) of the corresponding crossroad. Then, each path was generated by considering a starting vertex and performing a random walk in the graph, with equal probabilities of following each edge connected to a vertex. The starting point was the same for the paths of a single agent, with different agents having different starting points. The length of the path, i.e. the length of the random walk on the graph, was selected randomly for each path, picked from a uniform distribution. However, the mean value of the length distribution was different for agents with no mobility problems and impaired agents:

- For agents with no mobility problems, the mean length was set to 75 random walk steps.
- For agents with mobility problems, the mean length was set to 15 random walk steps.

3.1.2 Data for Network Analysis

With respect to data to use for network analysis, and available in the RESOLUTE project, some samples of the data in the Knowledge Base have been considered. As described in D4.2, these data are related to:

- ***Smart City Data***: it consists of a collection of Open and Private Data coming from the city and territory. The major part of this kind of information is published by governmental organizations as Open Data, in different file formats such as html, xml, csv, shp, etc., and typically provides information that may present links to web resources. Moreover, the information is usually static, but can also be distributed in real time or semi-real time modality (e.g. number of visitors in a museum, city tours that are starting in a specific

time and are going to make a guided visit of the city, weather forecasts for the different municipalities, events in the city, etc.). To acquire a wide range of these different kinds of data, as described above, a set of ETL (Extract Transform Load) processes will be realized. The velocity of data ingestion is related to the frequency of data update, and it allows distinguishing static from dynamic data. In the City of Florence (and in some cases in all the Tuscany) there is available data coming from the Public Administrations and typically covering:

  - location of points of interest (POIs) on the territory (including museums, touristic attractions, restaurants, shops, hotels, etc.);
  - major governmental services;
  - ambient data;
  - weather status and forecast;
  - dataset strictly connected with resilience aspects, such as: i) underpasses, bridges, etc. ii) risks and vulnerability analysis; iii) relations with Critical Infrastructures; iv) wi-fi flows (people); v) flood levels; vi) landslides and earthquakes, etc. vii) accidents, flooding, presence at schools, etc.
  - Other data

- ***UTS (UTM/UTP) Data:*** it will consist of a collection of a wide set of data regarding mobility and transport aspects in a smart city. Some examples are:
  - Intelligent Transportation Systems, ITS, for bus/train/ferry/tram/etc. management. Moreover, they also describe and model solutions for managing and control.
  - Traffic flows, people flows
  - Other
- ***City Network Data:*** it will collect a set of data coming from mobile Apps. This kind of channel, in a smart city context, is fundamental in order to take care of the people/citizen's presence and needs. It can be useful for receiving real time information coming from:
  - the Wi-Fi, or sensors networks having information related to the citizens' habits, collecting their activities that are fundamental, for example in order to study the citizen's flows, to establish what are the citizens' preferred services, what can be improved for them, to collect their ideas and necessities, etc.

In more detail, the public transportation network in Florence has been considered as a sample dataset to design, develop and preliminary test the algorithms and models of network analysis.

3.2 Other open-data available (outside RESOLUTE)

As recently stated by the Open Data Institute (ODI) in its recent report *"How can we improve urban resilience with open data"*, open data can boost urban resilience[2]. The ODI, teamed up with the Canadian body Open North, focused on how open data can be used to beef up cities' resilience planning, emphasizing the need for robust local and global data infrastructure, including the relevant organizations, datasets, technology, training, policies and regulations.

The report also highlights current barriers and limitations. While the urban resilience and open data communities are looking at similar issues, more work needs to be done to couple their efforts and build a culture of openness. One way to overcome this challenge is to use existing trusted networks to encourage collaboration and bring together the open data and urban resilience efforts.

[2] http://www.ukauthority.com/data4good/entry/6770/odi-says-open-data-can-boost-urban-resilience

- **General Transit Feed Specification** (GTFS), also known as *GTFS static* or *static transit* to differentiate it from the *GTFS real-time extension* – defines a common format for public transportation schedules and associated geographic information. GTFS is supported by Google and GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way. The feeds are represented in a series of text files that are compressed into a ZIP file, and include information such as fixed-route schedules, routes, and bus stop data. GTFS datasets are used in a variety of types of applications, including trip planners, such as Google Maps, mobile applications, timetable generation software, tools for transit planning and operations analysis, and other categories of applications outlined in this article. Figure 3 shows a diagram summarizing the GTFS data model.

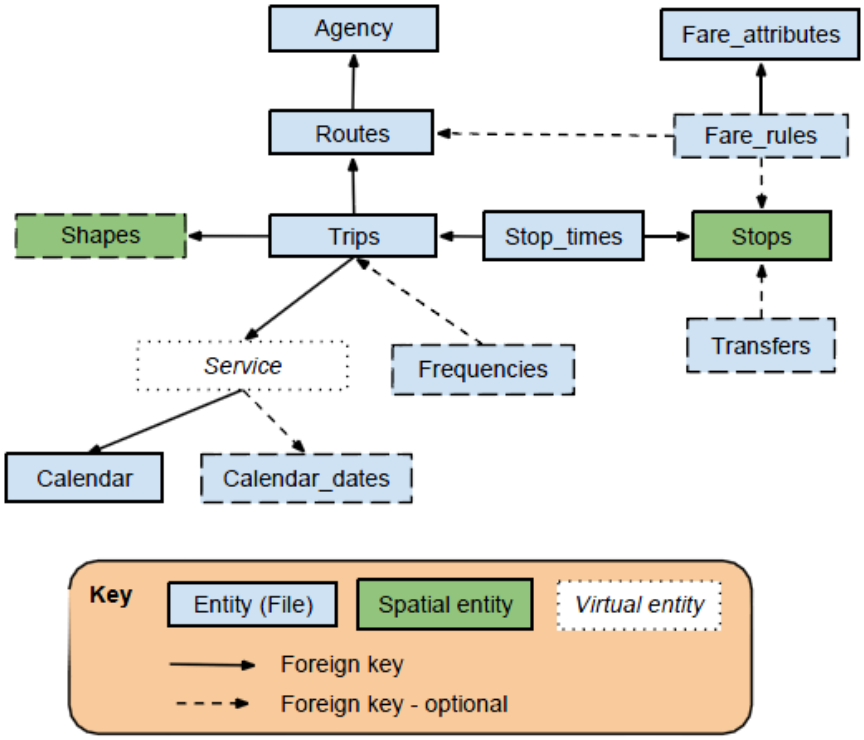


Figure 3 - GTFS data model diagram

A relevant open-data repository of GTFS data is available online [3] ("publicly-accessible public transportation data):

- **London Underground passenger counts** – All public Transport for London (TfL) data is freely released (open-data) for developers to use in their own software and services[4]. TfL encourages software developers to use these feeds to present customer travel information in innovative ways - provided they adhere to the transport data terms and conditions.
  A set of data is available online[5] that consists in passenger numbers entering and exiting London Underground stations, sampled at 30 minutes intervals, largely based on the Underground ticketing system gate data. This set of data is similar to the one used in (D'Lima & Medda, 2015); currently data

[3] https://www.transitwiki.org/TransitWiki/index.php/Publicly-accessible_public_transportation_data

[4] https://tfl.gov.uk/info-for/open-data-users/

[5] http://tap.data.tfl.gov.uk/

available on the web refers to 2015 (entries and exits, every 30 minutes, for every station and for a period of two weeks, 11 to 24 October 2015) instead of 2012, which are those analysed in the aforementioned paper.

- **World Weather Online API** – This weather API provides a simple way for developers and programmers to embed weather data into apps and websites. They are free for personal as well as commercial use depending on which license one wishes to use. The Premium Local Weather REST API method allows one to access current weather conditions and the next 15 days of accurate weather forecast at city level granularity. The Local Weather API returns weather elements such as temperature, precipitation (rainfall), weather description, weather icon and wind speed. Aside forecasting, the service is capable of returning historical data for the past 12 years.
- **Urban Transportation Network of the Attika region** – All the data related to the UTS in the Attika region is freely available through API[6], in particular lines, routes, stops as well as schedules and bus position.

3.3 Other relevant useful data – potentially available after RESOLUTE

Mobility and trajectory data will represent the most recent and huge source of useful information to analyse the behaviour of UTS's users both in normal conditions and during emergency management. The derivation of trajectories can be divided into four major categories, according to what has been proposed in (Zheng, 2015) together with few application scenarios for each category.

Trajectory data representing human mobility can help build a better social networks and travel recommendations.

- **Mobility of people**: people have been recording their real-world movements in the form of spatial trajectories, passively and actively, for a long time.
    - ***Active Recording***: People log their travel routes through GPS to record a journey and share experiences with friends. For instance, bicyclers and joggers record their trails for sports analysis, and in Flickr, a series of geotagged photos can formulate a spatial trajectory as each photo has a location tag and a time stamp corresponding to where and when the photo was taken.
    - ***Passive Recording***: A user carrying a mobile phone unintentionally generates many spatial data represented by a sequence of points associated to cell towers along with corresponding transition times. Furthermore, transaction records of a credit card indicate spatial trajectory of the cardholder, as each transaction contains a time stamp and a merchant identifier denoting the location where the transaction occurred.
- **Mobility of transportation vehicles**: A large number of GPS-equipped vehicles (such as taxis, buses, vessels, and aircrafts) are common in our daily life. For instance, many taxis in major cities have been equipped with a GPS sensor, which enables them to report a time-stamped location with a certain frequency. Thus, a large amount of spatial trajectories are collected and then used for resource allocation (Yuan et al. 2011b, 2013b), traffic analysis (Wang et al. 2014; Yuan et al. 2013a), and improving transportation networks (Zheng et al. 2011a).
- **Mobility of natural phenomena**: Meteorologists, environmentalists, climatologists, and oceanographers collect trajectories of natural phenomena, such as hurricanes, tornados and ocean currents, in order to capture changes of the environment and climate and, consequently, protect the natural environment and support natural disasters management.

[6] http://oasa-telematics-api.readthedocs.io/en/latest/

4 USER IDENTITY AND PROFILE MANAGEMENT

This chapter presents the approaches for user behaviour analysis and user profile management implemented in RESOLUTE. This chapter is therefore divided in two macro chapters: the first focuses on the profiling of RESOLUTE's users, and aims at supporting a more effective skill-based management of the teams, in particular during emergency management, and the second focuses on characterizing mobility behaviour of UTS' users.

4.1 RESOLUTE's users identity and behaviour- & skill- based profiling

This work will lead and direct the development of a user profile management system for improving the recommended quality of the system. The aim is to design a new user profile model based on individual behaviour and skill information, along with a recommendation system based on such a model, and finally to evaluate the recommendation performance of the proposed model in terms of widely adopted evaluation metrics.

The goal of mobility behaviour analysis is to extract mobility features from each agent, so that they can be represented as points on a low-dimensional space and be visualized or clustered.

In order to construct informative mobility profiles for the agents, multiple types of mobility behavior can be used, such as the length of the paths taken by an agent, the average speed of the agent, the randomness of the agent's moves, etc. In this respect, a data item, e.g. an agent, is considered as a multimodal object, consisting of multiple behavioral features. Combining these multiple features is crucial for the determination of similarities and differences in the mobility behavior of various agents.

In the literature, the combination of multiple types of information for purposes such as classification, clustering, visualization, etc., has been handled by so-called multimodal fusion methods (Atrey et al., 2010). A common approach for combining the available data features, so-called modalities, is by projecting the multiple features to a common space, where the specific task can be solved as if the data were of a single feature type. This projection is accomplished by first fusing either the unimodal features themselves, or distances between them, usually using a weighted sum. The weights of the sum correspond to the relative importance of each modality for the task at hand. In (Yang et al., 2009), such a weighted sum is computed for the distances between the respective modalities of two multimodal objects, while in (Tong et al., 2005), the combination is performed for the Laplacians of similarity graphs constructed from the data. The work presented in (Lin et al., 2011) deals with multimodal dimensionality reduction and addresses it using Multiple Kernel Learning (Gonen and Alpaydin, 2011). A multimodal kernel matrix is formed as a weighted sum of multiple unimodal kernel matrices and is then used to guide dimensionality reduction.

Another approach of multimodal fusion is to formulate the problem as an optimization problem for one of the modalities and to use information from the other modalities as constraints for the optimization. Such an approach is followed in the co-training setting (Blum and Mitchell, 1998). The problem is considered as an optimization loop, where, at each iteration, a different modality is considered, thus utilizing all types of features for the determination of the final solution. The authors of (Nigam and Ghani, 2000) present the co-EM algorithm, which combines the co-training algorithm of (Blum and Mitchell, 1998) with the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). The iterations of EM are alternated between the two modalities, so that the features of one modality assist in classifying the features of the other modality.

A different approach from the above is presented in works such as (Kalamaras et al., 2014; Kalamaras et al., 2015; Kalamaras et al., 2015b), where the problem of multimodal data visualization is formulated as a multi-objective optimization problem (Ehrgott, 2005; Coello et al., 2007). Such an approach allows to simultaneously optimize all objectives, resulting to set of Pareto-optimal solutions, instead of a single one. Such methods have been shown to

be more generic than the ones mentioned earlier, and to discover more effective solutions in cases where the weighted-sum-based methods cannot perform as well (Kalamaras et al., 2014). Thus, this multi-objective based approach will be used hereby for the combination of the multiple mobility features extracted from the agent data.

4.1.1 Notation

A path P is formally defined as a set of ordered points

$$P = \{p_1, p_2, \dots, p_L\}, p_k \in \mathbf{R}^2$$

The length L of each path is selected randomly, depending on the type of agent, as described in the previous section. For the purposes of mobility behaviour analysis, each agent A_i , e.g. an older person, is defined as a set of paths

$$A_i = \{P_{i,1}, P_{i,2}, \dots, P_{i,N_i}\}$$

which are the paths that this agent has taken within a course of several days, hereby generated as described in the previous section. Each path of an agent may be of different length. The number of paths each agent has taken, N_i, is hereby set to 100 for all agents, in the dataset used. However, the following analyses and features can also handle different numbers of paths per agent.

4.1.2 Mobility Features

The desired mobility features extracted from an agent are numerical descriptors, in the form of scalars or vectors, able to capture various characteristics of the mobility profile of each agent. In the current analysis, three types of features are considered:

- The average path length of an agent;
- The entropy of the paths of an agent, when collected in a 2D histogram;
- Geometric features based on the Frechet distance between paths.

In the following subsections, these features are presented in more detail.

Average path length features

The first feature used is a very simple one: the average path length of the paths taken by each agent:

$$f_{i,\text{length}} = \frac{1}{N_i}\sum_{j=1}^{N_i}\left|P_{i,j}\right|$$

The average path length feature is a positive scalar value. Agents with limited mobility capabilities are expected to move in paths of smaller length, so the average length feature can be used to discriminate between agent mobility profiles.

Agent paths entropy features

The second feature used is the agent paths entropy. It captures the randomness of the paths taken by an agent. It discriminates between agents who usually move in specific paths from agents whose behaviour is more randomized and unexpected.

For the paths entropy feature to be computed, a 2D histogram of the places from which an agent passes is first constructed. The procedure for computing the 2D histogram is depicted in the Figure 4. First, the paths of an agent are used to compute the limits of the rectangular area covered by him/her. The rectangular area is quantized into a number of bins, used to construct the 2D histogram. For the experiments presented hereby, the same number of bins has been used in the horizontal and the vertical direction, resulting in a square grid. The selection of the number of bins has been left as an external parameter and is hereby fixed to 100 for each dimension (i.e. 10000 square bins).

For each point of a path, the bin containing the geographical coordinates of the corresponding map point is found and is incremented by one. In order to also include the points belonging in the line between two crossroads, a linear interpolation is performed in each path, in order to also consider intermediate points between two path nodes.

The above procedure creates a 2D histogram of the bins from which the path passes through. Continuing this process to also include all the paths of an agent, results in a 2D histogram of the bins from which this agent has passed through at any time. Bins with high values correspond to geographical locations from which the agent has passed most frequently, while bins with small values correspond to less frequent locations.

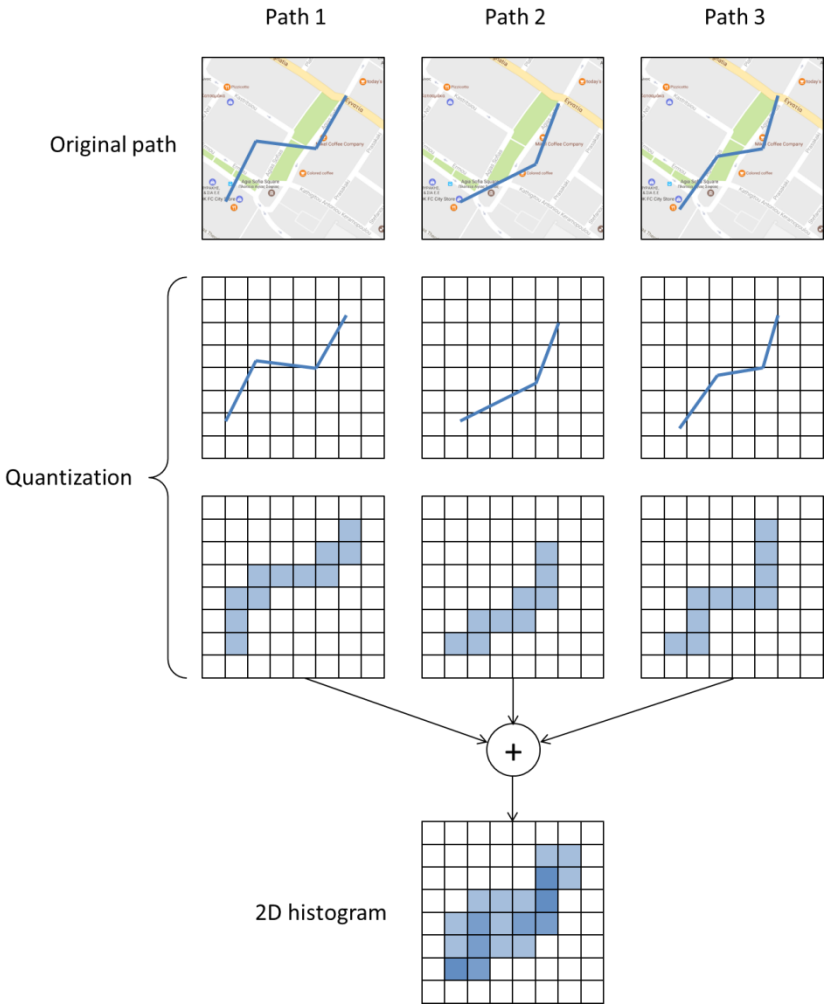


Figure 4 - Procedure for creating the 2D path histogram for an agent. The area in which an agent moves is quantized and the bins from which he/she passes are found. The final 2D histogram results from accumulating in each bin the number of paths that pass through it. Bins used frequently by the agent have higher values (here shown in darker colors).

Different agents may have different behaviour with respect to the distribution of the locations from which they pass, which are reflected in the 2D path histogram. Agents that generally visit the same locations repeatedly will have a 2D histogram with a small number of bins having large values. Such agents have a generally predictable mobility behaviour. On the other hand, agents whose mobility behaviour is rather unexpected and unpredictable, tend to visit many different places, thus their histograms will contain a large number of bins with relatively low values.

As an example, Figure 5 depicts the 2D histograms of an agent with no mobility and an agent with mobility issues, for comparison, created for agents of the generated dataset. Each image consists of a grid of 100x100 square bins, where the colour of the bin denotes the value of the corresponding histogram bin, following a grayscale from white (zero) to black (the highest value in the histogram).

An agent with mobility issues, as shown in the figure 4(b), follows a limited number of standard paths, thus forming a histogram containing a small number of non-zero bins having large values (dark colours). On the other hand, the distribution of visited places for an agent of no mobility issues is more uniform, with many non-zero bins having small values. This shows that this agent has a rather unexpected mobility behaviour. It should be noted however that such behavioural differences are not only due to mobility problems, but may also be determined by personal preferences of the individual agents.

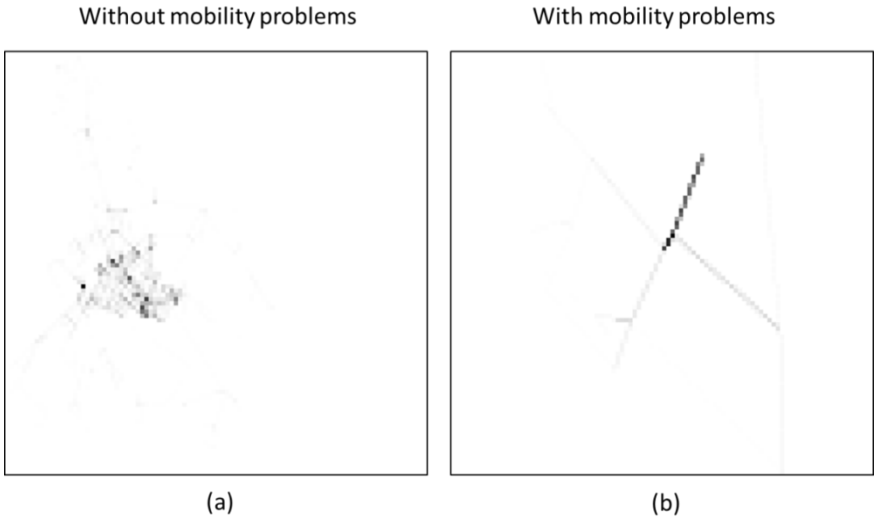


Figure 5 - Examples of 2D path histograms for (a) an agent with no mobility issues, (b) an agent with mobility issues. Each pixel in the images is a square bin of the 2D histogram. Darker values correspond to higher values, i.e. more frequent locations. The agent with mobility issues (b) visits a rather limited number of places, following a small number of standard paths, which is apparent by the small number of non-zero bins with high values (dark colors). The agent with no mobility issues (a) has a more unpredictable behavior, which is apparent from the large number of non-zero bins with relatively low values.

In order to quantitatively express this amount of "randomness" of the 2D path histogram of an agent, the entropy of the histogram is computed. This entropy is used as a scalar feature characterizing a specific agent. First, the histogram is considered as a 2D probability distribution, by normalizing its values so that they sum up to 1. Let then

$$Q^i(x, y), \qquad x = \{1 \dots B_x\}, y = \{1 \dots B_y\}$$

denote the 2D probability distribution for agent i, where B_x and B_y are the numbers of bins in the horizontal and vertical dimensions. For the experiments presented below, $B_x = B_y = 100$.

The entropy feature of agent i is then computed as the entropy of $Q^i(x, y)$:

$$f_{i,\text{entropy}} = -\sum_{x=1}^{B_x}\sum_{y=1}^{B_y} Q^i(x,y)\log Q^i(x,y)$$

The entropy feature is a positive scalar value. Large entropy values correspond to larger "randomness", i.e. to agents having more unpredictable behaviours.

A third mobility feature was also used in the hereby presented experiments, based on the Frechet distance between paths. The Frechet distance is a measure of similarity between two arbitrary curves, in terms of their geometry, i.e. how close one curve passes with respect to another. Intuitively, it can be described by considering a man walking along one curve and his dog walking along the other. The man holds his dog with a leash. The Frechet distance is the minimum required length of the leash, so that both the man and the dog are able to walk along their own curves, while the man always holds the dog by the leash. The man and the dog are not allowed to walk back in their curve, but they can move at any speed or even stop while waiting the other to move forward. Low values of the Frechet distance between two curves mean that the curves are kept close to each other throughout all their lengths (i.e. a short "leash" is enough), while high values of the Frechet distance mean that the curves largely deviate from each other (i.e. a long "leash" is required).

The Frechet distance is generally defined for arbitrary curved paths. Hereby, the discrete implementation presented in (Eiter and Mannila, 1994) was used, which is a specialisation for polygonal paths, suitable for the paths followed by the simulated agents.

Bringing the concept of Frechet distance between two paths a little further, a distance measure between *agents* can be defined, based on the distances between the paths that they follow. Let $\text{frechet}(P_{i,u}, P_{j,v})$ denote the Frechet distance between path u of agent i and path v of agent j. The distance measure between two agents A_i and A_j, namely $\text{agentFrechet}(A_i, A_j)$ is defined as the mean value of the distances between all pairs of paths between A_i and A_j:

$$\text{agentFrechet}(A_i, A_j) = \frac{1}{N_i N_j}\sum_{P_{i,u}\in A_i}\sum_{P_{j,v}\in A_j}\text{frechet}(P_{i,u}, P_{j,v})$$

A small value of the above distance measure for two agents means that their corresponding paths are close to each other on average, i.e. they exhibit similar mobility patterns, while a large value denotes agents with very different mobility patterns.

In order for the frechet distance calculation between two paths to be unbiased, the path coordinates are first normalized to the [0, 1] range, as if each path was scaled to fit in a square of size 1. This removes any bias caused by paths starting at completely different locations, as well as bias due to the path length, and concentrates of the form of the path.

Having a distance measure between two agents, i.e. the agentFrechet measure, a distance matrix can be computed, containing the value of the agentFrechet distance measure for all pairs of agents into the dataset using Multidimensional Scaling (MDS) (Cox and Cox, 2000). This distance matrix can be used to compute coordinates for each agent in a new, possibly high-dimensional, space. In this space, each agent is represented as a point, so that points that are close to each other correspond to agents with small agentFrechet distance between them. These coordinates are used hereby as the third kind of feature:

$$f_{i,\text{frechet}} = \text{MDS}(A_i)|_{\text{agentFrechet}}$$

where the notation in the right-hand side of the above denotes the coordinates computed for A_i using Multidimensional Scaling with a distance matrix based on the $\mathrm{agentFrechet}$ distance measure, as described above. The above feature is a high-dimensional vector descriptor for an agent, encoding characteristics of the agent regarding his/her mobility differences from all other agents.

In order to further investigate differences in mobility behaviour, and since the individual paths of the agents are the basis which characterize the mobility behaviour of the agents, a similar procedure has also been performed for individual paths, instead of agents. The Frechet distance between paths was used to compute a distance matrix among all pairs of paths (i.e. 2000x2000), and was used with MDS, in order to compute coordinates for the individual paths in a new space. This mapping makes possible the comparison of individual paths in terms of the Frechet distance among them and the detection of any patterns that may exist. Results of this kind of analysis are presented towards the end of the next chapter.

4.2 (Mobility) Data Mining and behavior-based profiling of UTS's users

Even if ***mobility data mining*** (Giannotti and Pedreschi, 2008) is a quite recent research field, a significant amount of approaches, sharing similar objectives, have already been proposed, in particular with respect to the mobility clustering literature, aimed at discovering sets of paths that are similar. Some of the first approaches adapt classical distance based algorithms and define ad-hoc distances for paths data (Pelekis et al., 2009), possibly with limited ad-hoc refinements (Andrienko et al., 2009). Based on this kind of approaches, a recommendation system has been proposed (Xiao et al., 2010) where the user trajectories are translated into sequence of regions of interest and then compared.

Analysis of trajectory data, in particular, proved to be a highly multidisciplinary field, ranging from Physics to Sociology, Transportation Research and Computer Science (Giannotti et al., 2011; Giannotti Pedreschi 2008; Renso et al., 2013; Zeng & Xie 2010). Recently, thanks to the advances in location-acquisition and mobile computing techniques enabling the acquisition of massive spatial trajectory data related to moving "entities" (people as well as vehicles), many new approaches have been proposed for ***processing***, ***managing***, and ***mining*** these data, fostering a broad range of applications. A systematic survey on the major research into ***trajectory data mining*** (e.g. ***trajectory pattern mining***, ***outlier detection***, and ***trajectory classification***) is presented in (Zheng 2015).

Analysing large trajectory datasets has been carried out towards different directions. Historically, basic statistics have been be applied to trajectory data mainly to infer origin-destination matrices (Calabrese et al., 2010), while other studies have been focused on trajectory data mining, aiming at finding correlations in large datasets of positioning data (Giannotti & Pedreschi, 2008; Zheng 2015).

Techniques to extract movement patterns include:

- **clustering discovery** - finding groups of objects moving together (Nanni & Pedreschi, 2006);
- **sequential pattern discovery** - finding the most frequent sequences of places visited;
- **flock detection** - extracting the convergence of people moving together for a certain amount of time (Giannotti & Pedreschi, 2008; Wachowicz et al. 2011). In (Trasarti et al., 2010), a software called M-Atlas encompassing a series of trajectory data mining algorithms is presented.

Analysing trajectory data usually requires a number of pre-processing steps, such as **noise filtering**, **segmentation**, and **map matching** (a.k.a. ***trajectory pre-processing***). *Noise filtering* is aimed at removing noisy "points" from a trajectory due, for instance, to poor signal quality of the location positioning systems. *Trajectory segmentation* is aimed at dividing a trajectory into segments according to time intervals, spatial shape, or semantic meanings; this step is strictly required to enable analytical processes such as *clustering* and *classification*. Finally,

map matching aims to project each point of a trajectory onto a corresponding road segment, where the point was truly generated.

Following trajectory pre-processing, a number of different trajectory/mobility data mining tasks can be performed; a possible categorization is (Zheng 2015):

- **Trajectory Uncertainty**: entities move continuously while their locations can only be updated at discrete times, leaving the location of a moving entity between two updates uncertain. Two research streamlines are ongoing: the first is aiming at managing/reducing uncertainty, while the second is devoted to protect users' privacy when they decide to disclose their mobility data.
- **Trajectory Pattern Mining**: the aim is to analyse the mobility patterns of moving entities, represented through individual trajectories, characterized by specific patterns, or a group of trajectories sharing similar patterns. Some specific activities are related to the analysis of patterns moving together, clustering trajectories, finding periodic/frequent patterns.
- **Trajectory Classification**: this task is to classify trajectories or segments into some categories, which can be activities (like hiking and dining) or different transportation modes, such as walking and driving.
- **Trajectory Outlier Detection**: trajectory outliers (i.e. anomalies) can represent relevant items (a trajectory or a segment) significantly different from other "typical" items, according to some predefined similarity metrics. Items might also be events or observations (i.e. collections of trajectories) that do not conform to an expected pattern (e.g., traffic congestion caused by a car accident).

Finally, besides studying trajectories in their original form, they can also be transformed into other formats, such as graph, matrix, and tensor. The new representations of trajectories expand and diversify the approaches for trajectory data mining, leveraging other existing mining techniques (e.g., graph mining, Collaborative Filtering, Matrix Factorization, and Tensor Decomposition).

Recently, efforts in the research community have been focused on developing techniques to analyse and better understand ***human mobility***. As a first example, a "**stay point detection"** algorithm allows for the identification of the location where a moving entity has stayed for a while within a certain distance threshold. A stay point could be, for instance, a restaurant or a shopping mall where a user has been; this result provides some more semantic information with respect to the sequence of "simple" points into a trajectory (Zheng 2015).

The possibility to complement raw mobility/trajectory data with additional information from the context may further support a better understanding of human mobility. Interpreting trajectories of persons within a city, by integrating some knowledge about the features of the city (e.g. map, points of interest), allows for replacing spatiotemporal coordinates with street and crossing names, or with names of places of interest, such as shops, restaurants, and museums, enabling the also known as **semantic enrichment process** (Parent et al., 2013; Chen et al. 2015; Liao et al. 2015). A **semantic trajectory** is the representation of the trajectory and a set of interpretations on the moves and on the stops made by the moving entity during its journey. Thus, the same trajectory can have different interpretations depending on the context of the application and the modelled domain.

Analysing semantic trajectories facilitates the understanding of the mobility data and patterns; however, the privacy issue must be taken carefully into account when human beings are tracked. Indeed the location of a person may disclose private information that a user may not wish to make public. Privacy has been recognized immediately as a main concern when dealing with large amount of mobility data (Giannotti & Pedreschi, 2008; Renso et al., 2013). For this reason a number of data mining methods have been studied to find a trade-off between the need to get useful mobility patterns and the need to preserve the privacy of tracked individuals. These methods are referred to as "privacy-aware trajectory data mining" (Giannotti & Pedreschi, 2008; Renso et al., 2013).

With respect to the application of mobility/trajectory data mining for disaster management, a relevant approach is proposed in (Song et al., 2015). This specific research topic is highly challenging, due to the uniqueness of various disasters and the unavailability of reliable and large scale human mobility data. The proposed approach is based on the analysis of a large and heterogeneous set of data (i.e. 1.6 million users' GPS records in three years, 17520 times of Japan earthquake data in four years, news reporting data, transportation network data, etc.) with the aim to infer human emergency mobility following different disasters and then develop a general model of human emergency mobility for generating and simulating large amounts of human emergency movements. The experimental results and validations demonstrate the efficiency of the suggested simulation model, and suggest that human mobility following disasters may be significantly more predictable and can be easier simulated than previously thought.

According to the increasing frequency and intensity of natural disasters, understanding and simulating human emergency mobility during the event is becoming a critical issue for planning effective humanitarian relief, disaster management and long-term societal reconstruction.

A particular type of mobility data have been used to analyse resilience of the London Underground network with respect to minor disruptions, such as delays and temporary interruptions (D'lima & Medda, 2015). Mobility data, in this case, are not trajectories but diversion of passenger flow and/or crowding on platforms (i.e. passenger counts over time) due to a minor disruption to an underground line. Indeed, passenger counts may increase or decrease depending on the type of disruption and the affected line(s). Furthermore, depending on the severity of the disruption, the spikes in passenger counts could either be gradual or steep. The effect of the shock eventually dissipates as passengers respond to the shock, and passenger flows return to normally expected levels. Finally, the passenger count time series represent the state – and the evolution – of the underground transportation system, and are used to analyse the resilience of the system, computed as the speed with which the passenger counts return to normal, which is an indicator of how quickly the underground transport system is able to recover from the shock and, thereafter, resume normal operation.

5 MULTI-RISK AND NETWORK ANALYSIS ALGORITHMS-MODELS

In this chapter, network analysis algorithms and models are presented, with a specific focus on the analysis of a UTS. After some preliminary notations and background, relevant graph-based measures, for analysing both structural/topological and service characteristics of UTS, are presented. Finally, a mapping of relevant events (in particular disruptions, even of different nature) into modifications to the graph associated to the UTS is proposed, along with approaches to evaluate the impact of disruptions at structural and service level.

5.1 Network Analysis and resilience

With respect to network analysis algorithms and models, a very interesting and recent review on resilience and vulnerability analysis is proposed in (Mattsson & Jenelius, 2015). It is clearly reported that there is substantial literature on **vulnerability analysis** approaches for transport systems, while literature on resilience is less extensive. This conclusion is also supported by (Faturechi & Miller-Hooks, 2014; Khademi et al. 2015) who offers a comprehensive overview of transport system performance during disasters: a lot of research findings are related to the **assessment of critical components** in the networked infrastructure (vulnerability studies) but much less on **disaster management**. Studies on vulnerability mostly deal with the knowledge about ***what to expect***, which is fundamental for defining adequate proactive actions.

Although, resilience is a concept closely related to vulnerability, it defines a much broader socio-technical framework to cope with infrastructure threats and disruptions including **preparedness**, **response**, **recovery** and **adaptation** (Worton, 2012). Thus, different tools – as well as network analysis approaches – are needed to analyse and support decisions for **anticipation**, **prevention**, **mitigation** and **restoration**, depending on different types of disruptions.

5.1.1 Notations

In order to apply network analysis to the study of resilience of UTS, a suitable graph-based representation of the networked infrastructure must be provided. A graph is a very flexible modelling technique to represent any interconnected system, through nodes and edges – even with information associated to both nodes and edges. The most important advantage is that this representation allows for an abstraction of the specific real-world underlying networked infrastructure – it could be a UTS, a water distribution network, an energy/power grid, a gas/oil supply system, a telecommunication network, etc. – enabling the application of algorithms and models on the graph associated to any networked system.

Let denote a graph with $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each edge of G is represented by a pair of nodes (i, j) with $i \neq j$, and $i, j \in V$ and $i, j = 1, .., n$, where $n=|V|$. If $(i, j) \in E$, i and j are called **adjacent nodes**.

A graph G is **undirected** if (i, j) and (j, i) represent the same edge. A graph G is **simple** if no self-loops are admitted (edges starting from a node and ending on the same node) and only one edge can exist between each pair of nodes (i, j), with $i \neq j$.

The adjacency relationship between the nodes of G can be represented through a non-negative n x n matrix A (i.e., the **Adjacency matrix** of G). The entry a_{ij} of the adjacency matrix A is 1 if i and j are adjacent nodes (i.e., $(i, j) \in E$), and 0 otherwise. Furthermore, $a_{ij} = a_{ji}$ if G is undirected and a_{ii} (entries on the diagonal) are 0 if G is simple.

Let denote with ***degree*** of the node *i* the number of edges having *i* as one of the two nodes of the edge. Any one of the edges having *i* as one of its nodes is called **incident** on *i*. When the order of the nodes in the edge definition is important (**directed graph**), the degree of the node *i* can be split into ***out-degree*** (number of edges having *i* as first node) and ***in-degree*** (number of edges having *i* as second node).

A **path** from *i* to *j* is a sequence of distinct adjacent nodes starting from *i* and ending to *j*. The **shortest path** between *i* and *j* is the one related to the shortest list of adjacent nodes from *i* to *j*, and it is usually named **distance** *d(i, j)*. The largest distance among each possible pair of nodes in *G* is named ***diameter***.

A **connected graph** is a graph where a path exists between each pair of nodes $i, j \in V$. A subgraph $G' = (V', E')$ of *G* is a graph such that $V' \subseteq V$ and $E' \subseteq E$; a **connected component** of *G* is a maximal connected subgraph of *G*. If the original graph *G* is connected it consists of just one connected component.

In addition to mere adjacency, a **weight** $w_{ij} \geq 0$ can be associated with every edge $(i, j) \in E$; in this case the graph *G* is called **weighted** and the (weighted) adjacency matrix is a *n* x *n* matrix *W* having $w_{ii} = 0$, if *G* is simple, $w_{ij} \geq 0$ and $w_{ij} = w_{ji}$ for each $i \neq j$ if *G* is undirected. In the case of weighted graphs, the previous definitions, related to degree, path and diameter, are modified in order to take into account weights of the edges rather than their number. In particular, *degree* of the node *i* is the sum of the weights of the edges incident on *i* (out-degree is the sum of the weights of the edges starting from *i*, while in-degree is the sum of the weights of the edges ending to *i*); shortest path between *i* and *j* is the list of adjacent nodes from *i* to *j* with minimal sum of the weights on the correspondent connecting edges; the diameter is the largest shortest path computed as just defined.

A **multigraph** is a graph which is allowed to have multiple edges between a pair of nodes. In some cases these edges are also called ***parallel edges***. According to the choice to assign, or not, a unique identifier to every edge, one of two different notions is used:

- *Edges without own identity* – in this case no identifier is associated to an edge, which is therefore identified solely by the two nodes it connects. In this case, the term multiple/parallel edges implies that the same edge can occur several times between a pair of nodes.
- *Edges with own identity* – in this case a unique identifier is associated to every edge, thus multiple/parallel edges means that several different edges connect a pair of nodes.

In some cases the term **pseudograph** is used as synonymous of multigraph, in other cases **pseudograph** is used to characterize a multigraph with loops.

It is important to highlight that a multigraph is different from a **hypergraph**, where an edge can connect multiple nodes.

Multigraphs are really useful in modelling UTS as many different connections (e.g. bus lines) may link the same pair of locations (e.g. bus stops).

5.1.2 A UTS as a graph: from the physical network to the associated graph model

The main elements of UTS can be easily mapped into elements of a graph, basically nodes and edges. Nodes represent locations of interest on the transportation network, such as towns, bus/rail stops, road intersections, etc. while edges represent connections/links between locations, such as roads, rail lines, bus line sections, etc.

Furthermore, other real-world concepts can be associated to the graph elements: for instance, nodes have the capacity to generate traffic that flows onto the edges (links) of the graph. A relevant concept in the real-world UTS is the “route” that is mapped into a series of connected edges (links) of the graph.

Specific properties of a graph used to model an UTS is that it is both a multigraph (more than an edge can connect the same pair of nodes) and a multi-modal graph, in order to model several mobility modalities (bus, rail, car, etc.).

Figure 6 shows a road network proposing a mapping of cities as nodes and roads as edges of a graph.

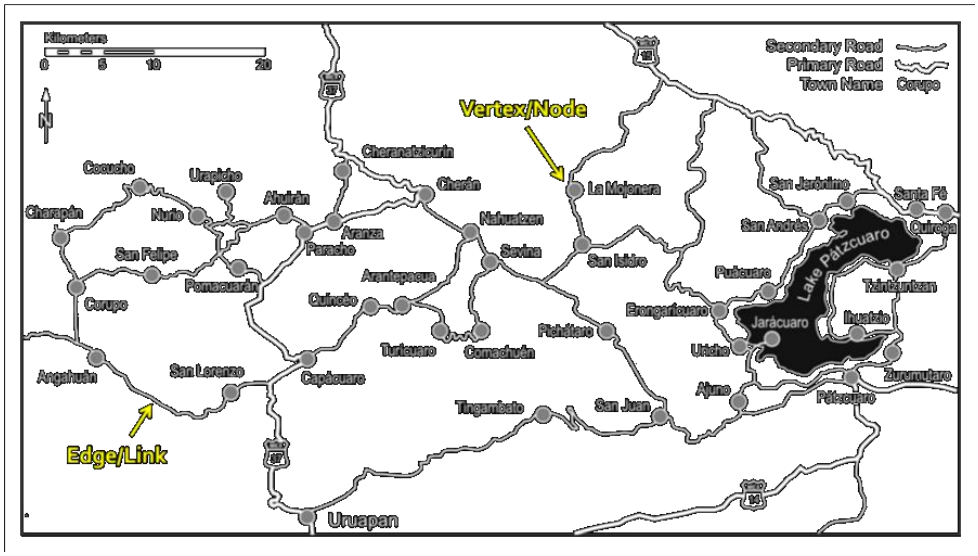


Figure 6 – From a transportation network to a graph: an example

When UTS is mapped as a graph, it is important to manage some specific concepts, such as the “distance”. In particular, two different concepts of distance occur: the real-world distance, which is basically the length of the road to be travelled, and the topological/geometrical distance, which is associated to the edge of the graph model (e.g. number of edges to be travelled or sum of weights, in case of a weighted graph). This difference is summarized in the following Figure 7.

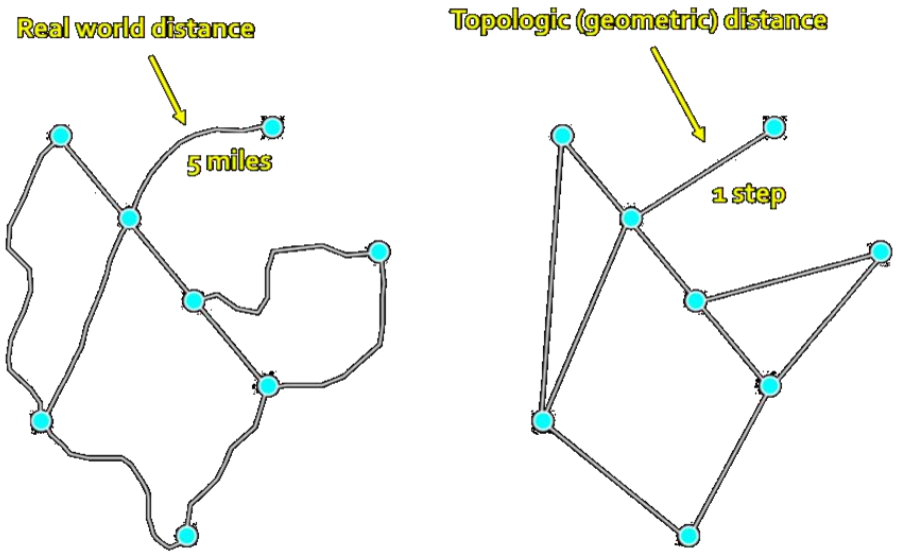


Figure 7 - Real world distance versus topological (geometric) distance

It is important to understand that attributes can also be associated to edges, similarly to nodes. Although real-world distance is the easiest choice as attribute, other possibilities are related to the availability of statistics and/or real-time data, such as traffic volume, travelling times, number of users (e.g. passengers), etc. Each one of these possible attributes can be used to weight edges enabling – through the same analysis – different insights.

5.1.3 Network Analysis: relevant graph-based measures

Networks from different domains share some properties that can be measured by a set of indices, which can take specific ranges of values in correspondence of each specific domain. Social networks and distribution systems, such as water, energy and oil/gas grids, are examples of application domains, different from UTS, with their own properties.

Some properties are more related to nodes and their "relevance" into the network, while others are more focused on the entire connectivity/linkage of the overall structure.

Starting from the nodes, the basic information is related to their degree; the **degree distribution *P(degree)*** of a network gives the relative frequencies of the different node degrees into the network. Real networks often show a skewed node-degree distribution, with most of the nodes having few links and only few nodes being extremely connected. This heavy tailed distribution is known as **Power-law** or **Scale-free** distribution, usually defined as $f(x)=ax^k$.

Contrary to distribution networks, such as energy/power, water and gas/oil, UTS have usually not a planar structure (that is, they can be drawn on a piece of paper without any edges crossing) and, therefore, they usually show a Power-low or Scale-free distribution, similarly to social networks. Another characteristic shared between both UTS and social networks is that they have usually "hubs" (i.e., high degree nodes), a property which is not held in planar networks (i.e. water distribution networks). The presence of hubs usually implies a small diameter in the network - a typical situation in social networks known as *small world phenomenon* (Barabasi, 2003; Backstrom et al., 2012).

As a basic measure of connectivity, the ***average degree (*$degree_{avg}$*)*** can provide immediate information about the organization of the network: it is near to 2 in planar networks (i.e. water distribution networks), and usually higher in social networks. Moreover, this measure is also linked to the ***link-per-node ratio (e)*** that is computed as the number of edges of a graph with respect to the number of its nodes: it is near to 1 for planar networks.

Density (q) is an important measure related to the overall structure of the network, it quantifies how much the nodes of the graph are connected among them. Given a graph *G=(V, E)*, its density is simply computed as the ratio between the number of edges of the graph, *m*=|*E*|, and the overall possible number of connections among the *n*=|*V*| nodes of *G* (i.e., n(n-1)/2 in case of undirected graph):

$$q = \frac{2m}{n(n-1)}$$

The density measure varies between 0 and 1, and when it is 1, *G* is ***complete*** (i.e., any nodes is directly linked to every other node of the graph).

By taking into account **connectivity information based on edges rather than nodes**, measures related to the network organization can be computed based on ***betweenness centrality***. Betweenness centrality can be computed **for each node or edge** (*edge-betweenness*); it can be simply defined as the number of all the shortest paths passing through that node or edge, respectively.

Central point dominance (c_b'), based on betweenness centrality, has been proposed as a measure for characterizing the organization of a network according to its path-related connectivity; in particular it is computed as the mean over the betweenness centrality values of all nodes indexed by the maximum value of betweenness (that is that of the most central node):

$$c_b' = \frac{1}{n-1} \sum_{i=1,..,n} \left(b_{\max} - b_i\right)$$

where b_i is the betweenness centrality of the node *i*, b_{max} is the maximum value of betweennees centrality over all the *n* nodes of the network.

The ***edge betweenness*** is really important because the edges connecting different "sub-graphs" (namely "communities" in a social network) will have high edge betweenness and probably belong to the **edge cut-set**: this is the fundamental idea of the algorithm for **graph clustering** (aka community discovery) proposed by Girvan-Newman (2002). The **min edge cut-set** is the set of edges, with minimum cardinality, to be removed from the graph in order to create a disconnection (i.e. generates new connected components): thus, the min edge cut-set allows for the identification of edges (i.e. links/connections in UTS) which could entail service inefficiencies or interruptions in case of their failure. Other graph clustering algorithms, further than Girvan-Newmann, may be used to identify the minimum edge cut-set, in particular **Spectral Clustering** (Luxburg 2007), which works with eigenvalues and eigenvectors of the (Normalized) Laplacian matrix associated to the graph. Graph clustering will be discussed in detail in the following.

Furthermore, the evaluation of network resilience and vulnerability issues requires extending the analysis in order to include more complex structural properties, such as **cycles**, as an indicator of **network redundancy**. In respect to this, the ***clustering coefficient (c)*** is used to characterise resilience of a network according to loops of length three and is computed as the **number of triangles ($N_{triangles}$)** with respect to the overall number of **possible connected triples ($N_{triples}$)**, where a triple consists of three nodes connected at least by two edges while a triangle consists of three nodes connected exactly by three edges (complete subgraph):

$$c = \frac{3N_{triangles}}{N_{triples}}$$

Final goal of graph clustering (Schaeffer 2007) is the same of traditional clustering approaches: that is partitioning objects into subsets so that objects in a cluster would be more similar than outside the cluster. However, graph clustering strategies, such as Spectral Clustering, solves the problem by taking into account a graph-based structure of the relations (edges) between objects (nodes). The aim is to group nodes of the graph into sub-graphs (clusters) maximizing the sum of the number-of/weights-on the edges within each cluster (intra-cluster similarity) while minimizing the sum of the number-of/weights-on the edges connecting nodes in different clusters (inter-cluster similarity).

The solution of the graph clustering problem can be easily described in the case of **bi-partitioning**. Given two sets of nodes (clusters), C_1 and C_2, the objective is to minimize the edge cut-set:

$$cut(C_1, C_2) = \sum_{x_i \in C_1,\, x_j \in C_2} s_{ij}$$

A *n*-dimensional vector *p* (i.e., *n* is the number of nodes in the graph) is used to represent the association of each node to cluster C_1 or C_2:

$$p_i = \begin{cases} +1 & if\ x_i \in C_1 \\ -1 & if\ x_i \in C_2 \end{cases}$$

The graph clustering problem can be formulated as minimization of the following function *f(p)*:

$$f(p) = \sum_{x_i, x_j \in V} L_{ij} (p_i - p_j)^2 = p^T L\ p$$

where L_{ij} are the entries of the **Laplacian matrix**, the core of Spectral Clustering. Different alternative definitions have been proposed and studied through graph theory; the usually adopted definition is:

$$L = D - A$$

where *A*, in this case, is the **Affinity matrix** of the undirected graph and *D* is the degree matrix, with each entry defined as:

$$d_{ij} = \sum_j a_{ij}\ , i = j$$

$$d_{ij} = 0, i \neq j$$

Affinity is some measure of similarity between nodes and can be also represented as weight of the corresponding edge.

The most important properties of the *L* matrix are:

- it is symmetric and positive semi-definite (it has *n* non-negative, real-valued eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$, irrespectively to their multiplicity);
- its smallest eigenvalue is 0 (where its multiplicity indicates the number of distinct connected components);

Many Spectral Clustering implementations use the **Normalized Laplacian matrix** instead of the basic one; the most common definition for the Normalized Laplacian matrix is the following:

$$L_{norm} = I - D^{-1/2} A D^{-1/2}$$

The combinatorial complexity of minimizing the objective function *f(p)* can be prohibitive for graphs associated to real world networks. However, a simple algebraic solution to the problem was proposed in (Fiedler, 1973): in particular, he used the result of the Rayleigh theorem and identified the **2nd smallest eigenvector of the Laplacian matrix** (usually known as Fiedler vector) as the vector *p,* which provides the **optimal bi-partitioning of the graph**.

This result has permitted to implement **recursive bi-partitioning Spectral Clustering** approaches (Hagen and Kahng, 1992) in order to perform partitioning in $K > 2$ groups. However, this approach requires the computation of matrices and eigenvalues, as well as the use of the Fielder vector, for each sub-graph until the desired number of clusters is reached.

Another possible schema to solve the ***K*-partitioning of a graph** uses a data representation in the – usually low-dimensional – space of relevant eigenvectors (Luxburg, 2007; Ng et al., 2001). The relevant eigenvectors are the first *K* smallest: the *K*-th eigenvalue is the one showing a sufficiently large variation in the **eigengap** that is the difference between two successive eigenvalues in the list of eigenvalues sorted in ascending order.

For example, the *K*-partitioning approach proposed in (Shi and Malik, 2000), consists in selecting the *K* smallest non-zero eigenvalues and performing a traditional *K*-means clustering on the resulting dataset having *n* rows (nodes of the graph) and *K* columns (eigenvectors corresponding to the K smallest eigenvalues). Any other traditional clustering algorithm may be applied in the space spanned by the eigenvectors instead of *K*-means.

Moreover, the number of relevant eigenvalues – thus eigenvectors – must not be necessarily equal to the number *K* of desired cluster, the *l* smallest eigenvalues can be selected according to the variation in the eigengap, while different *K* values can be tested to select the best partitioning.

5.1.4 Network Analysis: general results from other networked infrastructures

According to the almost direct correspondence between a critical networked infrastructure (such as UTS, energy/power grids or water distribution networks) and a graph, complex network analysis can be performed to study this system. A quite recent significant work has been done by Yazdani and Jeffrey (2012) in the field of complex network analysis for Water Distribution Networks (WDN). In their work, a set of four real life networks has been characterized and compared with respect to several graph-based indices and some relevant conclusions have been provided according to the values assumed by these indices for WDNs.

Although the study of Yazdani and Jeffrey was focused on WDNs, some specific graph-based indices defined for these networks could be applied to UTS in order to capture relevant topological and also service related aspects.

The *Meshed-ness coefficient* (r_m), for instance, is used in this paper, but it has been previously proposed in (Buhl et al. 2006) to quantify the cycles and loops in planar graphs:

$$r_m = \begin{cases} \dfrac{m-n+1}{2n-5} & for\ networks\ having\ only\ one\ source \\ \dfrac{m-n}{2n-5} & for\ networks\ having\ multiple\ sources \end{cases}$$

Efficiency, accessibility, and partly resilience, in a network are issues addressed through pathway-based connectivity analysis, in particular the analysis of shortest distance between all pairs of nodes. As in WDN, two basic measures can be used to evaluate accessibility and efficiency in UTS: Euclidean distance between nodes and shortest path length.

When shortest path length in a WDN is considered, it is easy to understand that the crucial operational goal of this kind of infrastructure is to maintain connectivity between water sources and consumption points, hopefully in the shortest and most efficient way. At the same time, this consideration can also be used in order to analyse the capacity of UTS to maintain connectivity between any possible source (origin) and destination. Thus, the measure of efficiency proposed in (Yazdani and Jeffrey, 2012) for WDN, measured as the connectivity between any source (root node of the path) and consumption points (other nodes of the network) rather than connectivity between all the possible pairs of nodes, could be adapted and then adopted also in UTS. In more detail, a specific connectivity factor, namely *network's route factor (g)*, proposed in in (Black, 2003) could be used. It has been defined as:

$$g = \frac{1}{(n-1)} \sum_{i=1}^{n-1} \frac{\varepsilon_{s,i}}{\delta_{s,i}}$$

where $\varepsilon_{s,i}$ is the Euclidean distance along the edges in the path connecting source *s* to destination *i*, and $\delta_{s,i}$ is direct Euclidean distance between *s* and *i*. In case of a network with a star-graph structure the route factor is equal to 1 (smallest value); greater values of the route factor indicate a greater deviation from the optimal structure and, consequently, larger costs to manage and operate the networked infrastructure.

5.1.5 Topological and system-based analysis

As previously mentioned, most of the literature is focused on vulnerability analysis, based on the study of the physical (static) infrastructure, rather than resilience. Vulnerability analysis of UTS – addressed through network analysis – has been gaining a renewed interest in the last decades, with mostly two distinct research lines with limited interaction:

- **topological vulnerability analysis** of transport networks
- **system-based vulnerability analysis** of transport networks

In **topological vulnerability analysis** approaches, a real transport network is represented as a graph consisting of a set of nodes (or vertices) and a set of edges (or links). The graph could be undirected or directed, as well as unweighted or weighted. Nodes and edges in the graph have different counterparts in the real transportation network (e.g. nodes can be stations or crossroads, links are usually physical connections such as roads or railways).

The **system-based vulnerability analysis** approaches also integrate demand and supply data/models to be used in the analysis. The transport network is anyway modelled as a graph, but edges are usually weighted with weights corresponding to actual lengths, travel times, costs or a combination of these in the form of generalized costs. In addition, the interaction between demand and supply is simulated through modelling and/or software simulation (e.g. agent-based).

Although the main drawback of the topological approaches is that they are too simplistic to define actual policy actions in UTS, they usually offer some relevant benefits, such as:

- they do not require a huge amount of data – typically the only information about the interconnections is needed to create the graph associated to the UTS infrastructure;
- they use elegant, rigorous and mathematically sound theory;
- they provide fundamental insights about the structural weakness of a transport network.

On the other hand, system-based vulnerability approaches aim to overcome some limitations of topological ones, usually requiring more data such as travel demand and supply, accurate behavioural models to simulate travellers' responses to disruptions and further predict repercussions on other travellers. Therefore, system-based vulnerability approaches are less uniform than topological studies, depending on the availability of data and modelling strategies.

Already in (Watling & Balijepalli 2012), even if it is a vulnerability analysis study, data about demand growth has been considered to assess road network vulnerability in order to separate the effect of demand growth from the mean, variance and skewness of travel times to identify the most vulnerable links of a network.

Other important data sources which can be used by transport authorities in vulnerability/resilience analysis are related to "opportunistic sensors" like taxi GPS systems that are becoming increasingly available (Jenelius & Koutsopoulos, 2013).

More important, system-based vulnerability approaches allow for a better definition of impact measures with respect to consequences of disruptive events. These measures may range from very simple ones (e.g. increase in travel time, increase in travel cost, cancelled travel options) over more general measures of accessibility to comprehensive economic measures of consumer surplus or financial impacts.

A really interesting study is proposed in (Dehghani et al., 2014), comparing topological-based and system-based measures. A hypothetical road network was defined, where each link has a given length and can be closed independently of the other links with a certain probability. An OD matrix for the travel demand between the nodes in the network is randomly generated.

Recently, (Reggiani 2013) the relationship between network resilience and transport security has been investigated, focusing on the connection between the topological structure and – different interpretations – of resilience/vulnerability. More recently (Reggiani et al. (2015) efforts have been focusing on how resilience and vulnerability can be framed, interpreted and measured, and their relationship with connectivity of the associated graph.

5.1.6 Types of events and associated modifications in the graph-based model

With respect to UTS, disruptions may be classified according to two dimensions: **causes** and **impacts**. Causes of disruptions can be *internal* or *external*, as well as referred to *accidental events* and *intentional interferences*. This distinction is usually important with respect to the separation between safety and security in transport.

Impacts of disruptions can be of different types. Accidents, infrastructure collapses and attacks may lead to direct/indirect injuries and fatalities. Common disruptions, such as a road link blocked, a rail service interruption, a strike, etc., have an impact with lower severity. These events will increase the travel time for passengers and lead to cancelled trips, generating social and/or economic costs (taking also into account the – possibly relevant – costs for restoring the service level and for repairing or rebuilding the infrastructure).

It is therefore important to define possible mapping of events into graph modifications, in particular to model causes, while network analysis – and usage of real-time data and/or transport simulation – allows for evaluating impacts, in particular with respect to reduction of quality of service and allocation of resources.

In the following, mapping of some relevant real-world events towards graph modifications is proposed:

- **Event: closure of a station/stop**
  This event refers to a node in the graph, however two alternative mapping options are available according to the situation occurring in the real-life setting:
  a) *Access to the station/stop is disabled but transport line/route is not interrupted*. In this case the node is removed from the graph but all the paths passing through it are still maintained. However, since it is no more possible to change transportation line/route at that station, overall connectivity and paths in the graph will change.
  b) *Transportation lines passing through the station are interrupted*. This case requires not only to remove the node from the graph (i.e. access to station/stop is disabled) but also to remove all the paths passing from that node, in order to model the interruption of the transport line(s)/route(s) passing through that station. Thus, this event has a greater impact than the previous one in terms of service level.
- **Event: interruption of a connection**
  This event refers to an edge in the graph and modelling is similar to the second setting of the "closure of a station/stop event". In particular, all the paths passing from the destination node of the edge, and belonging to the same transport line/route on the edges entering into the source node of the edge, are removed from the graph, along with the affected edge. In this case, stations (i.e. source and destination nodes of the edge) do not require to be closed, avoiding impact on the other lines/routs passing through them.
- **Event: reducing connection capacity / increasing traveling time**
  This event refers to an edge on the graph; the big difference with the previous case is related to modelling the real-world event through the weights on the edge. This case requires working with attributes of the edges rather than with the edges themselves, affecting the "flow" on the graph (i.e. transportation service) rather than its connectivity

The previous macro-categories of events – and relative mapping alternatives – are quite general and allow for modelling more complex real-world events through the combination of modelling solutions as well their extensions to more than one elements at the same time. For instance, during a flooding event, a number of stations/stops can be inaccessible, as some connections are interrupted and flow is strongly reduced on other links.

5.2 Network science for multi-layer resilience

Due to the complexity of UTS, and to the need to model, through its associated graph, multi-modalities, events and changes in mobility flows, a multi-layer approach is required in order to address all the relevant issues of supporting a more sustainable resilience management.

5.2.1 Analysing physical, service and cognitive levels, individually

As previously mentioned in this deliverable, most of the research, in particular on network analysis applied to UTS, has been focused on vulnerability analysis, which strictly involves the study of connectivity of the overall transportation infrastructure (i.e. the physical level). Only recently, approaches of system-based analysis have been proposed, integrating mobility data, in particular demand and flows, in order to include in the loop also information about the level of service and possible impacts on them due to disruptive events.

On the other hand, the service level has been investigated to model and study mobility behaviour of the UTS' users, also through the exploitation of new sources of data such as GPS data from taxis, smartphones and social networks. The well-known origin-destination matrices are usually adopted in order to represent and model both demand and flows – therefore mobility choices/behaviours – of public transport passengers as well as citizens walking in the city. Combining temporal information together with the spatial one allows for a further and more detailed analysis and inference/characterization of such behaviour, enabling the identification of frequent, periodic and typical patterns, as well as sudden/anomalous changes in the patterns (i.e., service level analysis).

Finally, an example of analysis of the cognitive level is reported in (D'Lima & Medda, 2015), where a new resilience measure is proposed according to the speed with which the passenger count time series return to normal condition, taking this information as an indicator of how quickly the underground transport system is able to recover from the shock and thereafter resume normal operations. The study has considered the London Underground as a use case with the aim to examine the resilience of the system to shocks, such as delays or disruptions in the underground service. For example, if the Victoria line is running with severe delays, there is a sharp fall in passenger counts on the Victoria line and, consequently, there will be an increase in passenger counts on lines which run parallel or North–South through London, perhaps causing minor delays on these lines too. The passenger counts will eventually return to normal levels once the shock dissipates. In particular, a mean-reverse approach is proposed to model data and estimate the resilience of the underground line.

Similarly, in (Khademi et al. 2015) a comprehensive approach to analyse vulnerability/resilience of a transport system with respect to a catastrophic event is proposed. The fundamental finding of this study is the need to analyse changes in travel demand and behaviour in the response phase of a disaste,r when emergency trips have to be prioritized and many roads are impassable.

It easy to understand that demand and flow data are the key information that enable a deeper study and analysis at both physical and cognitive level. The latter requires demand data (e.g. passenger counts) to be analysed in order to estimate response of travellers with respect to minor (e.g. delays and temporary interruptions) as well as major (catastrophic event) disruptions. The former can exploit the information value provided by demand and flow data in order to build a more representative graph model of the UTS and, therefore, perform more comprehensive analysis. However, the integration of all the three levels is still a challenging research field.

The integration of two or three of the above levels can enable the implementation of relevant decision support functionalities, offering different benefits according to the available data (e.g. descriptive statistics, forecasts or online real data).

One of the decision support functionalities is related to the problem of "bridging" and, in particular, its optimization (Kepaptsoglou & Karlaftis, 2009). In the case of bus bridging some disruptions are "expected" (i.e. scheduled, mainly for maintenance), but the task is still valid in case of unexpected disruptions. This specific task is generally framed as an optimisation problem, having as objective the maximisation of the "passenger welfare", as a function of route selection, configuration and bus assignment, and being subject to demand patterns, resource availability, and route and service constraints.

Therefore, data about the UTS network (physical layer), demand estimation and resource (service level) and some measure of the "passenger welfare", which could be just some kind of estimation of the cognitive layer, is needed. Bridging is usually formalised as an optimisation problem (i.e. a non-linear constrained optimisation problem, in the paper) and a number of mathematical/computational approaches can be used to solve it, such as genetic algorithms used in the paper in order to select optimal routes and allocate vehicles. The pilot considered in the study is Athens, with some indications about available data and bus availability scenario. A limitation of the proposed approach is the limited attention to demand characterization and forecasting and in particular to the modelling of uncertainty.

A more structured optimization approach is suggested in (Jin et al., 2015). The focus is on the optimisation of temporary services and allocation of buses to parts of the network; the same approach also applies to scheduled disruptions. The optimization problem is defined in three steps:

- network design, given the demand,
- vehicles allocation (line planning),
- and timetabling.

Also in this case, the strongest integration is related to the physical and service levels, with timetabling defined in order satisfy users' needs. The paper considers 2 disruption cases; computational results show that using additional "non-trivial" routes results in a significant decrease of the average travel delay, while an analysis related to the sensitivity of travel delay to the number of buses is also suggested.

Figure 8 summarizes the key concepts of the bus bridging problem and offers a possible formulation.

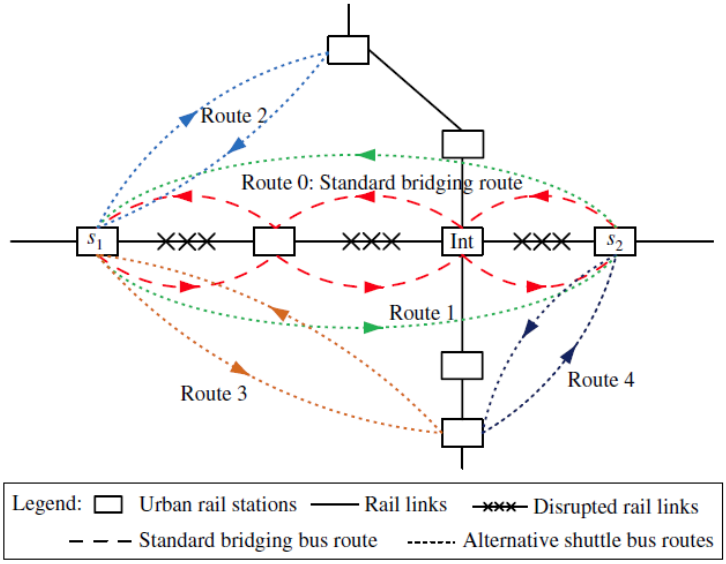


Figure 8 - Optimization of the bus bridging – a possible formulation [source (Jin et al., 2015)]

To solve the problem, a specific model of the transportation network has been proposed, as reported in Figure 9, consisting of rail stations, rail lines, bus stations and bus lines.

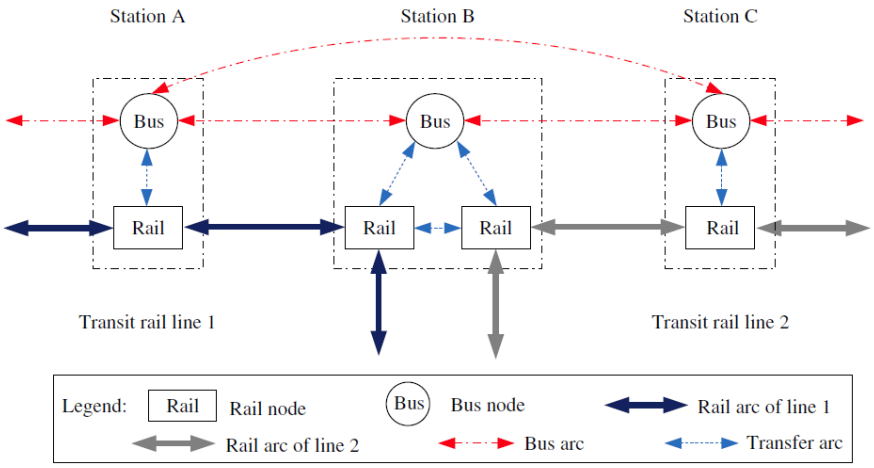


Figure 9 - A network representation for the bus bridging optimization problem [source (Jin et al., 2015)]

Bus bridging is a complex optimization problem and it is easy to prove that non-trivial solutions often perform better in terms of objective function. In figure 10 an example is reported.

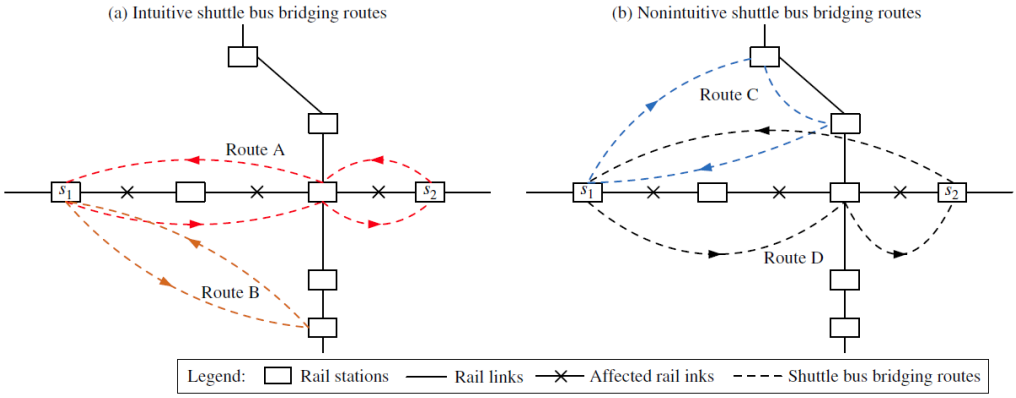


Figure 10 - finding a set of candidate routes for bus bridging [source (Jin et al., 2015)]

In the quoted paper the problem is addressed through hard optimization with column generation, resulting in a complex model considering – and modelling – both space and time. A representation is provided in figure 11.

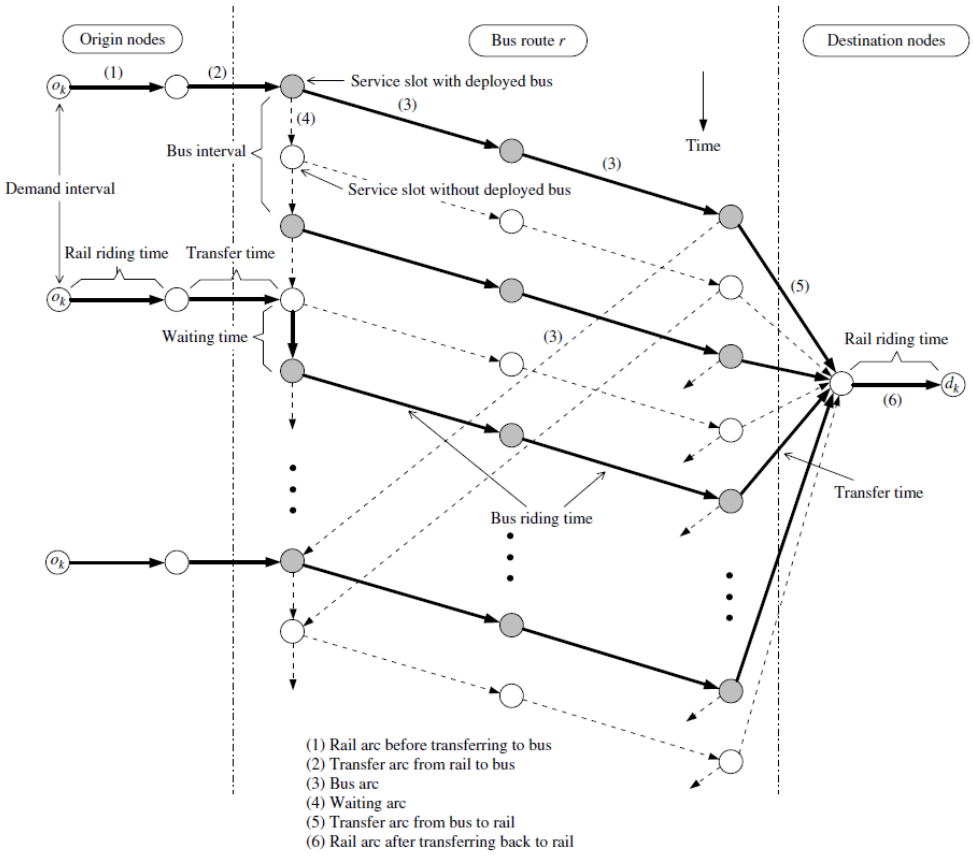


Figure 11 - time space network proposed in (Jin et al., 2015)]

5.2.2 Integrating physical, service and cognitive levels

A tighter integration among the three levels has been proposed in (Wang et al., 2015), where the problem of bus bridging disruption in rail services with frustrated and impatient passengers is addressed. The aim of the proposed approach has been to select stations to bridge by bus bridging according to track crossovers enabling trains to turn back to the incoming direction. The target network was Melbourne's, which in first half of 2011 experienced 15000 disruptions, 47 of which required bus bridging.

The paper highlights two specific points:

- Demand modelling and forecasting (passengers counts) depending on disruption. In this case there is no demand equilibrium model driven by Origin-Destination matrix: random factors are dominant both in demand, bus availability and travel time.
- Balking and reneging are factors which redefine actual bus demand along with car sharing and taxi hailing.

In order to derive a solution for the two previous points, a learning procedure on data available from previous and similar events is proposed. Bus bridging can now be modelled as a bulk queuing system with balking and reneging. The model is probabilistic and no analytical solutions are available for general configuration and probability distribution, in particular empirically derived. Thus, Monte Carlo simulation can be used to address the stochastic nature of the problem.

Furthermore, a resilience framework is proposed in (Vugrin et al., 2010) to simultaneously consider restoration of system performance and the resource expenditures required. Two key quantities are computed: Systemic Impact

(SI), which is the cumulative impact of decreased system performance following a disruption, and Total Recovery Effort (TRE), which is the cumulative resources expended in recovery activities. As illustrated in the Figure 12, recovery decisions and actions affect both of these quantities.

The proposition is to integrate these two indices in a composite resilience measure:

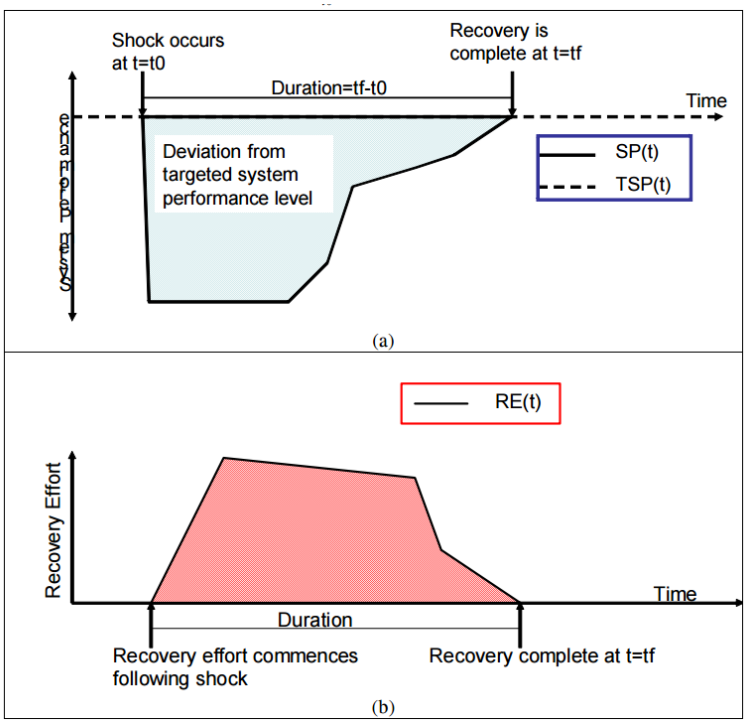


Figure 12 - An example of Systemic Impact and Total Recovery Effort (shaded areas under the curves) [source (Vugrin et al. 2010)]

In any particular application, SI itself may be composed by several different measures of system performance (with relative weights). The important contribution is related to consider system dynamics and solve an optimization problem formulated as a Multimode Resource Constrained Scheduling Problem (MRCSP) which includes also duration and costs of maintenance/repair activities

With respect to disaster/emergency management, a relevant work is (Wilson et al., 2013), where the problem of assigning ambulances to routes is addressed. The main steps proposed are:

- allocation of tasks to responders;
- ordering of tasks for each responder;
- allocation of each casualty.

Prediction and simulation of travel times is performed in several disruption scenarios modulated by a parameter ρ which is associated with the severity of the disruption and leads to an increase in travel times.

5.3 Modelling cascading effects through network dynamics

Finally, many disruptions in transport systems can propagate over the network, showing cascading effects on other inter-depending systems.

Methodological development addressing this specific topic is gaining an increasing interest for future research. As examples, some studies already focused on the geographically extended effects (Jenelius and Mattsson, 2012)

and interrelationships within systems, such as track-based systems (Johansson et al., 2011; Zhang et al., 2014) and road systems (Hémond and Robert, 2010; Hsieh and Feng, 2014).

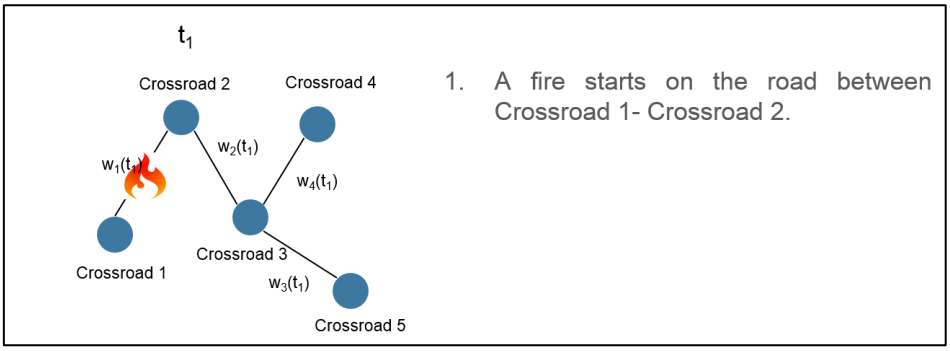
The key element of cascading disasters/effects is time: as time passes by, more locations or connections of the UTS – which are nodes and edges of the associated graph – can be affected consecutively as well as change their own condition.

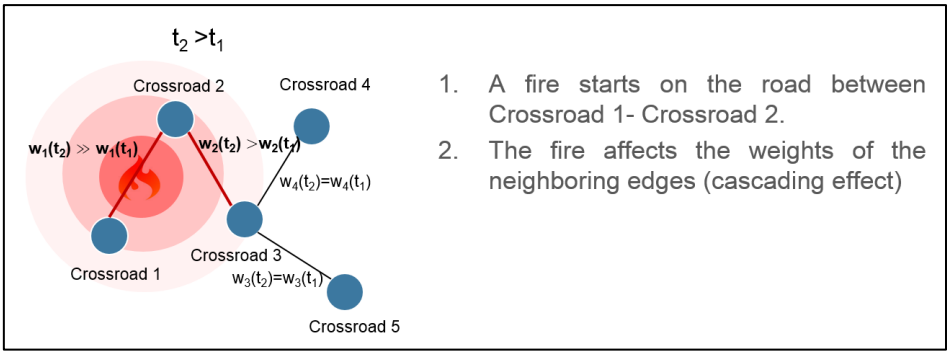
It is important to differentiate among cascading disasters and cascading effects. While cascading disasters are characterized by an event which occurs at a specific time and consequent effects experienced by the system (e.g. flood leads to problem in the electric network), the cascading effects are characterized by an event with its own evolution over time, implying an evolution also of its effect (e.g. fire spreading out)

Two types of analysis about cascading effects in UTS are possible: **environmental simulation** and **crowd simulation**. The first one allows for modelling how a disaster affects the environment over space and time (i.e. spreading of the disruptive event, such as fire) in order to understand which could be the locations (nodes) and connections (edges) at higher risk and how the evolution of the environment may affect, over time, the overall infrastructure and service. The second possible analysis, that is crowd simulation, takes into account demand, flow and changes of both these two items depending on how UTS users respond to the event. This allows for modelling how crowds of people will move, without guidance, in hazardous situations

The two different analyses require two different types of graph modelling of the UTS, in particular with respect to the weight to assign on the edges. In the first case (i.e. environmental simulation) edges are weighted according to the (increased) time travel due to the hazard. In the second case (i.e. crowd simulation) edges are weighted according to the crowd movement.

The Figure 13 and Figure 14 provide simple examples for environmental and crowd simulation, respectively.





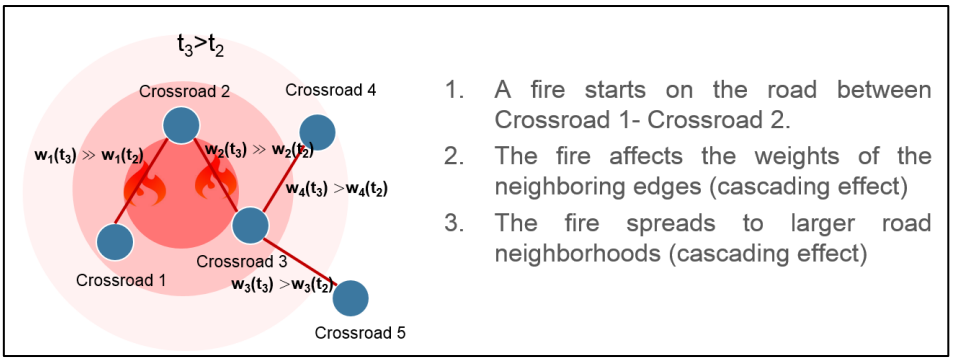


Figure 13 - an example of cascading effect in environmental simulation

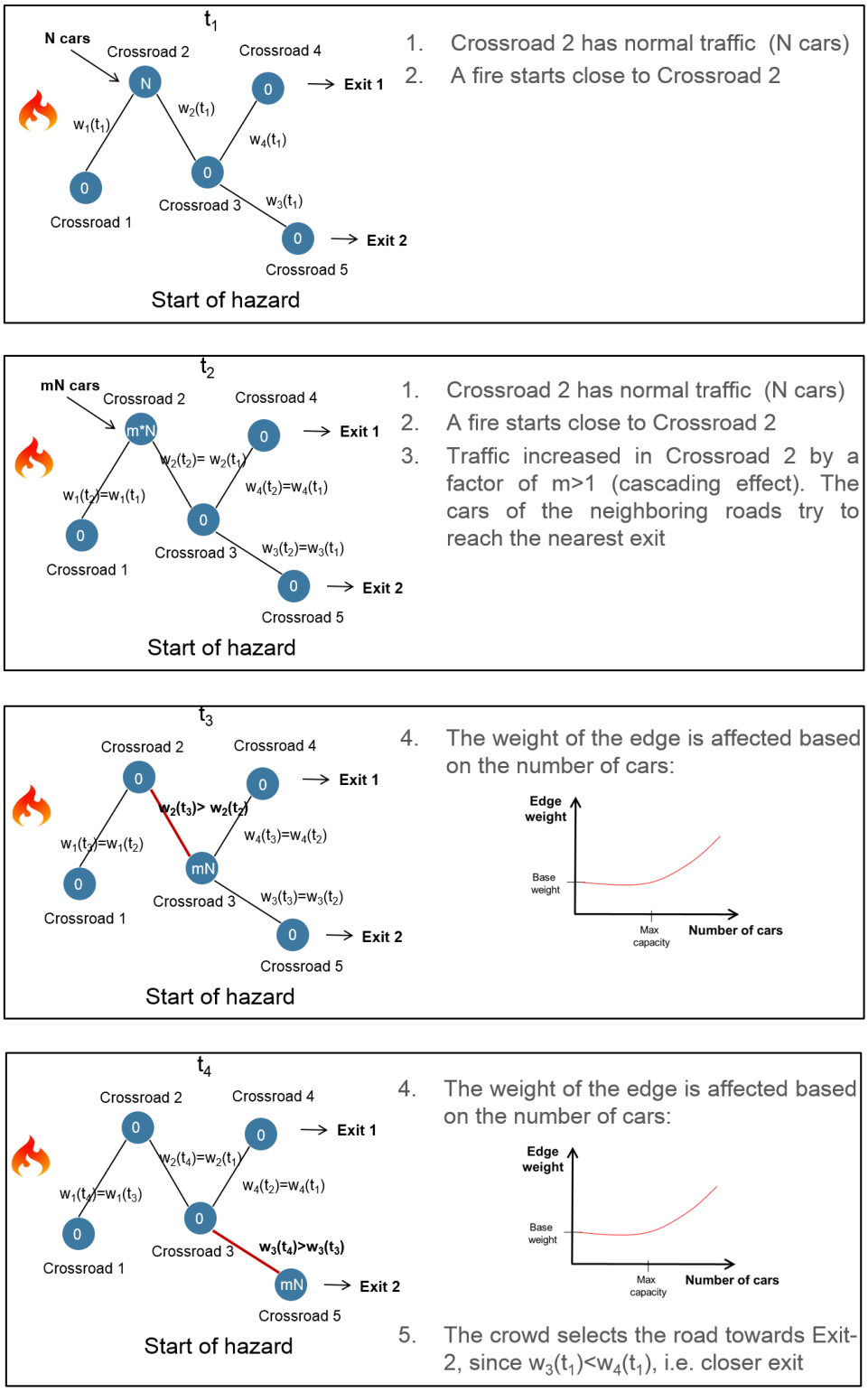


Figure 14 - an example of cascading effect in crowd simulation

Furthermore, as already reported in (Zimmermann, 2004; Zimmermann and Restepo, 2006) resilience of the WDN is critical also with respect to the possibility to produce cascading failures on other networked infrastructures. More specifically, experimental results proved that when a failure occurs in a WDN the damage is propagated to other infrastructures (i.e., roadways, sewer lines and sewage treatment, electric power distribution, gas mains,

telecommunications lines) with greater intensity (3.4 times). Therefore, it is important to take into account, among the possible disruptive events, those coming from other infrastructures physically interconnected to the UTS, as well as "linked" by the possibility that a disruption in these infrastructures may be propagated to the UTS.

Network analysis can be in particular applied in order to estimate how failures might propagate and evaluate the impacts on the level of service. Two indices have been proposed in the case of road network (Zou et al., 2013), namely the "network average Efficiency" (E) and the relative size of the Giant Connected Component (GCC), defined as follows:

Network average efficiency (E)

$$E = \frac{1}{n(n-1)} \sum_{\substack{i,j \\ i \neq j}} \frac{1}{d_{ij}}$$

Where d_{ij} is the path length between the two nodes *i* and *j*.

Relative size of the Giant Connected Component (S)

$$S = \frac{N'}{N}$$

Where *N'* and *N* are, respectively, the number of nodes in the Giant Connected Components after and before the failure.

In order to evaluate, dynamically, the propagation of a failure in the graph associated to the UTS the algorithm synthetized in the following Figure 15 has been implemented.

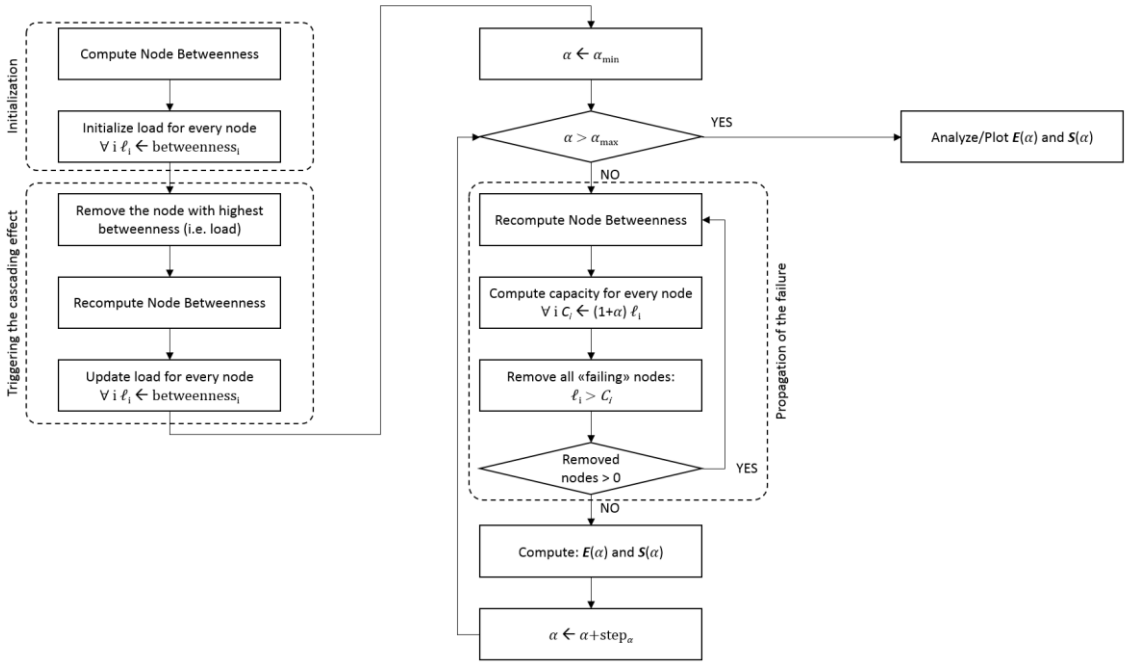


Figure 15 – Algorithm to simulate and evaluate the propagation of a failure in the graph associated to a UTS. The cascading effect is triggered through the removal of the node with the highest betweenness in the network. The analysis is performed varying the value of the parameter α, which is used to set the capacity of each node. Nodes with capacity lower then current load (i.e. betweenness) are removed from the graph and load of each node is updated. The process continues until no more nodes are removed.

5.4 Weather severity monitoring and associated flood hazard

In order to facilitate the resilience strategies of the RESOLUTE project and in order to enhance the capabilities range of the Multi-Risk Analysis module, the need arises to incorporate useful information about the environmental parameters that can possibly have a critical effect. The output of this analysis will provide useful input to the Collaborative Resilience Assessment and Management Support System (CRAMSS). Namely, when combined with susceptibility flooding maps (provided by UNIFI), within the CRAMSS system, new impedance values for the corresponding roads and road segments will be estimated during the evacuation process.

It should be noted, that the current module does not directly affect network topologies, but when combined with the aforementioned susceptibility flooding maps, it can also be applied on transport networks. Further details about the combination of the maps and the current weather-related risk analysis module can be found in the deliverable "D5.3 – CRAMSS Application".

In this respect, a related inference module accessing and manipulating weather measurements and their predicted values, has the capability to indicate the severity of upcoming phenomena, for example rainfall, that in turn can influence associated hazards, like flooding. This information combined with other topological and vulnerability information can demonstrably enrich the decision making process of the system.

In the weather domain, data come by in the unlabelled form, as meteorological measurements from weather stations are readily available but the hazards that they are associated with or other derived phenomena, are not. For the latter to be possible, a domain expert would need to be employed on annotating the data gathered. Even in that case, that would not account for the regional differences, as these phenomena are not universally described. It follows that an automated system that can read and train itself on presented unlabelled data, and afterwards extrapolate a risk factor based on newly acquired instances, is highly advantageous.

In past studies attempting to model flood hazards, (Kalayathankal et al., 2010) use a fuzzy soft set theory approach for a flood warning system. They collect five meteorological parameters over sixteen years and two geographical parameters. (Alfiery et al., 2015) propose a novel early warning system for heavy precipitation events in Europe, aimed at identifying forecasts of extreme rainfall accumulations over short durations. Their system is based on the recently developed European Precipitation Index based on simulated Climatology (EPIC). (Otsuka et al., 2014) implemented a low-cost environmental information collection system for predicting localized abnormal weather, by comparing the values of the observation nodes of a sensor network. (Lang et al., 2008) presented a novel machine learning method for automated condition monitoring, Neural Clouds. It provides a confidence measure for the classification of the complex system conditions. The presented adaptive algorithm requires only the data that correspond to the normal system conditions, which are typically available.

The Neural Clouds concept is an application of neuro-fuzzy methods, and an attempt to make the expert condition monitoring system more intelligent and able to face real world problems, as the flood hazard we are attempting to estimate. The basic idea behind using one-side classification in the field of condition monitoring and fault analysis is that the data that can be collected usually correspond to the normal conditions of the complex system in question.

Every instance of the acquired data from a real phenomenon could be considered as a point in n-dimensional space, where n corresponds to the number of different parameters or features of the system under consideration. First, we normalize and cluster the data by using the K-Means algorithm. We use Gaussian bells to encapsulate the data around the clusters. Next, the normalization and summation of the mentioned Gaussian bells is performed, in order to obtain a hyper-surface in the system variables space that will encapsulate all the data.

After all centroids have been extracted from the input data, the data should be encapsulated within the hypersurface formed from the probability distributions dispersed around each centroid. For this purpose, Gaussian distributions are used:

$$R_i(x) = e^{-\frac{|x-m_i|^2}{2\sigma^2}}$$

Where m_i is the center (or mean) of the Gaussian bell and σ is the width (or standard deviation) of the Gaussian bell. By adding all the Gaussian bells, we obtain the encapsulating surface. The summation may be more than unity; therefore, normalization similar to the partitioning-to-one known for radial-basis function networks is performed. To avoid encapsulating outliers, an additional bias factor g_0 is introduced, corresponding to the first percentile of all the sums of Gaussian bells.

The normalized data encapsulation is defined as:

$$P_c(x) = \frac{\sum_{i=1}^{n} R_i(x)}{\sum_{i=1}^{n} R_i(x) + g_0}$$

This constructed mechanism could be described in a form of Radial Basis Function network, as it is shown in Figure 16. After such a network is manually trained on the measured data, it could be used for the confidence level estimation of the new measurements, with P_C standing for the degree that the new measurement instance represents normal, or similar to what historically has been modelled as normal, conditions.

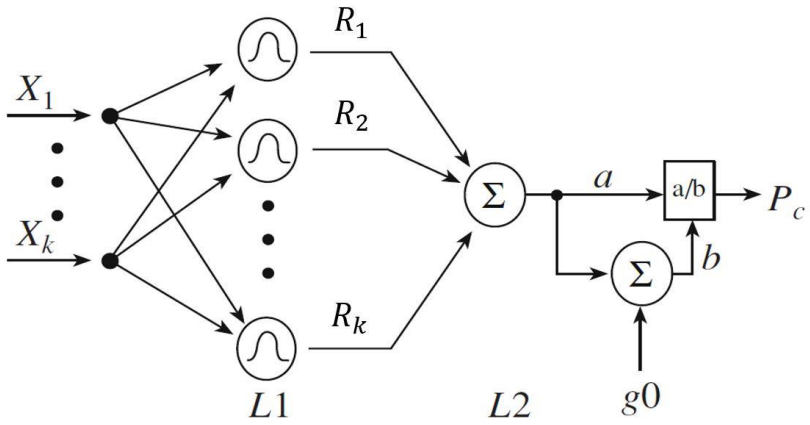


Figure 16: Radial Basis Function network representation. L1 and L2 are Layer one and Layer two respectively. $X_{1\ldots k}$ are input patterns and P_C is a confidence value for the concrete pattern $X_{1\ldots k}$

The output vector, generated by the NC algorithm, could otherwise be described as a confidence value between 0 and 1. The failure probability of the system, when it is expected to operate under unprecedented conditions, is calculated according to the following expression:

$$P_f = |1 - P_C|$$

where P_f is failure probability and P_C is the confidence level.

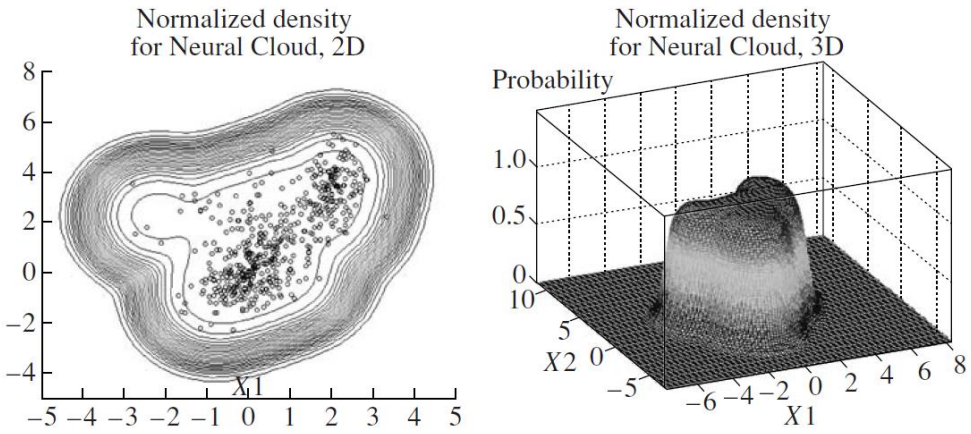


Figure 17: Visualization of Neural Clouds, depicting confidence levels with 2D contour line plots on the left and normalized density 3D plots on the right.

Monitoring extreme precipitation values is integrated by querying an online weather forecast API, which is capable of returning both historical and predicted values. The available granularity is at city level. We tested the Neural Cloud framework by gathering precipitation measurements on a number of locations, namely Florence and Athens. Historical data were gathered at the range of a year (3/2016 – 3/2017) as to capture the seasonal variance in the measurements.

Afterwards, by querying the same web based weather API for future predicted values, we classify them by using the trained Neural Cloud model, and a hazard factor for rainfall is returned. Table 1 & Table 2 hold the results from testing the model on past data from Florence and Athens respectively. The dates were chosen so that they contain values that were amongst the most extreme recorded within the year. Here, the hazard factor P_f attempting to indicate the flood hazard on selected 3-hour intervals is shown, based on a single feature Neural Cloud constructed by the recorded value of 3-hour aggregated precipitation.

Table 1: Hazard Factor and respective precipitation measurement in Florence

05-06/02/2017	00:00	06:00	12:00	18:00	00:00	06:00	12:00	18:00
Hazard Factor	0.09	0.09	1.00	0.44	1.00	0.16	0.35	0.09
Measurement	0.0	0.2	6.4	2.3	7.6	1.4	2.1	0.1

Table 2: Hazard Factor and respective precipitation measurement in Athens

27-28/11/2016	00:00	06:00	12:00	18:00	00:00	06:00	12:00	18:00
Hazard Factor	1.00	0.09	0.15	1.00	0.09	0.13	0.09	0.35
Measurement	7	0.3	1.3	6.7	0.3	1.1	0.4	2.1

We notice that the factor does not zero out when the respective measurement does. That is because the statistical representation of rainfall is approximate to a gamma distribution and we model this distribution in our case, with a half-normal distribution; that is using only one side of the Gaussian. This way the mean of the Gaussian falls on the zero of the gamma, and a normal and uneventful measurement can be described as slightly deviating.

As a side note, the system architecture is designed in a way that it can also accommodate labelled data. The process remains largely the same, this time though the target function is known (the hazard factor) for each train instance, and thus it is possible to train a simple RBF network instead of a Neural Cloud.

6 APPLICATION FRAMEWORK

This chapter aims at presenting software tools which have been used, expanded and integrated with the aim to provide algorithms and models for user profiling and network analysis.

The main contribution of the application framework is to offer an integration of the most update analytical functionalities, related to the two mentioned topics, in order to support a more effective operationalization of the resilience management.

6.1 Software components for user identity and profile management

The algorithms presented previously, in chapter 4.1 of this deliverable are developed by using any specific data analysis library but by implementing the logic presented so far.

With respect to the analysis of UTS behaviours, the following suites have been considered and integrated:

R is a language and environment for statistical computing and graphics[7] . It is a GNU project and provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques. The most important property of R is its highly extendibility (through "packages"). R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

From a user (data scientist) viewpoint, R is an integrated suite of software facilities for data manipulation, analysis and graphical display. It includes an effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis and display either on-screen or on hardcopy, and a well-developed, simple and effective programming language, which includes conditionals, loops, user-defined recursive functions and input and output facilities.

In particular the package *skmeans* (spherical k-means) has been used to implement a task of data analysis aimed at inferring typical mobility behaviour of travellers from the passenger counts time series of the London Underground. Skmeans, in particular offers the possibility to clustering time series data by using cosine similarity, which allows managing similarity in time when comparing tow time series. This approach has been also already applied by the authors in a previous project with the aim to characterize – and then forecast – hourly water consumptions at urban and individual levels (i.e. in the second case smart metering data have been used) (Candelieri et al., 2014b, Candelieri et al. 2015b).

[7] https://www.r-project.org/

Weka is open source software[8] issued under the GNU General Public License, and provides a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. It does not provide cosine similarity to cluster time series data – contrary to skmeans, in R – but offers a large set of algorithms which are interesting and potentially useful to analyse the data collected into the RESOLUTE platform.

6.2 Software components for multi-risk and network analysis model

GraphStream[9] is a Java library for the modelling and analysis of dynamic graphs. Graphs can be generated, imported, exported, analysed (i.e. measures can be computed), and visualized, even dynamically. The goal of GraphStream is to provide a way to represent graphs and work on it. To achieve this goal, it offers several graph classes to model directed and undirected graphs, 1-graphs or *p*-graphs (a.k.a. multigraphs, that are graphs that can have several edges between two nodes).

GraphStream allows storing any kind of data attribute on the graph elements: numbers, strings, or any object. In addition, GraphStream provides a way to handle the **graph evolution in time**. This means handling the way nodes and edges are added and removed, and the way data attributes may appear, disappear and evolve.

GraphStream is also able to deal with GIS data and it is therefore perfectly suited for analysing UTS. Figure 18 shows a road network modelled and visualized through GraphStream

[8] http://www.cs.waikato.ac.nz/ml/weka/

[9] http://graphstream-project.org/

Figure 18 - an example of network modelled through GraphStream and using GIS data

Attributes can be associated and valued for every node and edge. According to the value of a given attribute is possible to colour, in a different way, graph elements. Figure 19 presents an example where bridges and a tunnel are highlighted in yellow and red, respectively,

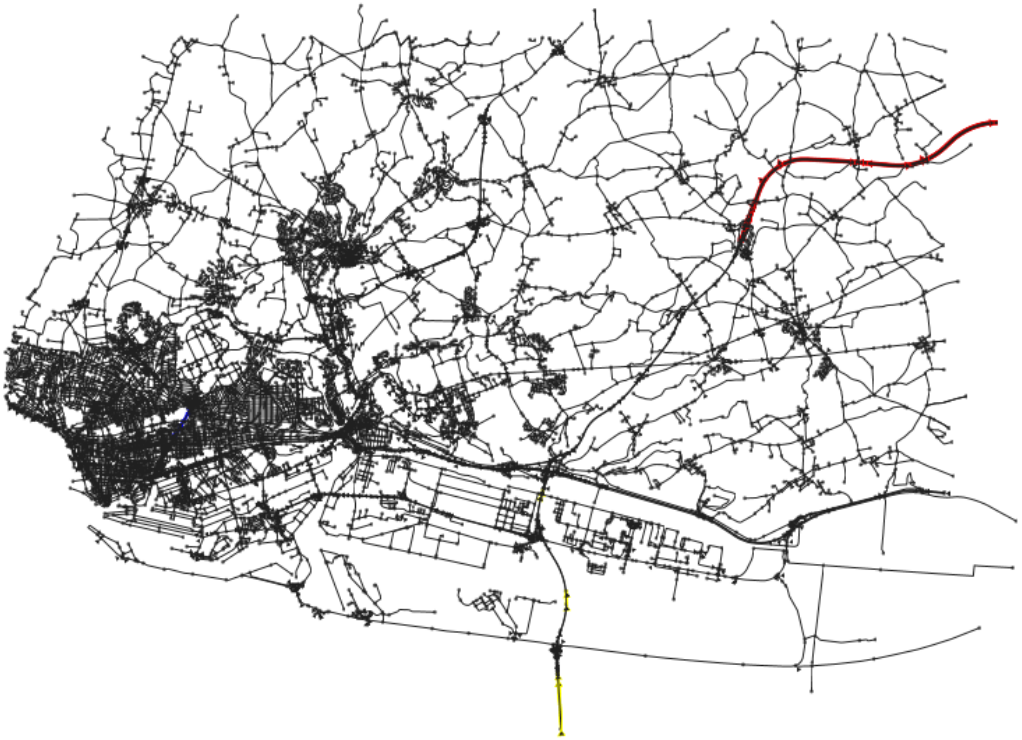


Figure 19 - some edges highlighted with a different color for tunnel and bridges

Also zooming is possible, both dynamically and programmatically, allowing for a more accurate visualization and identification of graph elements of interest, such as in Figure 20.

Figure 20 - an example of zooming-in aimed at focusing on a specific network element (i.e. blue link)

In Figure 21, instead, the result of online modifications on edges colour is shown, according to traffic volume on the associated road segments.

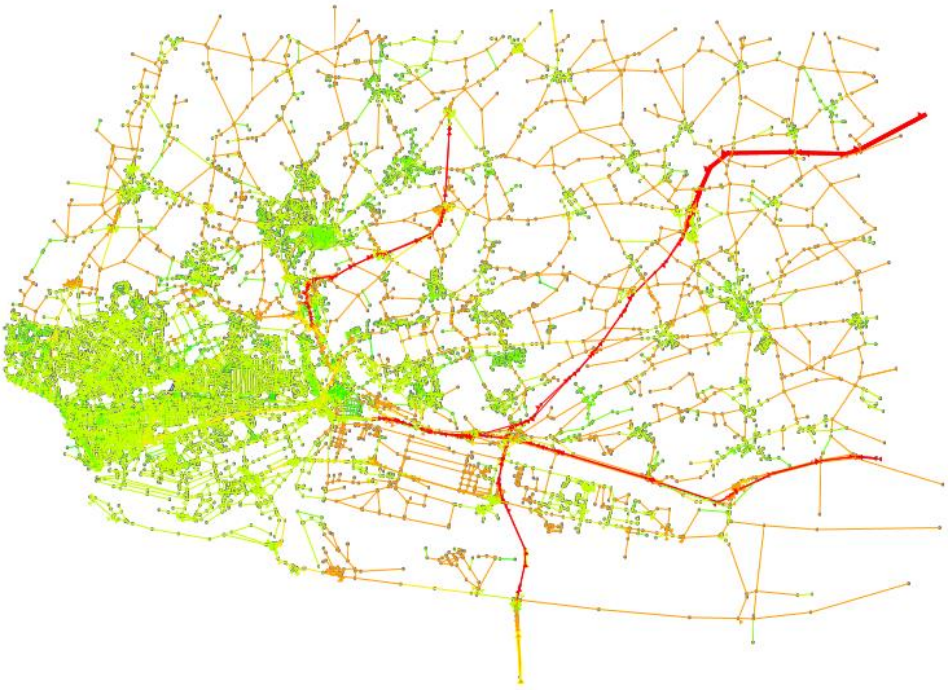


Figure 21 - edges and nodes coloured, dynamically, according to value of their attributes

Finally, pinpointing relevant points of interest is also possible, even dynamically.

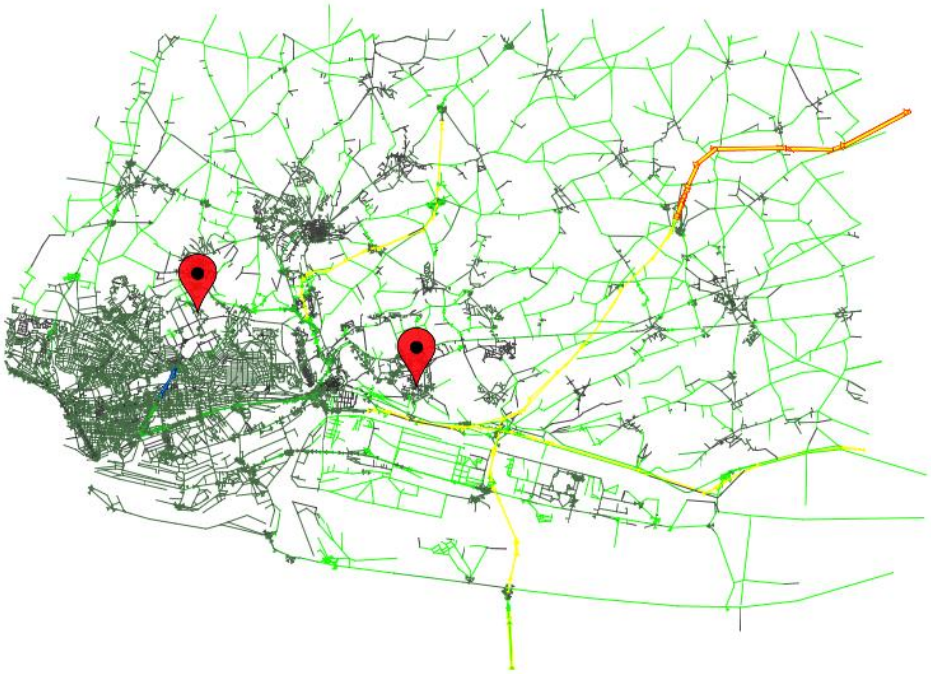


Figure 22 - dynamic colouring and pinpointing

GraphStream is not limited to graph modelling and representation; it also provides a wide set of algorithms from graph theory, such as:

- Connected Components
- Centroid
- Eccentricity
- Betweenness Centrality (both for nodes and edges)
- Several often used algorithms on graphs (degrees, density, diameter, clustering coefficient, etc.)
- Random walks on graphs
- Welsh-Powell (for graph colouring problem)
- Tarjan Strongly Connected Components
- PageRank
- Spanning Tree:
    - Base for spanning-tree algorithms
    - Kruskal
    - Prim
- Shortest Path
    - A* - it computes the shortest path from a node to another in a graph. It can eventually fail if the two nodes are in two distinct connected components.
    - All Pair Shortest Path - implements the Floyd-Warshall all pair shortest path algorithm where the shortest path from any node to any destination in a given weighted graph (with positive or negative edge weights) is performed.
    - Bellman-Ford - computes single-source shortest paths in a weighted digraph (where some of the edge weights may be negative). Dijkstra's algorithm accomplishes the same problem with a lower running time, but requires edge weights to be non-negative. Thus, Bellman-Ford is usually used only when there are negative edge weights

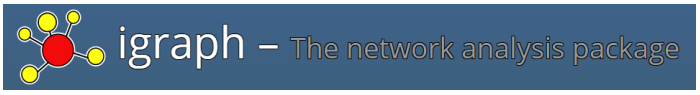
- Dijkstra - computes the shortest paths from a given node called source to all the other nodes in a graph. It produces a shortest path tree rooted in the source. This algorithm works only for nonnegative lengths.

One of the most relevant features of GraphStream is its extendibility, following specifications defined into the documentation and related to both algorithms and visualization.

Cytoscape[10] is an open source software platform originally created for visualizing molecular interaction networks and biological pathways, and integrating these networks with annotations, gene expression profiles and other state data. Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization.

Cytoscape core distribution provides a basic set of features for data integration, analysis, and visualization. Additional features are available as Apps (formerly called Plugins). Apps are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. They may be developed by anyone using the Cytoscape open API based on Java™ technology and App community development is encouraged. Most of the Apps are freely available from Cytoscape App Store.

A lot of algorithms, in particular graph-clustering are available, while they are currently missing in GraphStream. On the other hand, GraphStream is more powerful than Cytoscape with respect to the possibility to dynamically and programmatically modify and visualize graphs. Since both software are Java based, it is quite easy to exploit functionalities from both the libraries.



igraph[11] is a collection of network analysis tools with the emphasis on efficiency, portability and ease of use. igraph is open source and free and can be programmed in R, Python and C/C++. When the R-based release of igraph is used, it proves to be really effective and powerful for operations on matrices such as Adjacency and Laplacian matrices associated to a graph. It offers the possibility to easily compute eigenvalues and eigenvectors of these matrices thus enabling the computation of spectral gap and algebraic connectivity (which are not provided in GraphStream and Cytoscape).

[10] http://www.cytoscape.org/

[11] http://igraph.org/redirect.html

MATSim[12] provides a framework to implement large-scale agent-based transport simulations. The framework consists of several modules which can be combined or used stand-alone. Modules can be replaced by own implementations to test single aspects of your own work. Currently, MATSim offers:

- a framework for demand-modelling, agent-based mobility-simulation (traffic flow simulation), re-planning,
- a controller to iteratively run simulations as well as methods to analyse the output generated by the modules.

The key features of MATSim are the following:

- Fast Dynamic and Agent-Based Traffic Simulation: MATSim is able to simulate whole days within minutes
- Private and Public Traffic: both private cars and transit traffic can be simulated
- Supports Large Scenarios: MATSim can simulate millions of agents or huge, detailed networks
- Versatile Analyses and Simulation Output: compare simulated data to real-world counting stations
- Modular Approach: MATSim can be easily extended with your own algorithms
- Open Source and Java based, running on all major operating systems
- Active Development: a large community of developers constantly adds new features and improves MATSim

The following table 3 compares the selected software for data analysis and network analysis according to relevant characteristics required for the implementation of the decision support functionalities. Votes are expressed through a sequence of "*", where * is the lowest vote and *** is the highest. MATSim is not included in the table since it is the only transport software identified. From the table it is easy to understand that the integration of the different software allows to cover a wider set of basic functionalities enabling the implementation of most sophisticated decision support functionalities to be used in RESOLUTE.

Table 3 – Comparative evaluation of software for data analysis and network analysis

Tool	Analytical Functionalities	Visualization	Dynamic Modifications	Interoperability	Extendibility
Data Analysis					
R	***	**	**NA**	**	***
WEKA	***	*	**NA**	***	***
Network Analysis					
GraphStream	**	***	***	***	***
Cytoscape	**	***	*	**	**
igraph	***	**	*	**	***

[12] http://www.matsim.org/

7 EXPERIMENTAL RESULTS

This section presents preliminary results obtained from the tests performed during the development phase. Results are divided according to the two modules of the Application Framework, which are user profiling and network analysis.

7.1 Results on User Identity and Profile Management

7.1.1 Agents profiling

The three features defined in the previous section were used to create 2D plots of the agents of the used dataset, by representing each agent as a point. This kind of representation allows for a visual inspection of the similarities and differences between agents with regard to the various mobility characteristics considered, so that any patterns of mobility behaviours can be apparent to the human eye.

The Average path length and Entropy features defined in the previous section are scalar values, so they can be used together to form a 2-dimensional plot, where each axis corresponds to one of the features and the agents can be plotted as points, using their values for each feature as the coordinates. The following Figure 23 illustrates this kind of plot, as was produced for the agents of the synthetic dataset. The horizontal axis corresponds to the Average path length feature, while the vertical axis corresponds to the Path entropy feature. Each point in the plot corresponds to an agent, while the colours denote the types of agents, with respect to their mobility capabilities. It can be observed that the points corresponding to the agents with no mobility problems (blue points) are separated from the points corresponding to agents with mobility problems, mostly along the average path length axis.

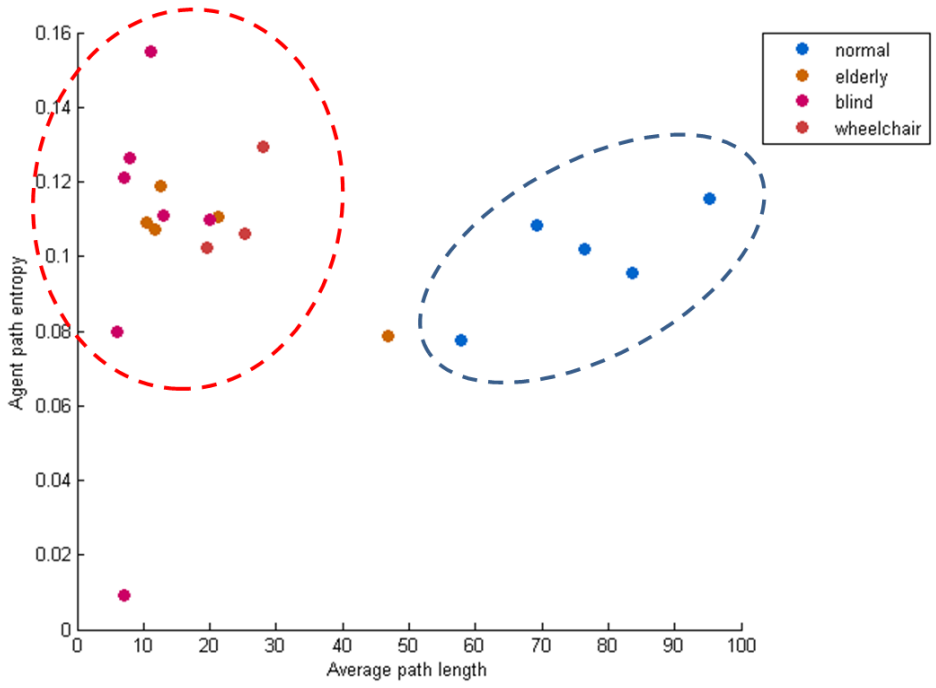


Figure 23 - Plot of agents using the average path length and the agent path entropy as coordinates. Each point corresponds to an agent. Colours denote different agent classes. Points corresponding to agents with no mobility problems (blue points) are separated from points corresponding to agents with mobility problems.

The agent features based on the Frechet distance are high-dimensional vector descriptors. In order to plot each agent as a point on a two-dimensional space, the feature vectors of the agents were hereby truncated to two dimensions. Again it can be observed that the points corresponding to agents with no mobility problems are visually separated from the points corresponding to agents with mobility issues. This shows that the corresponding classes of agents have different behavioral profiles with respect to the geometric forms of the paths they follow (Figure 24).

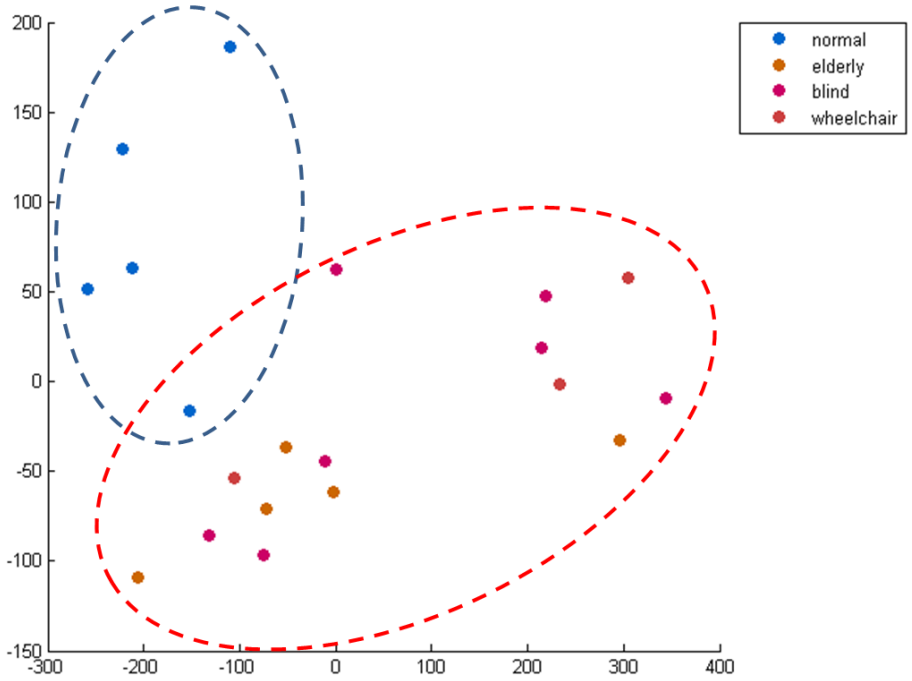


Figure 24 - 2D plot of the agent features based on the Frechet distance. Each point corresponds to an agent. The high-dimensional Frechet-based features were truncated to 2 dimensions, in order to be presented. The colors denote the different agent classes. Points corresponding to agents with no mobility problems (blue points) are separated from points corresponding to agents with mobility problems.

As a further investigation, the Figure 25 depicts a similar MDS-based plot of the individual paths followed by all agents. The features for individual paths were computed similar to the Frechet-based features of the agents, as described in the end of the previous section. Each point in the figure corresponds to an individual path, while the colors still denote the class of the agent that followed each path.

A first observation from this figure is that the paths tend to form several small clusters. Each of these clusters corresponds to a different agent (there are 20 clusters observed, corresponding to the number of agents in the dataset). This suggests that, in the dataset used, the paths that a single agent follows tend to be similar in form. Moving to a larger scale, the paths followed by agents with no mobility problems (blue points) are seen to be gathered towards the top of the plot, separated from most of the paths followed by agents with mobility problems, which is a complementary confirmation of the results obtained by the previous two experiments.

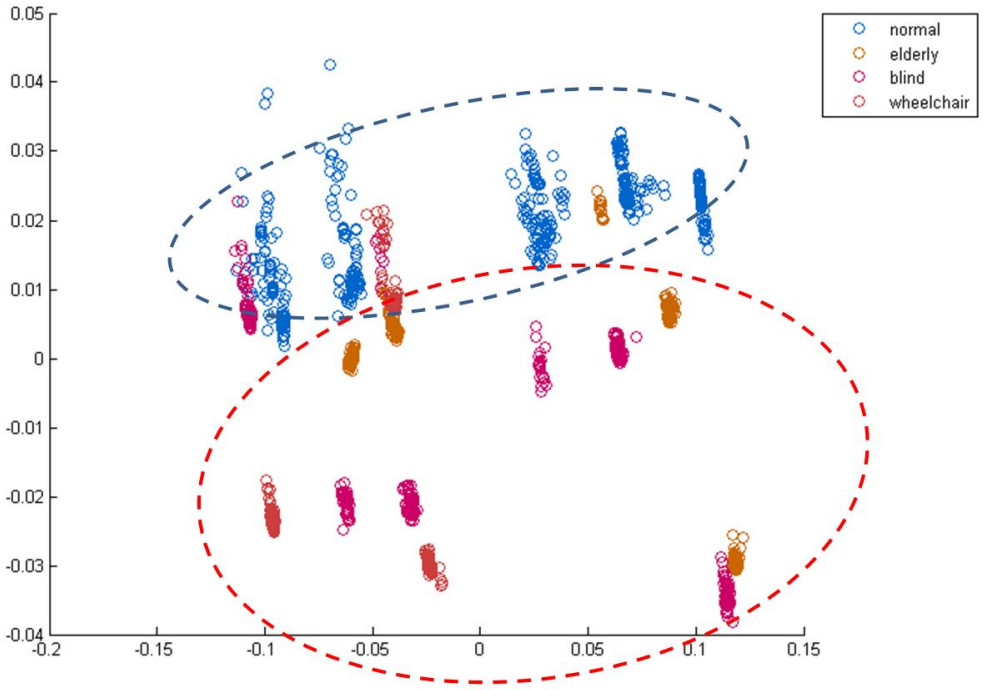


Figure 25 - Plot of all agent paths, using the Frechet features as coordinates. Each point is a path of an agent and colors denote the agent classes. Paths tend to form small clusters, with each cluster corresponding to a different agent, suggesting that each agent of the dataset tended to follow paths similar in form. In a larger scale, the paths corresponding to agents with no mobility problems are gathered towards the top of the plot, separated from most of the paths of agents with mobility problems, which are gathered towards the bottom.

As a final experiment, the three agent features described above were combined using the multi-objective visualization method described in (Kalamaras et al., 2014), producing the visualization of Figure 26. In short, for each of the different features, a distance matrix is formed, using the Euclidean distance among the features. The distance matrix is used to construct a complete undirected weighted graph, having the agents as its vertices and the distances among them as the weights of the edges. The minimum spanning tree (MST) of this graph is then formed. Vertices which are connected in the MST correspond to agents having a small distance between them, i.e. having similar mobility characteristics. The MSTs for the features used hereby are depicted in the following set of Figures, where a force-directed placement algorithm is used for the vertex placement.

After the MSTs for the individual features have been constructed, a combined graph is formed, where the weight of an edge connecting two vertices is the weighted sum of the weights of the edges connecting the corresponding vertices in the multiple individual MSTs. Vertices that are connected in the combined graph correspond to agents with similar mobility characteristics, regarding not only a single one, but all the used features. The combined graph for the data used hereby, using equal weights for the multiple individual features, is presented in Figure 26(d). The combined graph manages to combine information from all features, being able to separate the agents with no mobility problems (blue points) from the agents having mobility problems. It is this combined graph that will be ultimately presented to the operator, in order to assist him/her in discovering differences in the mobility profiles of the agents.

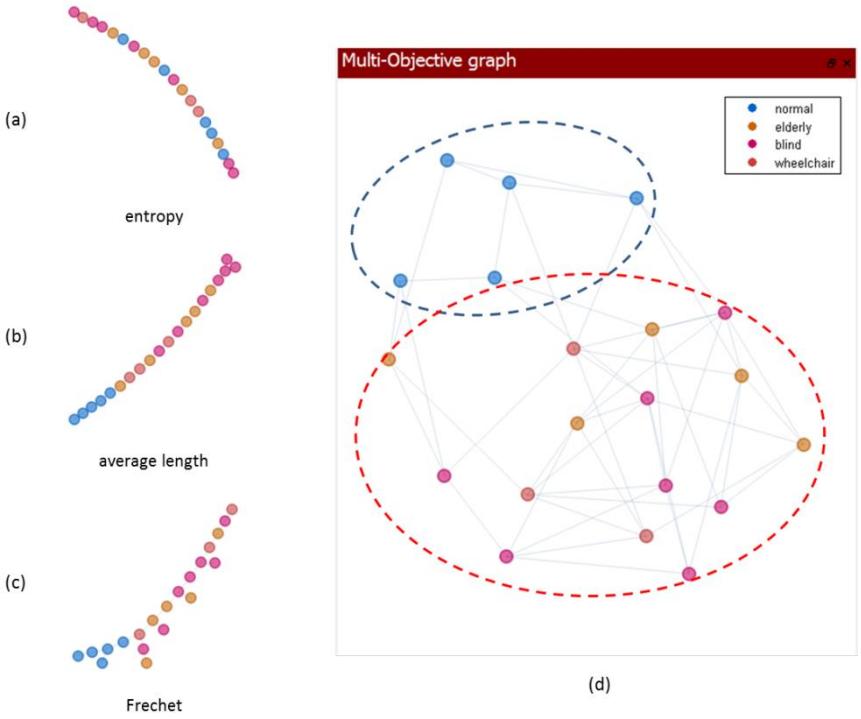


Figure 26: Multi-objective visualization of the three described features. (a)-(c) Visualizations of each feature separately. (d) Visualization of the combination of the three features, as will be presented to the operator. Each point denotes an agent and colors denote different agent classes. Again, points corresponding to agents with no mobility problems (blue points) are visually separated from points corresponding to agents with mobility problems.

The following Figure 27 illustrates a visualization similar to the previous one. However, hereby the length and entropy features, which are both one-dimensional, have been merged into a two-dimensional feature. Thus, two features are provided as input to the multi-objective visualization method: the combined length-entropy feature and the Frechet feature. The MSTs for these two features are depicted in (a) and (b). The combined graph produced using them is illustrated in (c). It can be observed that again the users with no mobility problems (blue points) are visually separated from those with mobility problems. In fact, the visual separation between the two groups of points is now more distinct.

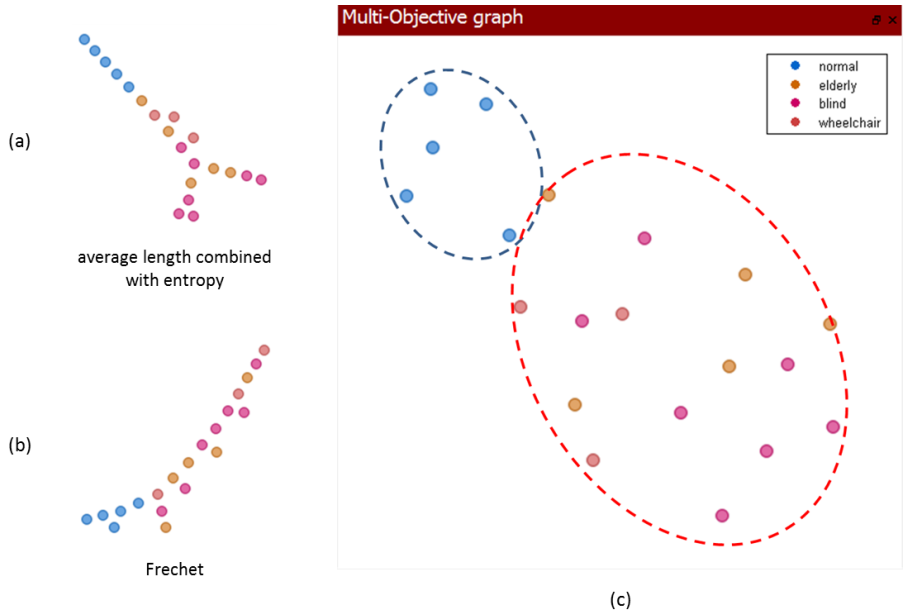


Figure 27: Multi-objective visualization of the agent mobility data, where the average length and entropy features are merged as a single 2D feature type. (a) Visualization of the combined length-entropy features. (b) Visualization of the Frechet features. (c) Visualization of the combination of the length-entropy and the Frechet features using the multi-objective method, as will be presented to the operator. Each point denotes an agent and colors denote different agent classes. Again, points corresponding to agents with no mobility problems (blue points) are visually separated from points corresponding to agents with mobility problems.

7.1.2 UTS users behaviour analysis

With respect to the analysis of UTS'users behaviour, the open-data provided by Transport for London (TfL), related to the number of entering and exiting passengers every 30 minutes from every metro station, has been considered.

The available dataset consists of entries and exits, every 30 minutes, for every station and for a period of two weeks (11 to 24 October 2015). To perform the analysis, the data has been organized in a dataset where every row is associated to a specific station and a specific date. The number or columns – excluding the identifier of the station and the date – are *n* time stamps, with a step of 30 minutes, from the beginning to end of service (the end of service could be in the first hours of the day following the date associated to the row).

These vectors, consisting of *n* numeric components (i.e. passenger counts), have been clustered together in order to identify groups of similar behaviour. The size of clusters allows for identifying the most frequent (i.e. typical) with respect the most unusual (i.e. anomalous patterns due to some event). Since transport usage is characterized by habits of the users, peaks and bursts in the passenger counts time series should occur at the same in the day. The similarity measure knows as **cosine distance** (also named **triangle similarity**) has been used to compare, and then group together, time series according to this specific consideration, so different typical patterns (i.e. representative time series for each cluster) will characterize different users mobility habits/behaviours.

More in detail, cosine similarity is given by the cosine of triangle between two vectors, so the range of value of cosine similarity is [-1; 1].

$$s(x,y)=\frac{\langle x,y\rangle}{\|x\|\|y\|}$$

As the components of the analysed vectors are not negative (i.e. they are counts) cosine similarity varies in [0; 1]. To perform the time series clustering, a *K*-means algorithm has been adopted, in particular the implementation of the Spherical *K*-means provided by the R package and named "skmeans". This specific implementation performs a simple *K*-means strategy based on the cosine distance, which is directly computed starting from the cosine/triangle similarity:

$$d(x,y) = 1 - s(x,y) = 1 - \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

The most suitable number *k* of cluster has been selected via the "Silhouette" clustering validity measure (Arbelaitz et al., 2013).

As a result, some behaviour (passengers counts time series) are more frequent, according to the size of the cluster they belong to, while others are really sporadic/anomalous.

In Figure 28 two recurrent patterns are reported, just as an example. The "protoypes" are depicted, which are the representative of each cluster, scaled into 0-1 to make them comparable (this is why on the y-axis is not the numeric count of passengers).

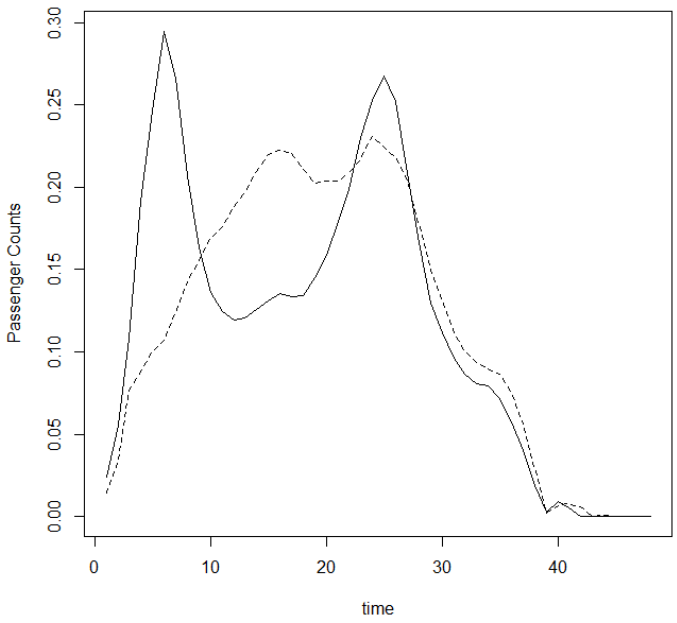


Figure 28 - two typical behaviours obtained through time series clustering of the passenger counts time series (counts are scaled in the range 0-1 to make comparable the "prototypes" from different clusters)

In Figure 29, instead, two unusual patterns are presented, along with the two previous typical ones. It is easy to understand the big differences, in particular the variation with respect to the more typical mobility behaviour measured through the passenger counts.

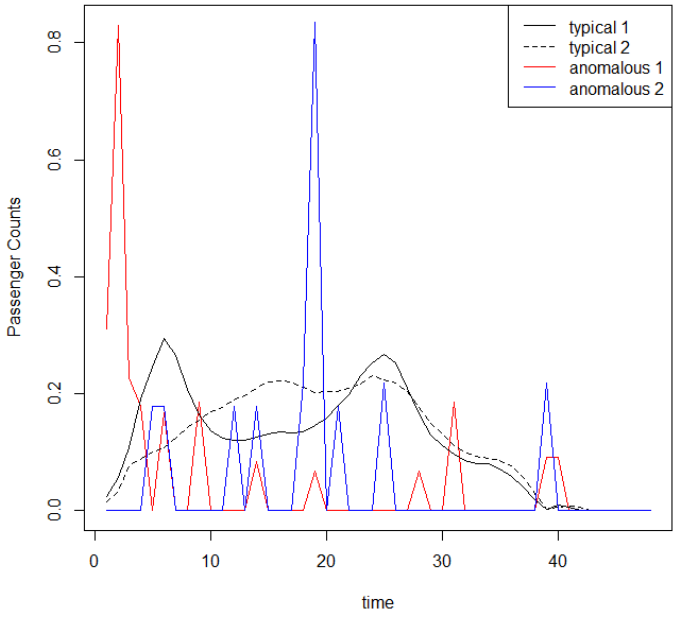


Figure 29 - two anomalous behaviours – compared to the two previous typical ones.

To conclude this analysis, it is really important to take into account clusters associated to unsual patterns. For each one of them, the original time series belonging to that cluster have to be considered in order to retrieve both stations and dates. In particular, when the original series are related to the same date and to different stations this means that some kind of event/disruption had an impact on the UTS and that its users try to rearrange thei plan to achieve destinations. This will permit to analyze and study which are (i) the relationship between the event/disruption at a specific location and its impact on the overall set of stations; (ii) the possible relationship among stations, even according to specific time of the day or type of day (weekdays or weekend/holydays); (iii) the evolution and changes in mobility habits of the UTS users over time (when an online and adaptive clustering schema is implemented).

Furthermore, the application of the proposed time series clustering approach could be also applied to the passenger counts time series of each station, separately. In this case a better characterization of typical/unusual behaviours could be possible at each station. However, due to the limited set of open-data provided by TfL (i.e. 14 time series for every station) this second approach resulted inapplicable.

7.2 Results on Network Analysis

7.2.1 Transportation network in Florence

In this section, the results obtained from network analysis applied on the public transportation network in Florence – stored into the RESOLUTE Knowledge Base – are presented.

The public transport network is modelled through a directed multi graph. Multi-graph is used because more than one route/line may connect two bus stations; direction on edges is used to model direction of each line from one stop to the next one.

The following figure shows the graph model build from the public transport network, where a specific line has been highlighted with a different colour for the two possible directions (forward and backward).

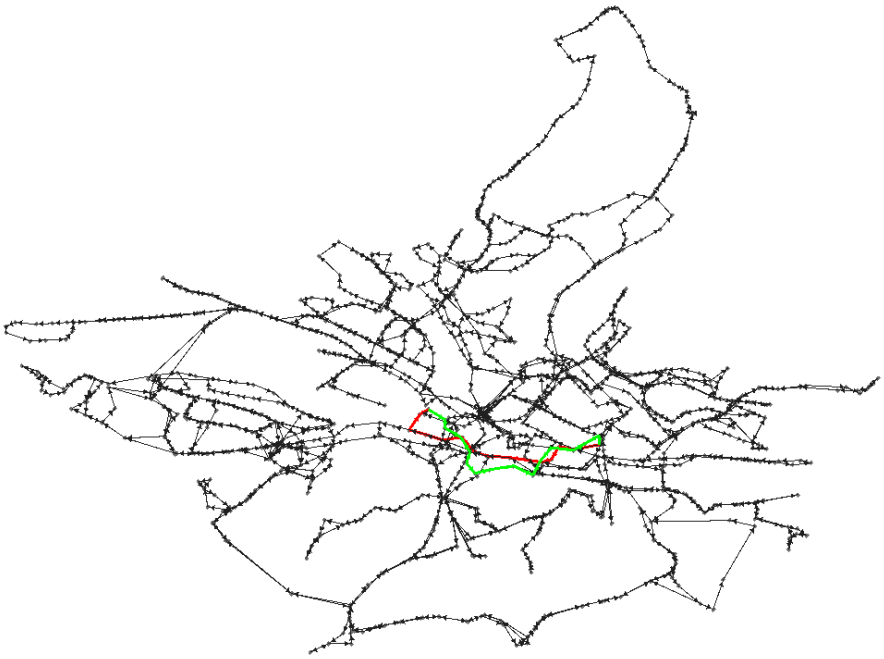


Figure 30 - directed multi-graph associated to the public transport network in Florence (data retrieved from the RESOLUTE Knowledge Base) as shown by GraphStream viewer

In Figure 31 the nodes having one of the highest 5 values of **degree** (hubs) are highlighted. The colour scale adopted is **red**, **orange**, **yellow**, **cyan** and **green**, according to descending degree.

The same colour scale will be used also for the other graph-based measures.

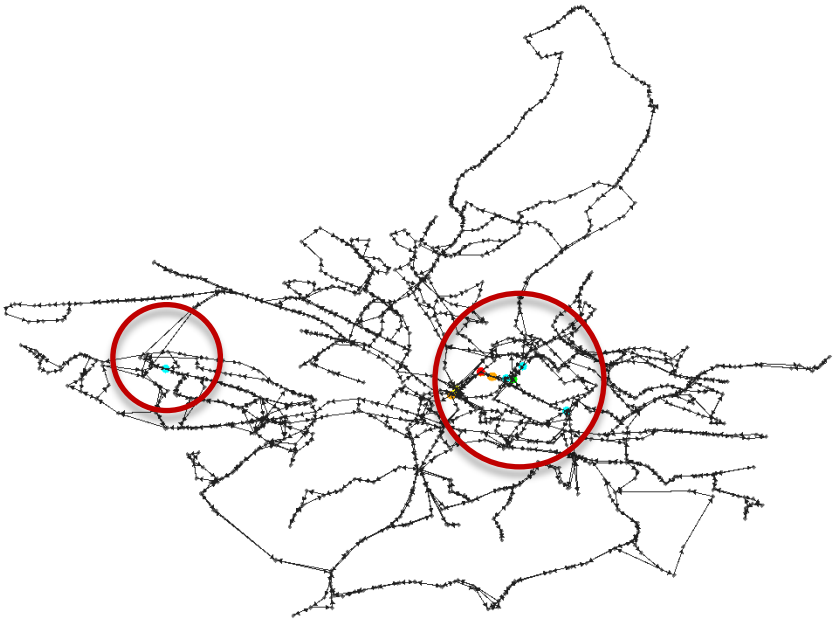


Figure 31 - Nodes with highest values of degree

Figure 32 is a zoom-in of the central zone of the network, where most of the hubs are located.

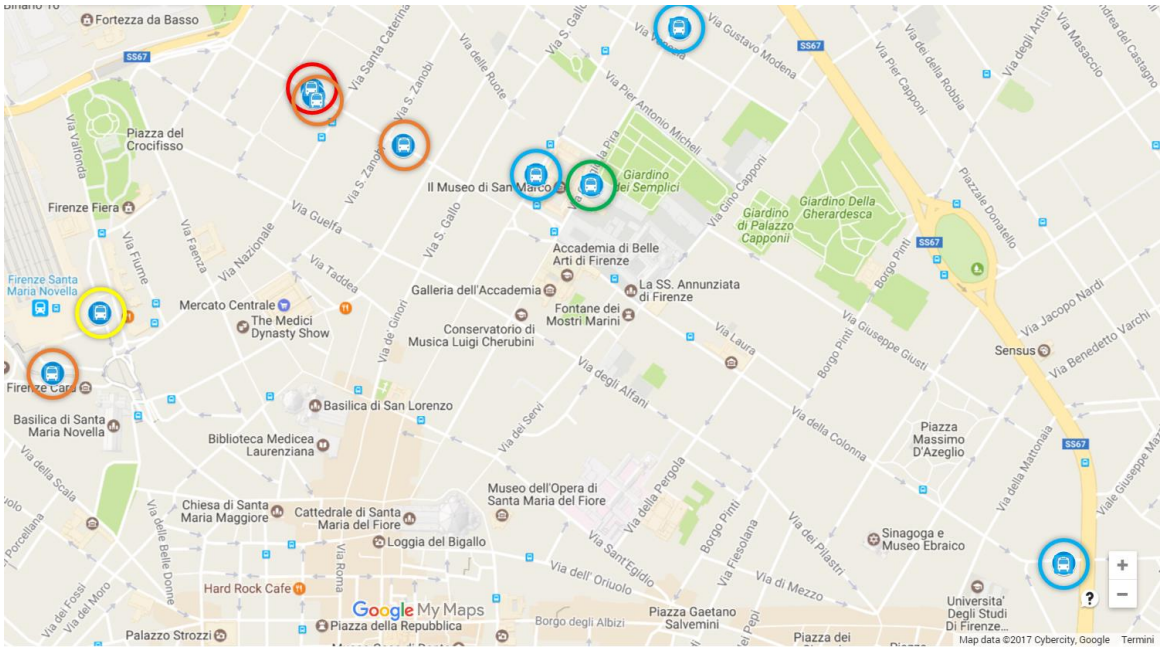


Figure 32 – Stops associated to the highest degree of the central area of the transportation network in Florence

In Figure 33 a zoom of the hub in the southern part of the network is shown.

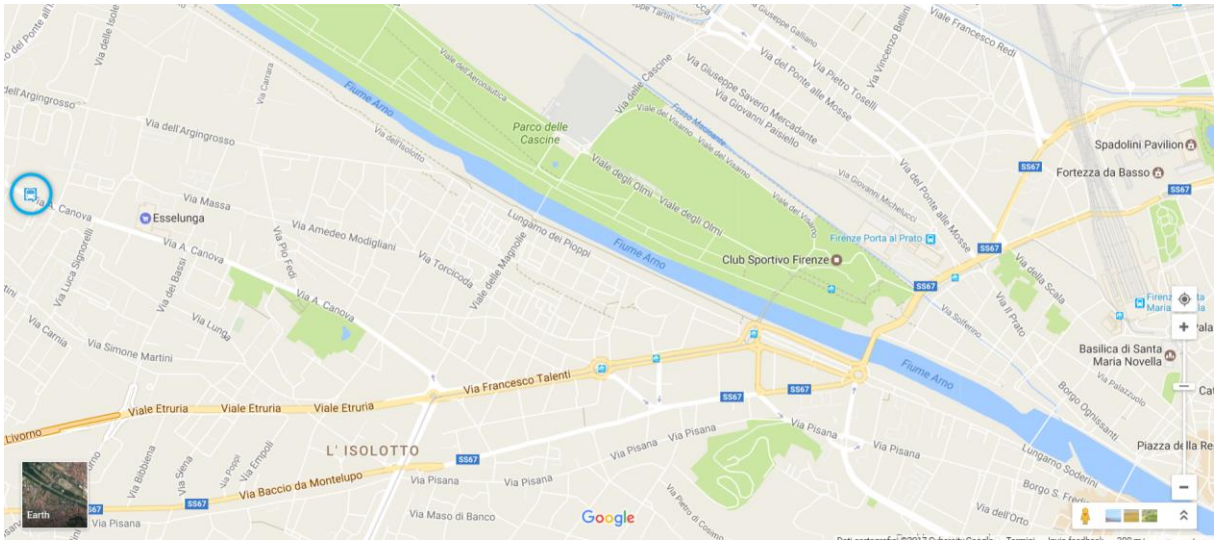


Figure 33 - Zoom in of the southern area of the network

As already mentioned in chapter 5, another possible graph-based measure to identify relevant/critical nodes is (node) betweenness. Contrary to degree, betweenness is a measure based on paths – therefore global rather than local connectivity. According to this consideration, it is easy to understand why the set of relevant/critical nodes identified is different from the previous case.

Figure 34 - Five nodes with highest (node) betweenness

The following figure shows a zoom-in of the area where nodes with highest betweenness are located.

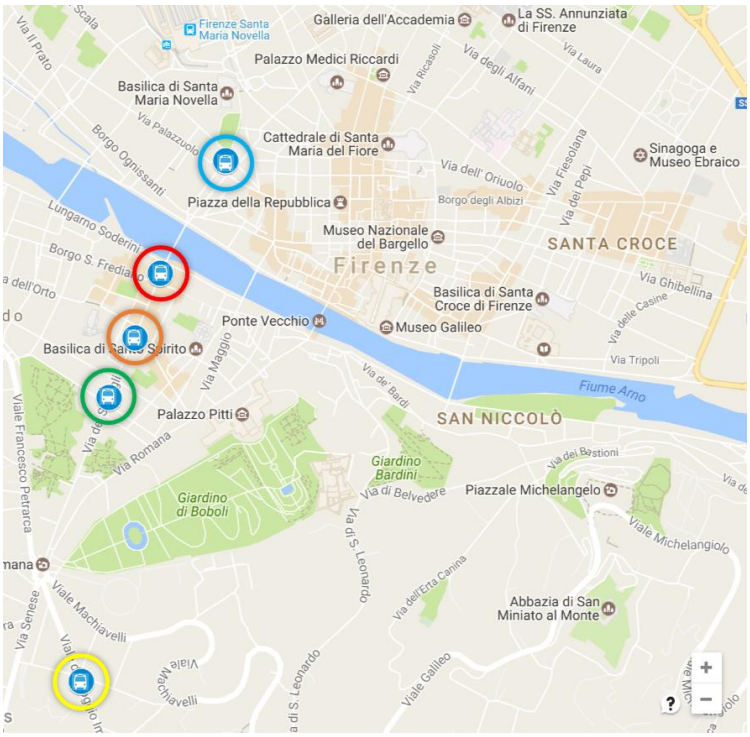


Figure 35 - Zoom of the five nodes with highest (node) betweenness

To address the analysis of connectivity on edges, edge betwenness is firstly considered. The following figure shows the five segments, into the graph, having highest edge betweennness.

Figure 36 - Five segments (sequence of consecutive edges) with highest edge betweenness

Following, a zoom of the five edges with highest betweenness.

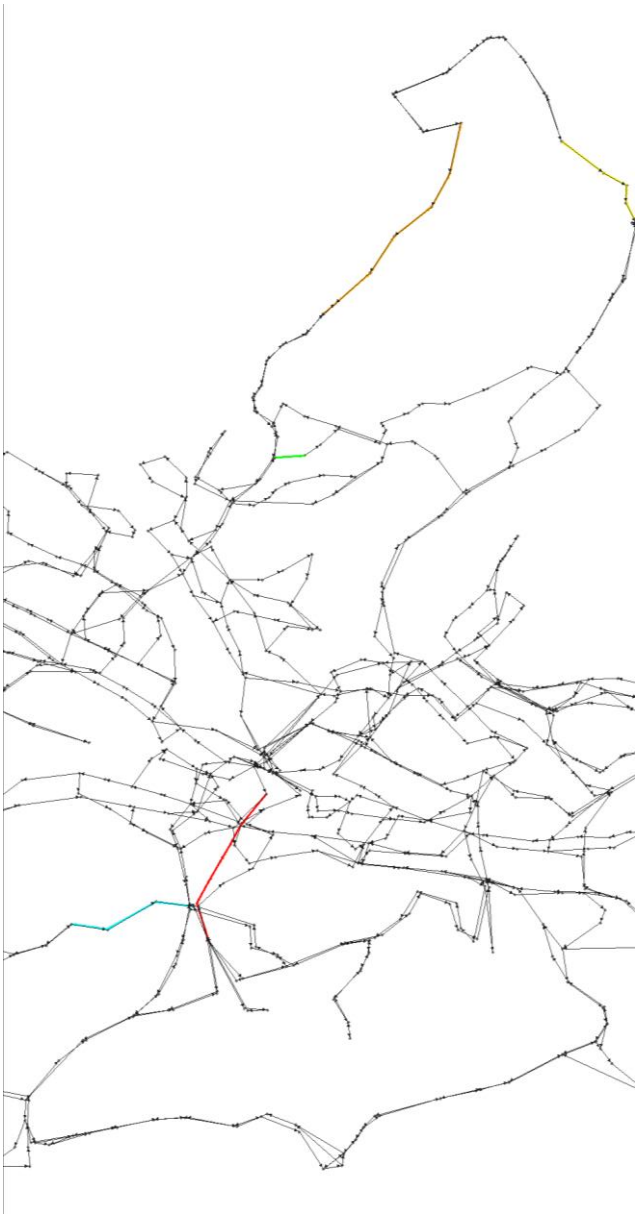


Figure 37 - Zoom in of the five segments (sequence of consecutive edges) with highest edge betweenness

Although edges with high edge betweenness have a high probability to belong to the min edge cut set, graph clustering must be applied in order to obtain the min cut set.

To make more evident how the min cut set divide the network into two (or more) sub-graph, the visualization provided by Cytoscape is used (left hand-side of the following figure), then the edges belonging to the min cut set are highlighted, in red, in the right hand-side part of the following figure.

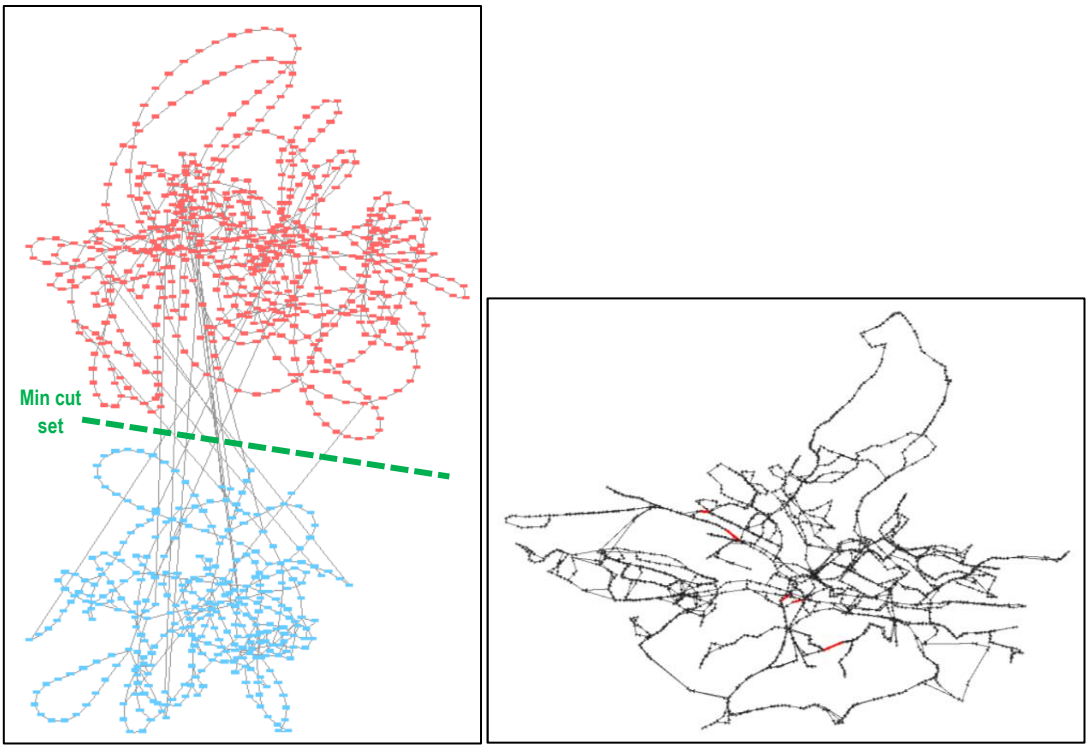


Figure 38 - Min cut set identified through graph clustering (in particular Spectral Clustering implementation in the Cytoscape's App "ClusterMaker2")

Edges belonging to the min cut set, identified through graph clustering, are different from the five with highest edge betweenness; however is important to highlight that faults of the edges in the min cut set imply a physical disconnection of the network: so, the higher the number of edges in the min cut set, the lower the probability a disconnection occurs.

Then events are considered and new critical elements, induced by the new configuration, is computed and compared with the previous one, for each one of the measures previously considered.

The first event is to make unavailable the stop associated to the node with highest degree in the graph, but allowing for "jumping" the stop, so avoiding interruption of the affected lines.

On the top of the following figure the new hubs, compared to the previous ones, on the bottom.

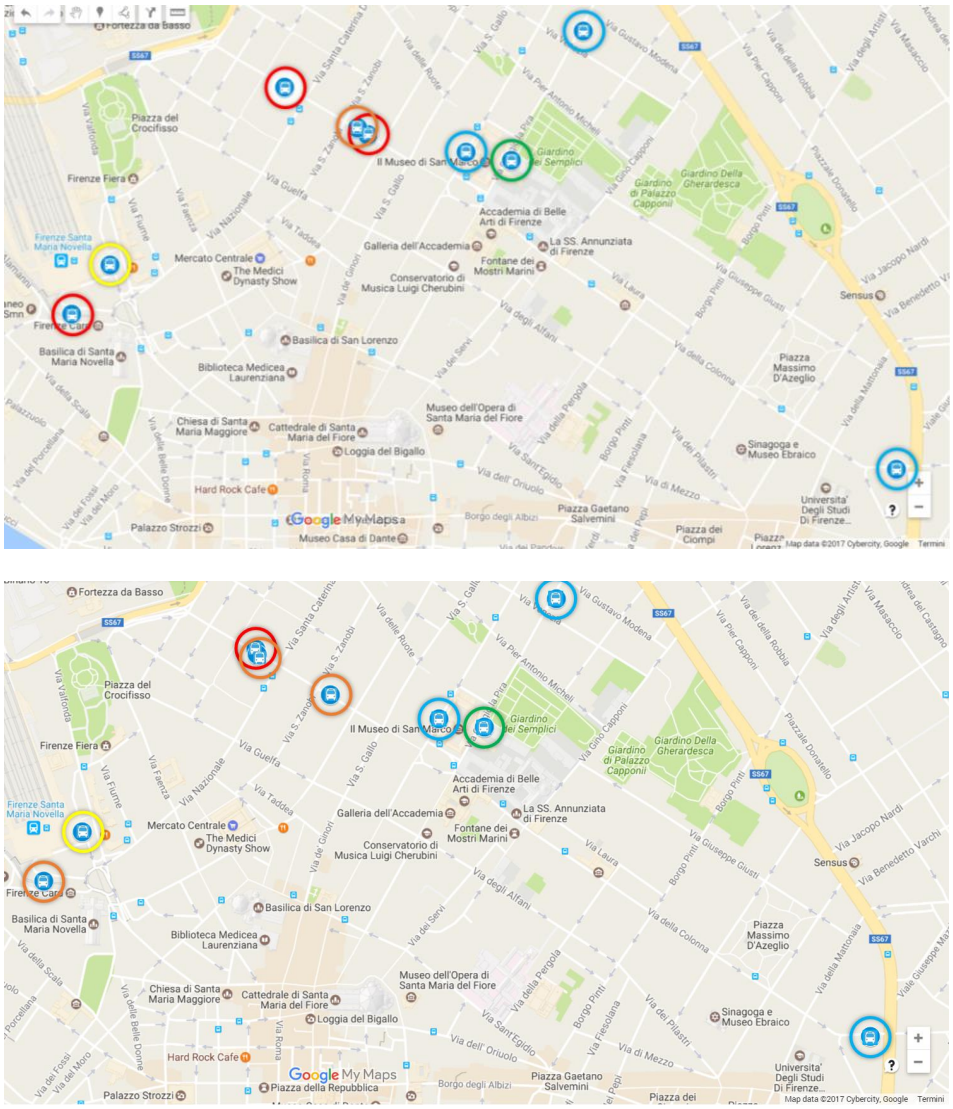


Figure 39 - Node degree: new (top) versus previous (bottom) hubs in the network

It is possible to see that the graph results significantly modified near the node with highest degree (before the event) and that its unavailability – when jumping stop is allowed – also modify relevance of nodes into the new setting.

The same type of event is then performed by considering the node with highest (node) betweenness. In this case modifications are less relevant and are more related to some changes in the order of the critical nodes.

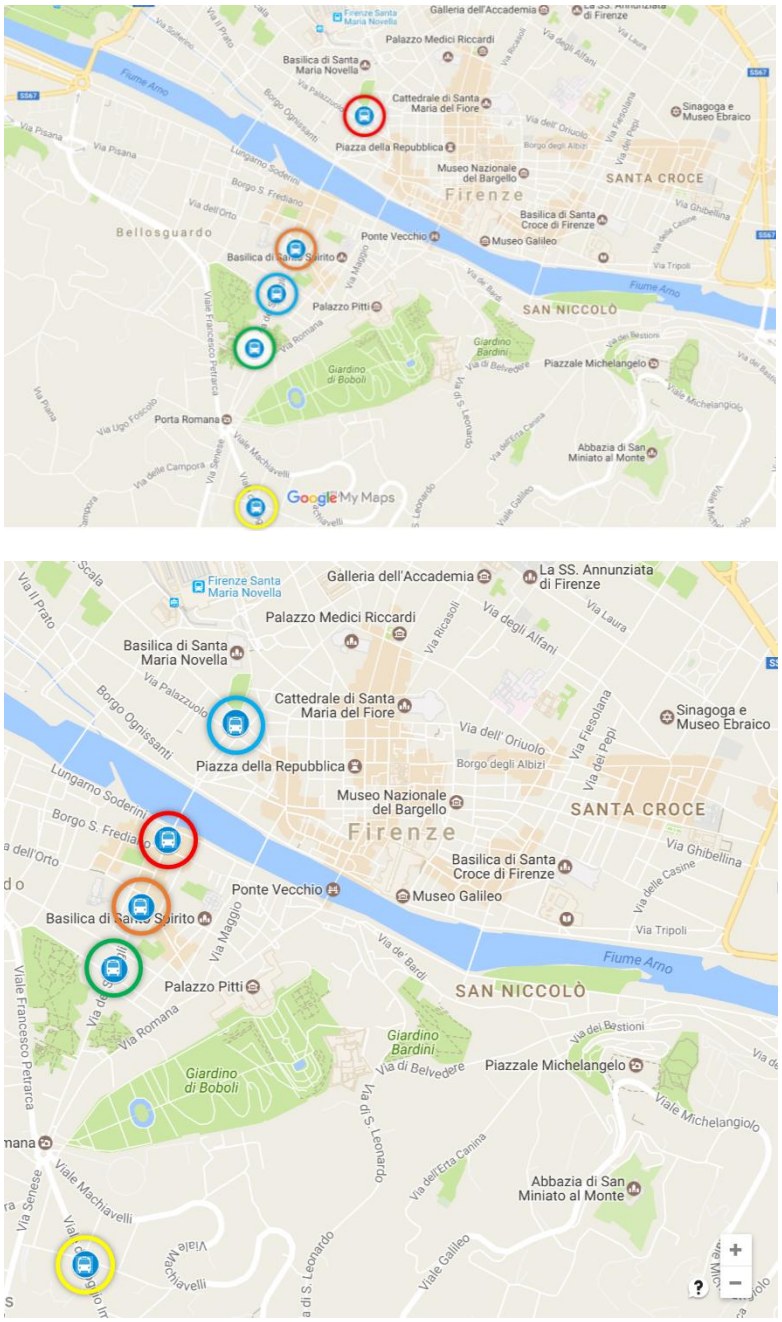


Figure 40 - Node Betweenness: new (top) versus previous (bottom) nodes in the network

Then the same event, but considering line interruption, is analysed. In this case it easy to understand that modifications on the graph are very significant, implying removal of a larger number of edges, and in case nodes.

From the following figure it is easy to see that spatial distribution of hubs significantly changes with respect to the normal setting.

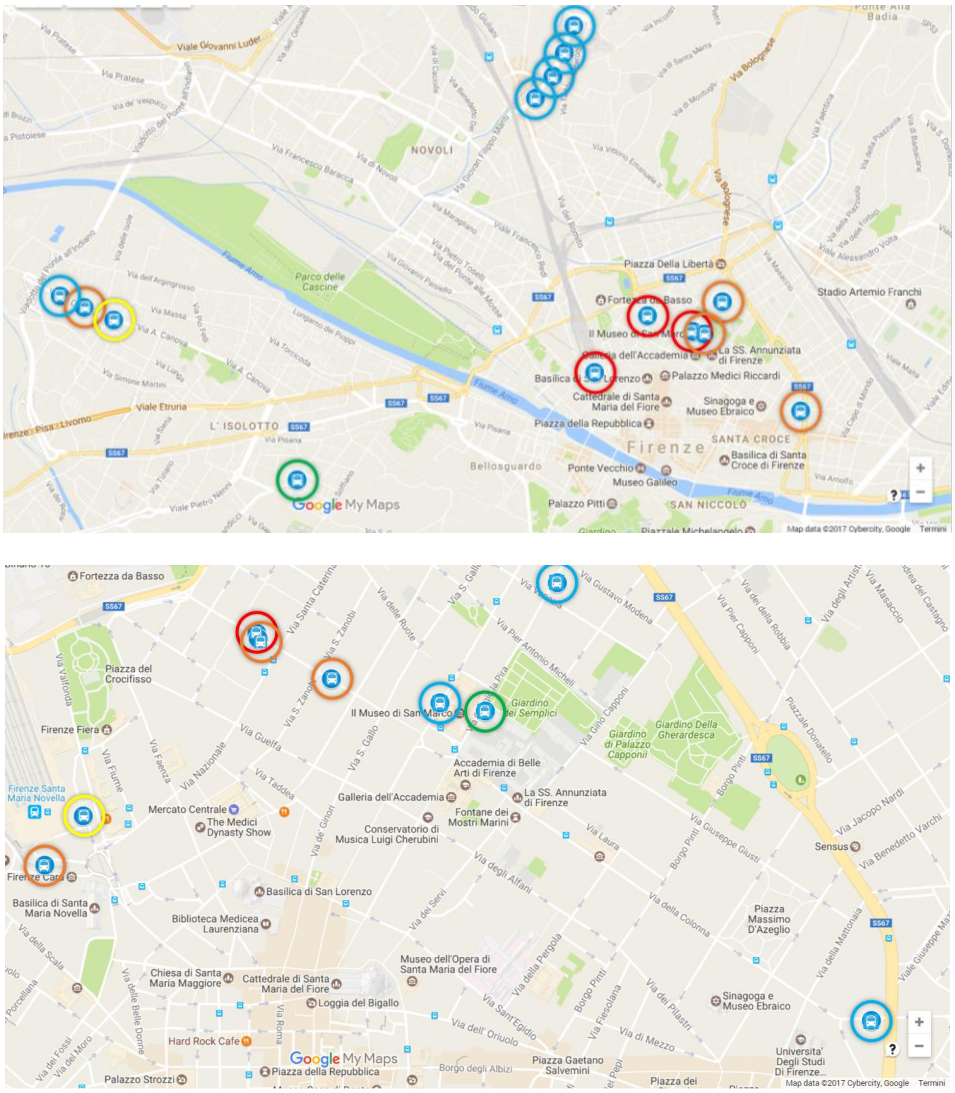


Figure 41 - Node Degree: new (top) versus previous (bottom) hubs in the network

The modifications induced on the graph is quite significant also in the case the node with highest node betweenness is selected. Again, the spatial distribution of hubs changes significantly.

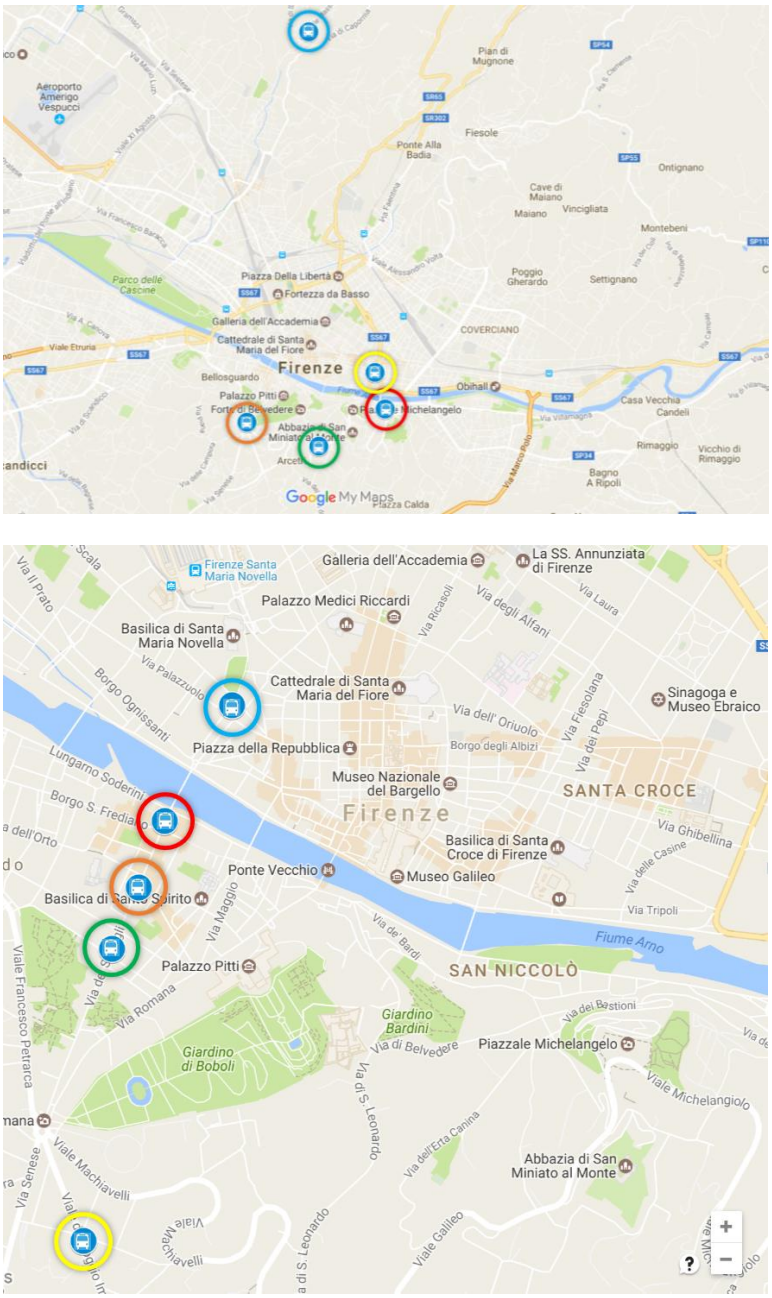


Figure 42 - Node Betweenness: new (top) versus previous (bottom) nodes in the network

Finally, modifications induced by interruption of lines passing through the edge with the highest value of edge betweenness are considered.

Figure 43 shows as ranking of the segments changes and a new segment is now among the five most critical.

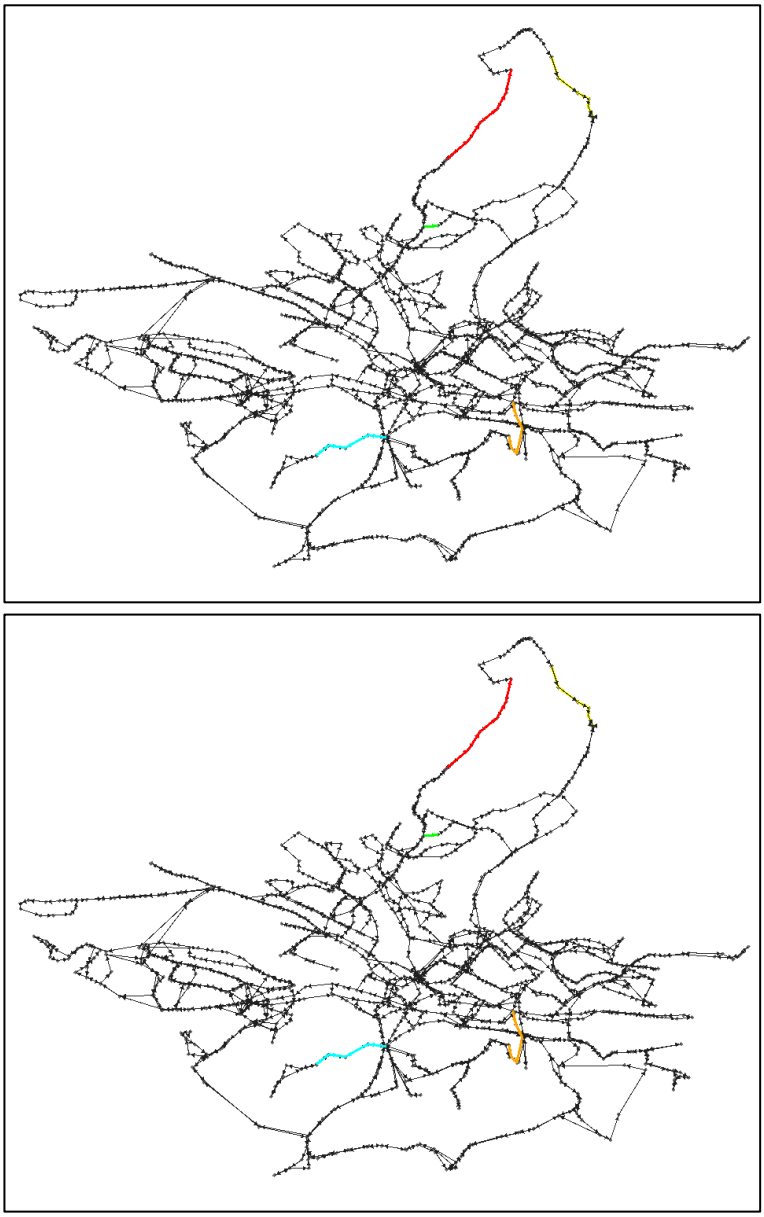


Figure 43 - Edge Betweenness: new (top) versus previous (bottom) edges in the network

Furthermore, network-wide measures from graph theory can be also computed for the “faultless” network and then compared to the new setting induced by the considered events.

The table 4 summarizes the values of the measures computed on the original "faultless" Florence UTS graph.

Table 4 – graph-based measures computed on the original UTS in Florence

Measure	Value	Note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	Only GraphStream
Diameter	53	
Shortest Paths	997002	
Characteristic path length	19,988	
average degree	6.46 (s.d. 5.23)	Only GraphStream
Density	0,006	in Cytoscape is "network centralization"
Algebraic connectivity	0.008155974	Only igraph
Spectral gap	2.211108	Only igraph

After removing the node with the highest degree and allowing for "jumping" station, the measures change as reported in the Table 5:

Table 5 – measures after removing node with the highest degree and allowing for "jumping" station, in Florence

Measure	Value	note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	
Diameter	53	
Shortest Paths	**995006**	*Changed*
Characteristic path length	**19,928**	*Changed*
average degree	**6.42 (s.d. 5.10)**	*Changed*
Density	0,006	
Algebraic connectivity	**0.008158018**	*Changed*
Spectral gap	**2.47689**	*Changed*

After removing the node with the highest degree and interrupting line(s) going from it, the measures change as reported in the Table 6:

Table 6 – measures after removing node with the highest degree and interrupting line(s), in Florence

Measure	Value	note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	
Diameter	53	
Shortest Paths	**995006**	*Changed*
Characteristic path length	**20,041**	*Changed*
average degree	**6.38 (s.d. 4.99)**	*Changed*
Density	0,006	
Algebraic connectivity	**0.008054851**	*Changed*
Spectral gap	**4.783684**	*Changed*

After removing the node with the highest node betweenness and allowing for "jumping" station, the measures change as reported in the Table 7:

Table 7 - measures after removing node with the highest node betweenness and allowing for "jumping" station, in Florence

Measure	Value	note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	
Diameter	53	
Shortest Paths	**995006**	*Changed*
Characteristic path length	**20,007**	*Changed*
average degree	**6.46 (s.d. 5.23)**	*Changed*
Density	0,006	
Algebraic connectivity	**0.008166731**	*Changed*
Spectral gap	**2.211107**	*Changed*

After removing the node with the highest node betweenness and interrupting line(s), the measures change as reported in the Table 8:

Table 8 - measures after removing node with the highest node betweenness and interrupting line(s), in Florence

Measure	Value	note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	
Diameter	53	
Shortest Paths	**995006**	*Changed*
Characteristic path length	**20,282**	*Changed*
average degree	**6.45 (s.d. 5.23)**	*Changed*
Density	0,006	
Algebraic connectivity	**0.008061911**	*Changed*
Spectral gap	**2.210969**	*Changed*

After removing the edge with the highest edge betweenness, the measures change as reported in the Table 9:

Table 9 - measures after removing edge with the highest edge betweenness, in Florence

Measure	Value	note
Clustering coefficient	0.015	
Average Clustering coefficient	0.0097	
Diameter	53	
Shortest Paths	997002	
Characteristic path length	**20.117**	*Changed*
average degree	**6.46 (s.d. 5.23)**	*Changed*
Density	0,006	
Algebraic connectivity	**0.008085002**	*Changed*
Spectral gap	**2.211108**	*Changed*

As it is possible to see from the previous tables, some network-wide measures do not modify depending on the events considered.

One of the most relevant results is related to the modification in the spectral gap after removing the node with the highest degree and interrupting line(s) going from it. The value of the spectral gap is indeed doubled, indicating that the new setting is more connected that the original one. Although the result seems to be counterintuitive, this really highlights the relevance of the removed node to the overall connectivity of the graph associated to the public transport network, since the overall connectivity (measured by the spectral gap) significantly changes.

Finally, all the above analyses can be performed also by weighting edges with information such as length, travelling time, number of travelling passengers, etc. These weights can be valued according to available data, in particular through statistics, real-time and/or simulation.

The relevant contribution of network analysis for RESOLUTE is to offer a set of analytical functionalities updated to the state-of-the-art – by integrating a number of software libraries and developed code – and to use tools to interact with and analyse graphs, dynamically and efficiently.

7.2.2 Transportation network in the Attika region

In this section, the results obtained from network analysis applied on the public transportation network in the Attika region – accessed through the open API[13] reported in section 3 – are presented.

Even in this case, the public transport network is modelled through a directed multi graph because more than one route/line may connect two stations/stops; direction on edges is used to model direction of each line from one stop to the next one. The following figure shows the graph model build from the public transport network as shown by the GraphStream viewer.

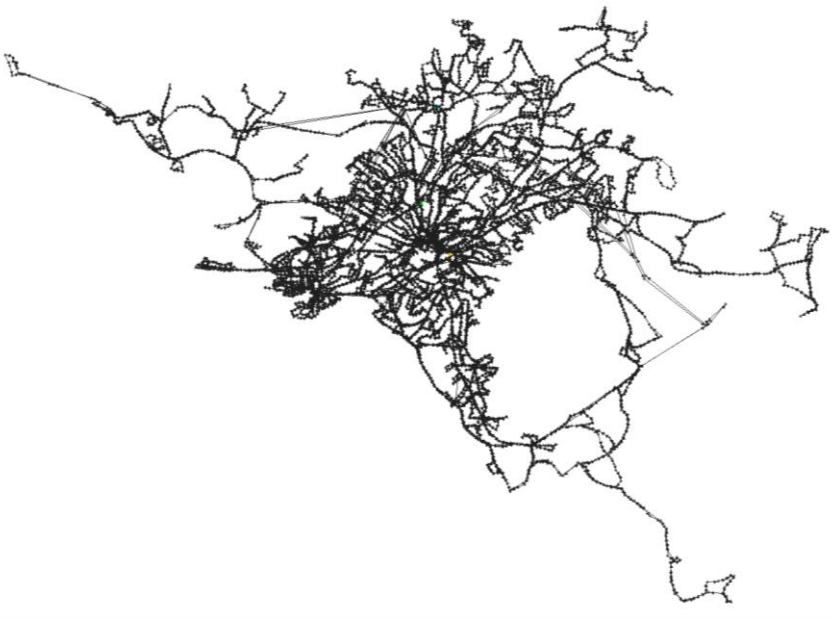


Figure 44 - directed multi-graph associated to the public transport network in the Attika region as shown by GraphStream viewer

[13] http://oasa-telematics-api.readthedocs.io/en/latest/

In figure 45 the nodes having one of the highest 5 values of **degree** (hubs) are highlighted. The colour scale adopted is **red**, **orange**, **yellow**, **cyan** and **green**, according to descending degree.

The same colour scale will be used also for the other graph-based measures.

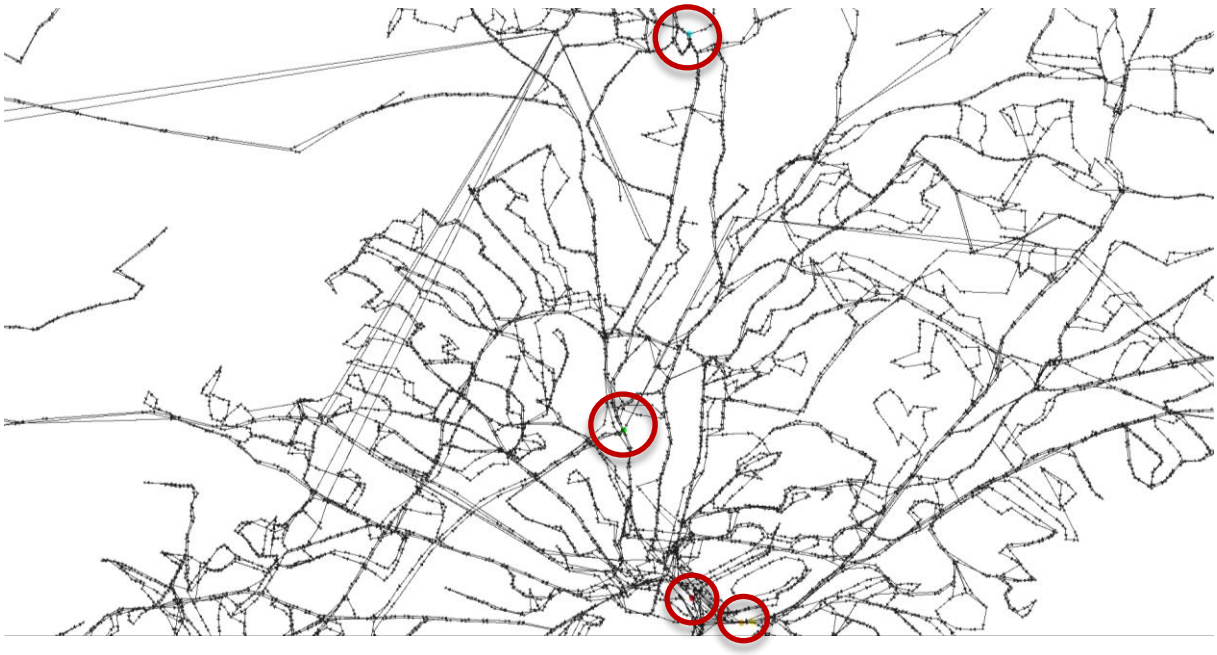


Figure 45 - Nodes with highest values of degree

Figure 46 shows where the hubs are located.

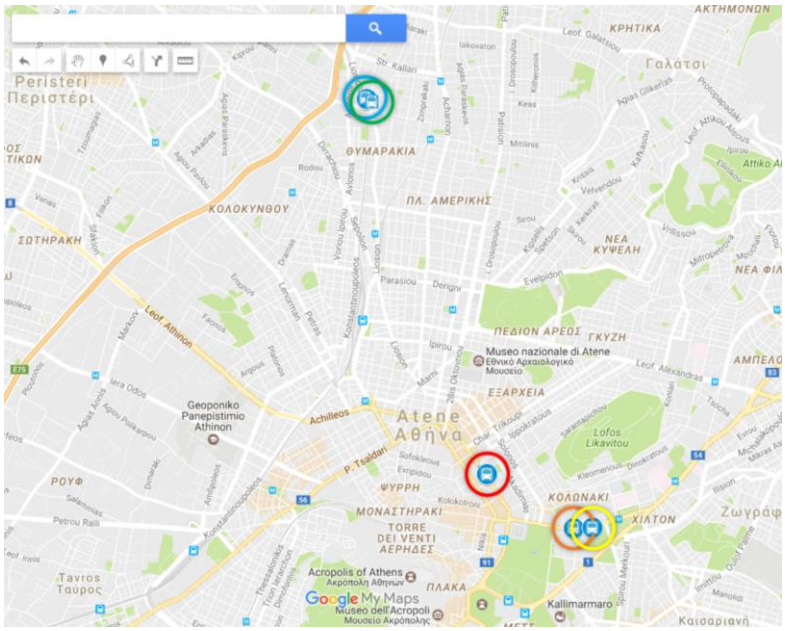


Figure 46 – Stops associated to the highest degree (hubs) of the transportation network in Athens

As already mentioned in chapter 5, another possible graph-based measure to identify relevant/critical nodes is (node) betweenness. Contrary to degree, betweenness is a measure based on paths – therefore global rather than local connectivity. According to this consideration, it is easy to understand why the set of relevant/critical nodes identified is different from the previous case.

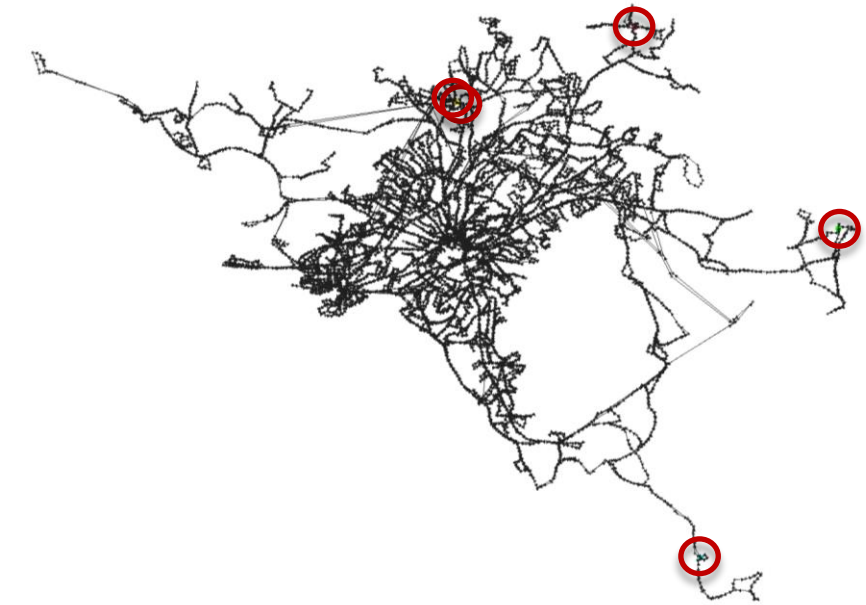


Figure 47 - Five nodes with highest (node) betweenness

The following figure shows the geographical location of the nodes (stops) with highest betweenness.

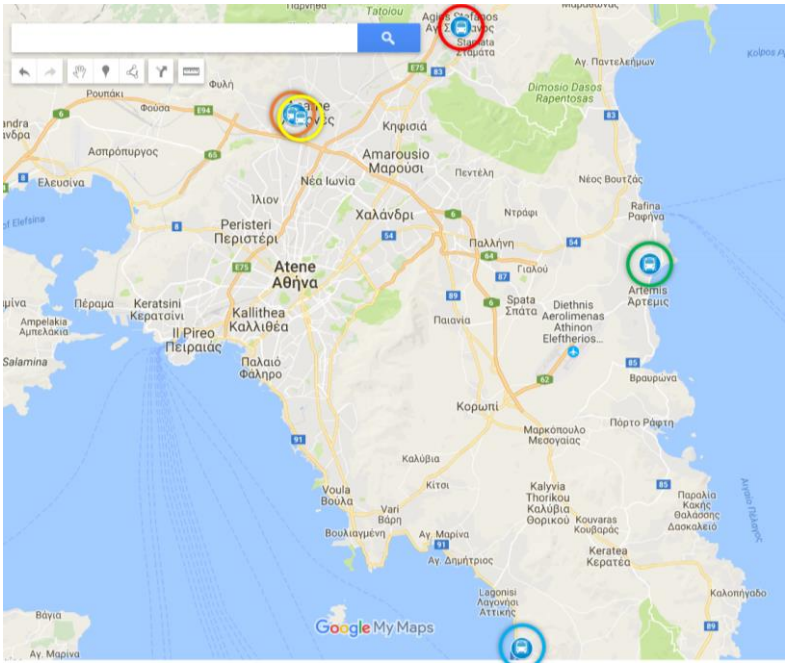


Figure 48 - Zoom of the five nodes with highest (node) betweenness

To address the analysis of connectivity on edges, edge betwenness is firstly considered. The following figure shows the five segments, into the graph, having highest edge betweennness.

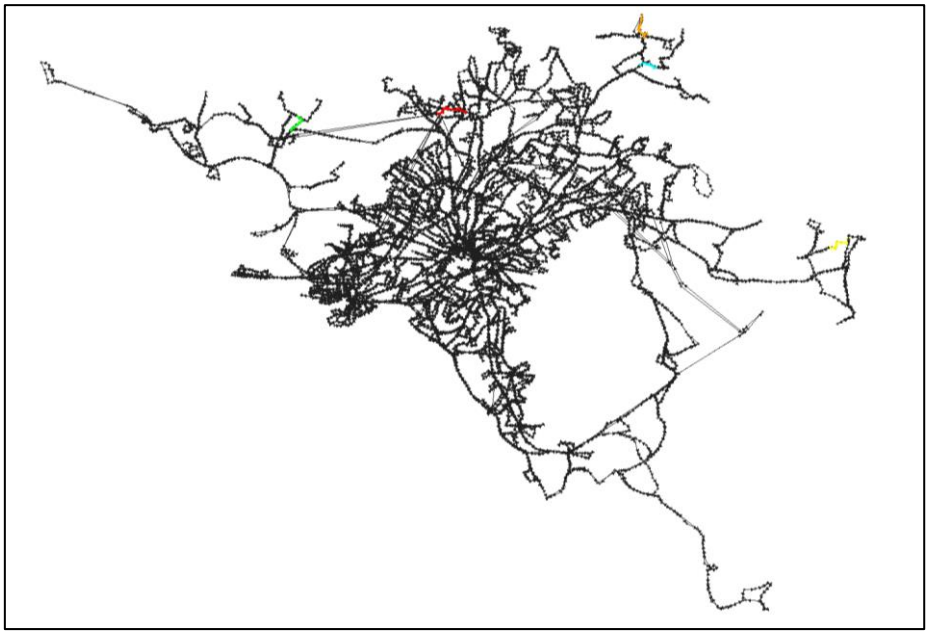


Figure 49 - Five segments (sequence of consecutive edges) with highest edge betweenness

Then, events are considered and new critical elements, induced by the new configuration of the affected UTS, are computed and compared with the previous ones, for every the measures previously considered.

The first event is to make unavailable the stop associated to the node with highest degree in the graph, but allowing for “jumping“ the stop, so avoiding interruption of the affected lines.

On the top of the following figure the new hubs, compared to the previous ones on the bottom.

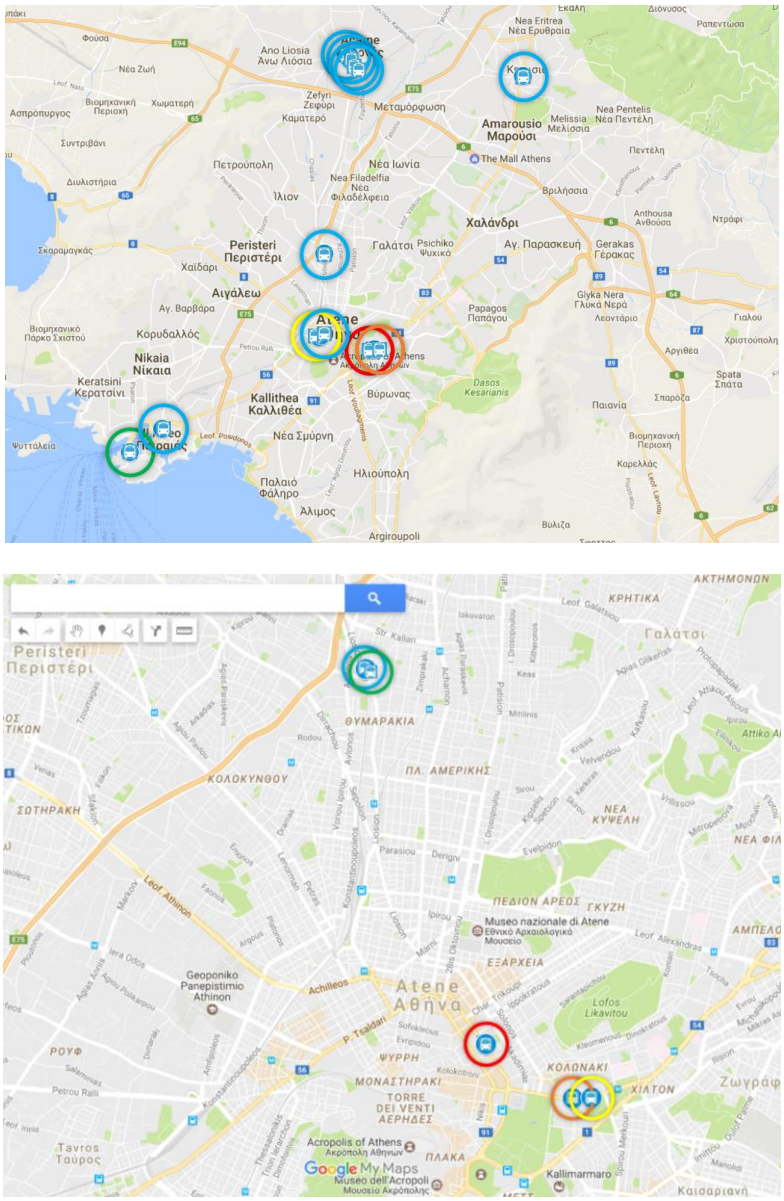


Figure 50 - Node degree: new (top) versus previous (bottom) hubs in the network

It is possible to see that the graph results significantly modified near the node with highest degree (before the event) and that its unavailability – when jumping stop is allowed – also modify relevance of nodes into the new setting.

The same type of event is then performed by considering the node with highest (node) betweenness. In this case modifications are less relevant and are more related to some changes in the ranking of the critical nodes.

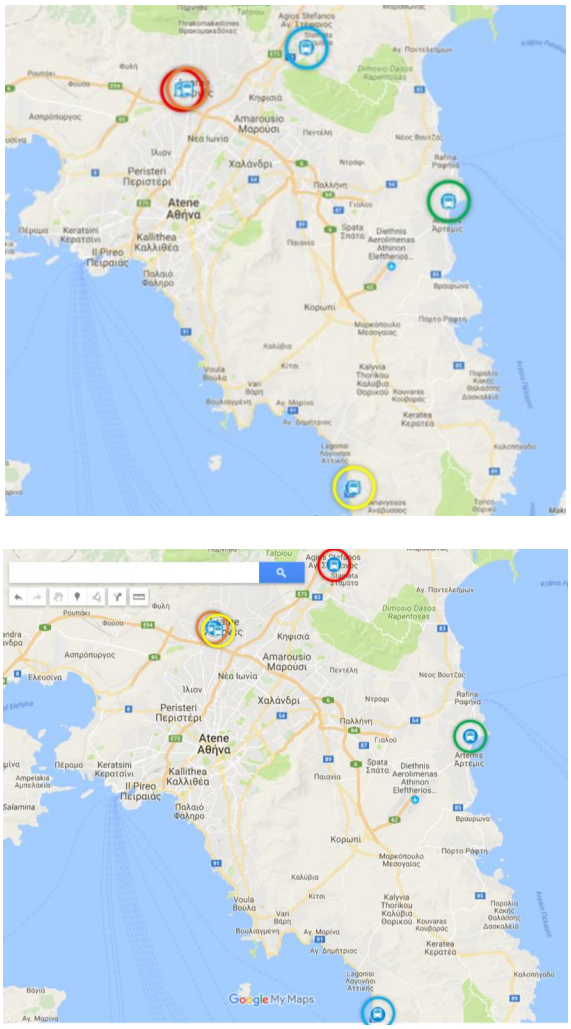


Figure 51 - Node Betweenness: new (top) versus previous (bottom) nodes in the network

Then the same event, but considering line interruption, is analysed. In this case it easy to understand that modifications on the graph are very significant, implying removal of a larger number of edges and, in case, nodes.

From the following figure it is easy to see that spatial distribution of hubs significantly changes with respect to the normal setting.

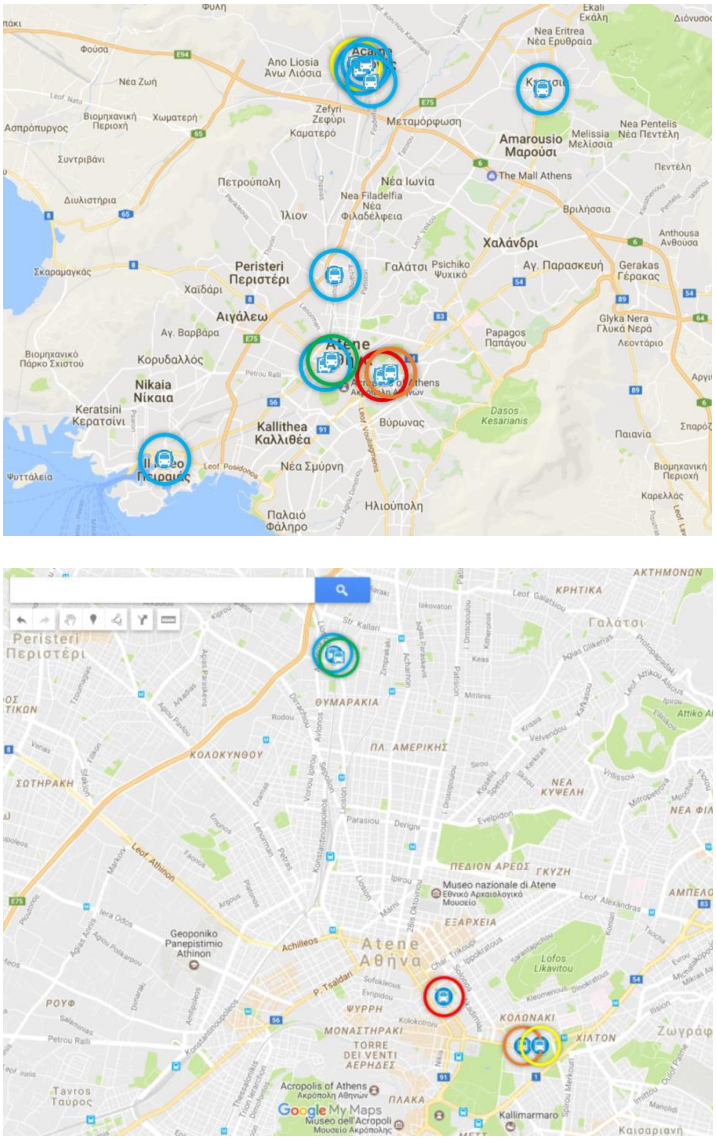


Figure 52 - Node Degree: new (top) versus previous (bottom) hubs in the network

The modifications induced on the graph are quite significant even in the case that the node with highest node betweenness is selected for removal. The spatial distribution of relevant nodes does not change so significantly with respect to the "jumping station" case.

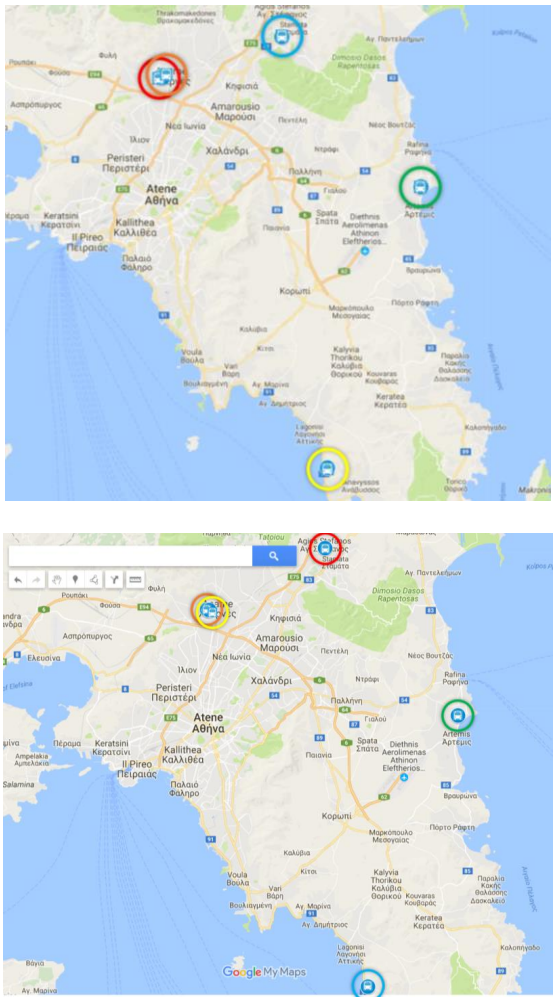


Figure 53 - Node Betweenness: new (upside) versus previous (downside) nodes in the network

Finally, modifications induced by interruption of lines passing through the edge with the highest value of edge betweenness are considered. Figure 54 shows as ranking of the segments changes and a new segment is now among the five most critical.

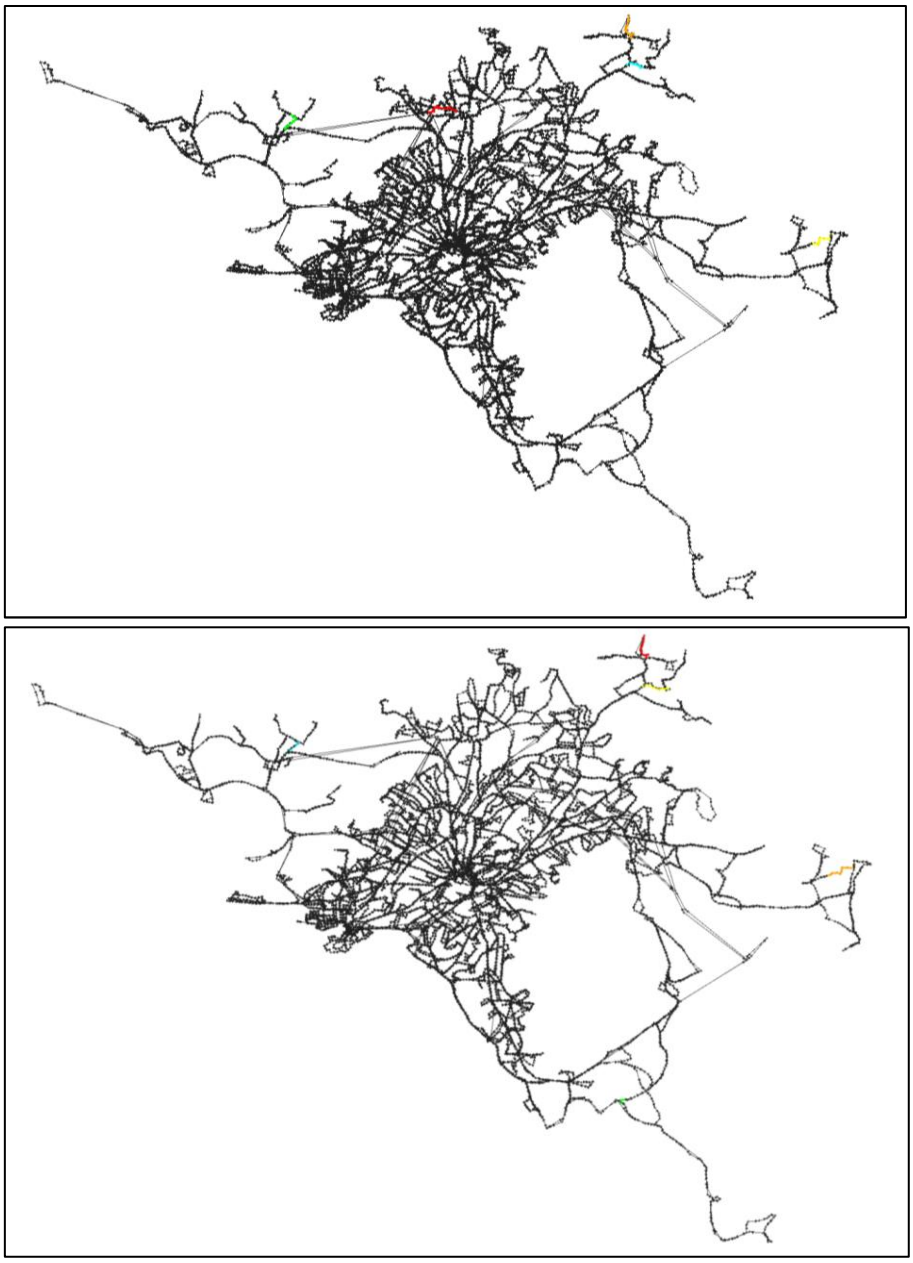


Figure 54 - Edge Betweenness: new (top) versus previous (bottom) edges in the network

Furthermore, network-wide measures from graph theory can be also computed for the "faultless" network and then compared to the new setting induced by the considered events. Due to the extension of the Attika region UTS, it is not possible to obtain – in a reasonable time – the measures associated to the spectral analysis of the adjacency matrix associated to the graph, in particular *algebraic connectivity*, *spectral gap* and *min cut set*. In the next activities related to the validation on the pilot sites, the analysis will be limited to the transportation line in the pilot, that is Athens, allowing also for the computation of these measures.

The table 10 summarizes the values of the measures computed on the original "faultless" Attika region UTS graph.

Table 10 – graph-based measures computed on the original UTS of the Attika region

Measure	Value	Note
Clustering coefficient	0.007	
Average Clustering coefficient	0.0106	Only GraphStream
Diameter	142	
Shortest Paths	58'959'363	
Characteristic path length	55.386	
average degree	4.72 (s.d. 3.86)	Only GraphStream
Density	0.001	in Cytoscape is "network centralization"
Algebraic connectivity	NA	Only igraph
Spectral gap	NA	Only igraph

After removing the node with the highest degree and allowing for "jumping" station, the measures change as reported in the Table 11:

Table 11 – measures after removing node with the highest degree and allowing for "jumping" station (Attika region)

Measure	Value	note
Clustering coefficient	**0.015**	*Changed*
Average Clustering coefficient	0.0106	
Diameter	142	
Shortest Paths	58'974'720	
Characteristic path length	**37.494**	*Changed*
average degree	4.71 (s.d. 3.83)	
Density	0.001	
Algebraic connectivity	NA	
Spectral gap	NA	

After removing the node with the highest degree and interrupting line(s) going from it, the measures change as reported in the Table 12:

Table 12 – measures after removing node with the highest degree and interrupting line(s) (Attika region)

Measure	Value	Note
Clustering coefficient	**0.015**	*Changed*
Average Clustering coefficient	**0.0107**	*Changed*
Diameter	142	
Shortest Paths	58'041'542	
Characteristic path length	**37.649**	*Changed*
average degree	4.63 (s.d. 3.82)	
Density	0.001	
Algebraic connectivity	NA	
Spectral gap	NA	

After removing the node with the highest node betweenness and allowing for "jumping" station, the measures change as reported in the Table 13:

Table 13 - measures after removing node with the highest node betweenness and allowing for "jumping" station (Attika region)

Measure	Value	note
Clustering coefficient	**0.015**	*Changed*
Average Clustering coefficient	0.0106	
Diameter	143	
Shortest Paths	58'974'720	
Characteristic path length	**37.492**	*Changed*
average degree	4.72 (s.d. 3.86)	
Density	0.001	
Algebraic connectivity	NA	
Spectral gap	NA	

After removing the node with the highest node betweenness and interrupting line(s), the measures change as reported in the Table 14:

Table 14 - measures after removing node with the highest node betweenness and interrupting line(s) (Attika region)

Measure	Value	Note
Clustering coefficient	**0.015**	*Changed*
Average Clustering coefficient	0.0106	
Diameter	143	
Shortest Paths	58'087'262	
Characteristic path length	**36.995**	*Changed*
average degree	4.68 (s.d. 3.86)	
Density	0.001	
Algebraic connectivity	NA	
Spectral gap	NA	

After removing the edge with the highest edge betweenness, the measures change as reported in the Table 15:

Table 15 - measures after removing edge with the highest edge betweenness (Attika region)

Measure	Value	Note
Clustering coefficient	**0.015**	*Changed*
Average Clustering coefficient	0.0106	
Diameter	142	
Shortest Paths	58'713'906	
Characteristic path length	**37.444**	*Changed*
average degree	4.72 (s.d. 3.86)	
Density	0.001	
Algebraic connectivity	NA	
Spectral gap	NA	

As it is possible to see from the previous tables, some network-wide measures do not modify depending on the events considered.

Again, it is important to highlight that all the above analyses can be performed also by weighting edges with information such as length, travelling time, number of travelling passengers, etc. These weights can be valued according to available data, in particular through statistics, real-time and/or simulation.

7.2.3 Cascading effects

Finally, this section presents the results on the simulation of cascading effects on the two different transportation networks: Florence and Attika region.

The algorithm is the one proposed in section 5.3. The choice was to consider the situation of "jumping station", that is a node is removed but lines passing through it are not interrupted. This just affects possible exchange options, requires the re-computation of the paths over the network and, most important, usually shows a slower convergence to the "equilibrium" with repsect to the "line interruption" scenario, because a lower number of links – and nodes – is removed at each iteration, increasing the probability that other nodes fail over a longer number of iterations.

The following figure shows how network average efficiency (*E*) changes with respect to different values of α. Contrary to what could be expected, an increase of the node capacity induces a decrease in the value of *E*. A small increase (5%) of the capacity of each station/station (node) induces a significant reduction of *E* with respect to its current value. This effect is less relevant if the capacity of each node is increased of 10%. Finally, *E* become stable – but lower than the current value – if the capacity of each station/stop is increased of 60% or more.

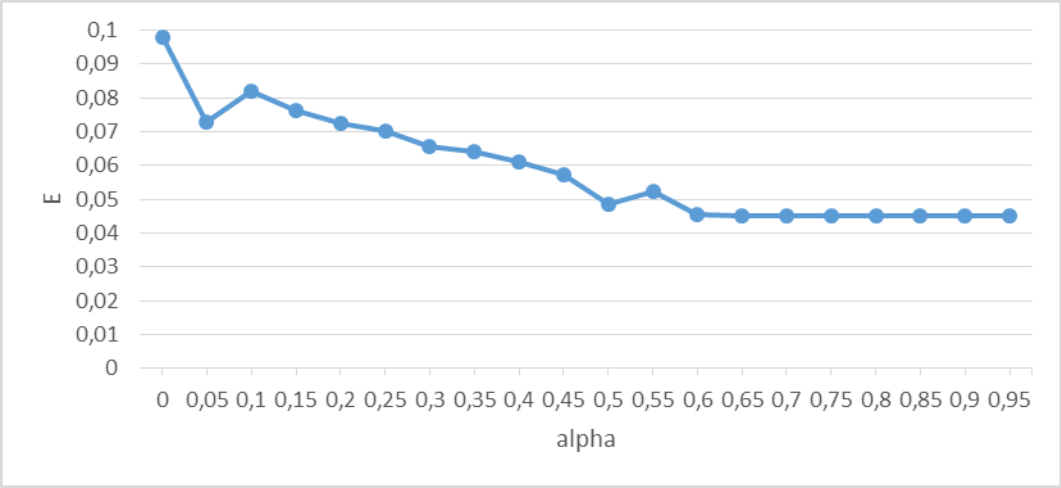


Figure 55 – Values of E depending on α for the Florence UTS

As expected, the relative size (S) of the GCC increases with α. A significant increase is observed if the capacity of each node would be increased of the 10% with respect to the current value. Finally, *S* become stable when α is 60% or higher.

In conclusion, *E* and *S* result to be "antagonistic" with respect to the Florence UTS: so it is impossible to improve one of them – in particular *S* – without worsening the other one – in particular *E*.

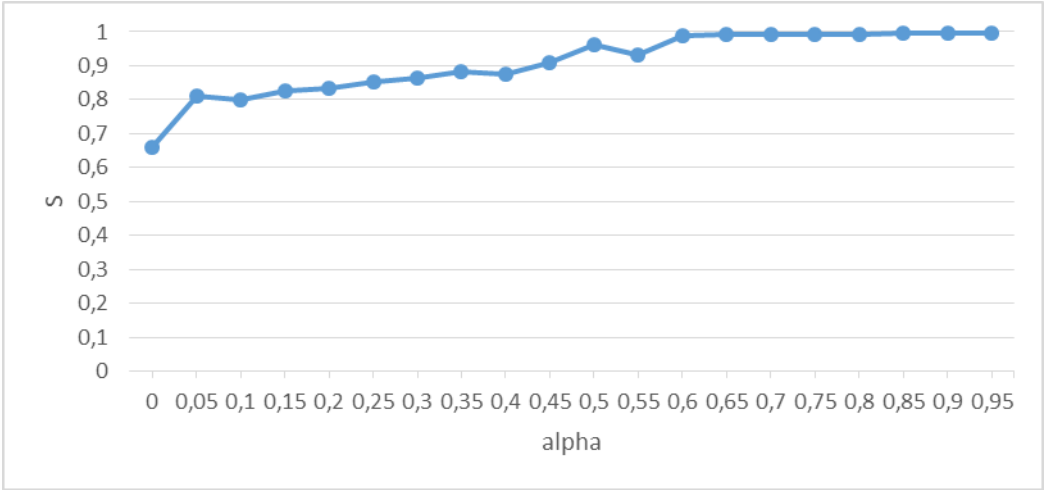


Figure 56 – Values of S depending on α for the Florence UTS

Quite similar results have been obtained for the UTS of the Attika region. Again *E* decreases with α increasing. Furthermore, values of *E* are quite lower when compared those computed on the Florence UTS (this could be due to the different size of the two network). The results for *E* are presented in the following figure.

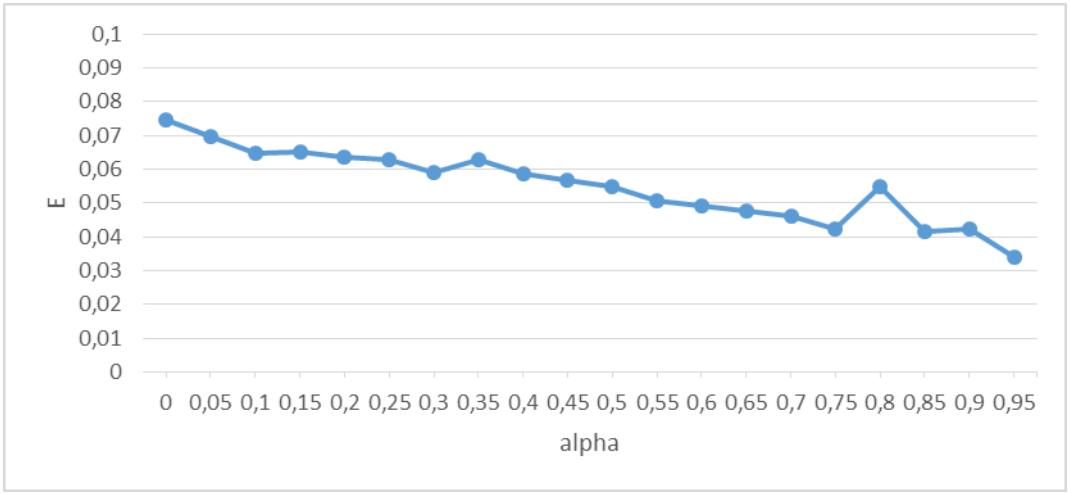


Figure 57 – Values of E depending on α for the Attika region UTS

Even in the case of the UTS of the Attika Region the value of *S* increases with α increasing. Similarly to *E*, values of *S* result lower than those obtained on the Florence UTS.

Again the *E* and *S* measures result "antagonistic", as observed for the Florence UTS. There is anyway a substantial difference, when $\alpha \geq 60\%$ values of *E* and *S* are stable for the Florence UTS, while stability is not reached for the UTS of the Attika region (in particular *E* continues to decrease).

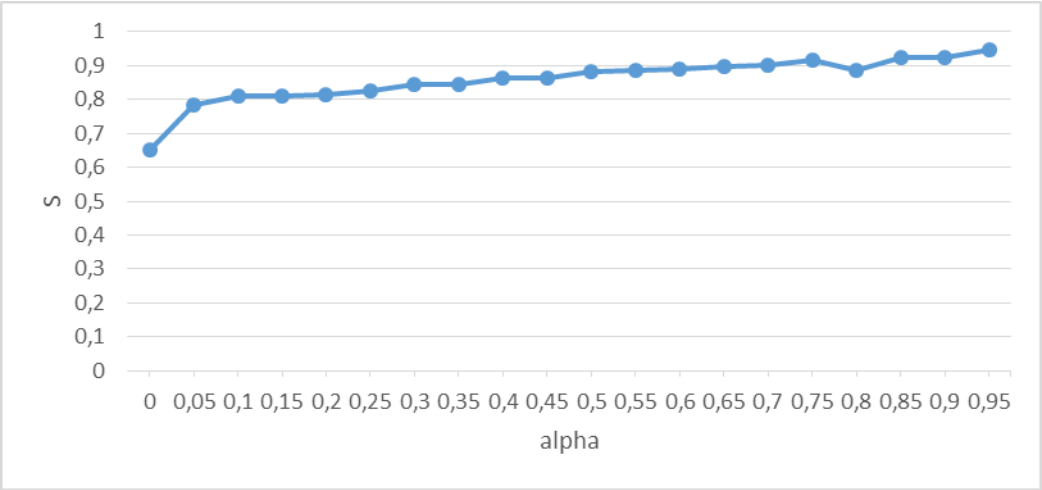


Figure 58 – Values of S depending on α for the Attika region UTS

This analysis permits to better understand how the UTS behaves under a cascading effect situation and may also support in planning capacity of nodes (stop/stations) in order to better respond to the cascade.

It is important to highlight that this application to the two public transportation networks is well different from the use case reported in (Zou et al., 2013) where a road network is considered. Furthermore, in the paper edges and nodes are removed "locally" creating disconnections in a meshed graph while the multi-graphs associated to the two public transportation networks are more planar and, in the case of "jumping station", disconnections are less probable. Therefore, the evolution of *E* and *S* in these networks are significantly different from the one reported in the paper.

8 CONCLUSIONS

The Application Framework, which is part of the back-end architecture of the RESOLUTE project, has been presented.

Some datasets, available in RESOLUTE as well as accessed from open-data initiatives, have been used to perform the design, development and preliminary testing of the user profiling and network analysis modules of the Application Framework.

State of the art methodologies and tools have been investigated, and development activities have been focused on the integration of different libraries and their extension, in particular with respect to the possibility to use them to perform online analysis by dynamically bringing changes (e.g. nodes and edges removal from the graph associated to the UTS.

Some preliminary results have also been provided, in order to present evidence of the benefits provided by this analytical layer of the RESOLUTE platform so as to build decision support functionalities.

The users profiling procedure and its derived knowledge are of great significance and can be used in case of emergencies. The described method can not only increase the resilience by seeing it conceptually within the CRAMSS, but also may have a vital role in many of the emergency phases. The experimental results with the synthetic dataset demonstrate that the three mobility behavior features defined hereby are capable of discriminating between different mobility profiles of the agents. Furthermore, the numerical nature of these features allows for a visual inspection of the agents and the paths they take as points on the screen, which facilitates significantly in discovering any behavioral patterns among the agents.

Moreover, the combination of the defined features using the multi-objective visualization method allows information from all features to be exploited, producing visualizations where the agents with mobility issues are visually separated from those with no mobility issues, forming distinct visual groups. Each of these groups corresponds to a different type of agent mobility, e.g. agents who generally follow a limited set of paths, as opposed to agents who generally follow a wide variety of path shapes and lengths. Such distinct behavioral groups can be used to form more comprehensive mobility profiles for the agents, which can be further exploited by the resilience management system (CRAMSS) and more specifically by the evacuation decision support system (eDSS).

In the future, it would be interesting to verify the applicability of the above types of features and procedures in real data collected from actual paths of various types of agents in a city. However, the presented methods are not restricted to these three types of features. Application with real data could allow the definition of more types of features, which may be more appropriate for real-world scenarios.

With respect to (UTS's) user profiling, the experimental results obtained on the open-data related to the passenger counts at the Transport for London Underground stations, have demonstrated that the time series clustering approach proposed is able to identify typical as well as unusual patterns (i.e. half an hour passenger counts time series over the day). Unusual patterns are in particular relevant because they might be associated to disruptions and/or events affecting one or more stations. Observing the composition of the clusters containing unusual patterns allows for the identification of relations among stations (i.e. those who are associated to an anomalous pattern for the same day) and temporal correlations between extreme peaks/bursts occurring at specific time(s) over the day. It will be interesting to validate the approach on real data collected at the pilots. The most important contribution of

this functionality is to better understand – and categorize – the behaviour of the UTS users, even according to specific conditions of the system as well as under specific event (e.g. delays, disruptions, etc.). Such an information can effectively support planning activities and also more strategic decisions, even during emergency, taking into account modifications into the flows of passengers, in particular through stations/lines.

With respect to network analysis, the experimental results obtained on the two public transportation networks of the RESOLUTE project (where Florence UTS data are retrieved from the RESOLUTE Knowledge Base, while Attika region UTS have been accessed via open APIs) have demonstrated that a set of analytical functionalities can be used to identify critical components (nodes as well as edges) and evaluate, dynamically, the new setting induced by a disruption event as well as the simulation of cascading effects. The capability to interact with and analyse graphs, dynamically and efficiently, is a core contribution in order to support decision making activities in the UTS resilience management. Future investigations and validations will be focused on the extensions of these functionalities on other networks (e.g. the road networks of the pilot sites) and by also using real-time data, enriching the graph-based models used for the analysis (e.g. by weighting edges with travel time or number of passengers, etc.). The specific computational module will be able to dynamically analyse the graph associated to the UTS, even under changing conditions, and to identify the new critical components into the network over time. Although this functionality is really important to support decisions in the "anticipate" and "learn" phases, the information it provides could be potentially useful to support also the "monitor" and "respond" phases. The RESOLUTE user can also merge the information about critical nodes and edges with the output of the computational module devoted to the weather severity monitoring and associated flood hazard, to identify potentially risky areas and, in case, the critical nodes or links belonging to that areas. Finally, an analysis of cascading failures have been implemented and preliminary validated on the two UTS. The main output is a series of two measures, namely the average network efficiency (*E*) and the relative size of the GCC (*S*), with respect to the capacity of nodes (i.e. stops/stations) and when the triggering failure is associated to the removal of the node with highest betweenness. The re-computation of the load for each node (i.e. the betweenness) permits to identify the new failing nodes in the cascade (i.e. nodes with capacity lower than the current load). These nodes are removed and the process iterated until no more nodes fail. The functionality has been evaluated following the aforementioned schema and proved to be useful in the "anticipate" and "learn" phases to support more effective planning strategies. However, it may also be used to evaluate cascading effects in "real" situations, where the triggering event can be the removal of one – or even more – node different from the one having highest betweenness and where the load is computed according to some weights on the edges of the graph, for instance the number of lines, vehicles or passengers passing through them.

9 REFERENCES

Alfieri, L., Thielen, J. (2015), A European precipitation index for extreme rain-storm and flash flood, Meteorological Applications, 3-13.

Andrienko, G., Andrienko, N., Rinzivillo, S., Nanni, M., Pedreschi, D., Giannotti, F. (2009) Interactive visual clustering of large collections of trajectories, in: IEEE VAST, 2009, pp. 3–10.

Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez J, Perona I. An extensive comparative study of cluster validity indices, Pattern Recognition, 46 (1) (2013) 243-256.

Backstrom L., Boldi P., Rosa M., Ugander J., Vigna S., (2012), Four Degrees of Separation, Proceedings of the 3rd Annual ACM Web Science Conference, 33-42.

Barabási A.L., (2003), Linked: How Everything is Connected to Everything Else and What it Means for Business, Science, and Everyday Life, Plume, New York.

Bellini E., Paolucci M., Nesi P. D4.2 – Multi source data acquisition (RESOLUTE deliverable)

Black W.R., (2003), Transportation: A Geographical Analysis, Guildford Press, New York.

Buhl J., Gautrais J., Reeves N., Sole R.V., Valverde S., Kuntz P., Theraulaz G., (2006), Topological patterns in street networks of self-organized urban settlements, The European Physical Journal B, 49, 513-522.

Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., and Ratti, C. Real-time urban monitoring using cellular phones: a case-study in rome. IEEE Transactions on Intelligent Transportation Systems (2010).

Candelieri, A., Archetti, F. Detecting events and sentiment on twitter for improving urban mobility (2015a) CEUR Workshop Proceedings, 1351, 106-115;

Candelieri, A., Soldi, D., Archetti, F. (2015b) Short-term forecasting of hourly water consumption by using automatic metering readers data, Procedia Engineering, 119 (1), pp. 844-853.

Candelieri, A., Archetti, F. Analyzing tweets to enable sustainable, multi-modal and personalized urban mobility: Approaches and results from the Italian project TAM-TAM (2014a) WIT Transactions on the Built Environment, 138, pp. 373-379.

Candelieri, A., Archetti, F. (2014b) .Identifying typical urban water demand patterns for a reliable short-term forecasting - The icewater project approach, Procedia Engineering, 89, pp. 1004-1012.

Chen, C. C., Kuo, C. H., and Peng, W. C. Mining spatial-temporal semantic trajectory patterns from raw trajectories. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (Nov 2015), pp. 1019-1024.

Cox, T. F., & Cox, M. A. (2000). Multidimensional scaling. CRC press

D'Lima, M., Medda, F. A new measure of resilience: An application to the London Underground, Transportation Research Part A 81 (2015), 35-46.

Dehghani, M.S., Flintsch, G., McNeil, S. (2014). Impact of road conditions and disruption uncertainties on network vulnerability. J. Infrastruct. Syst. 20, 04014015.

Eiter, T., & Mannila, H. (1994). Computing discrete Fréchet distance. Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna.

Ferreira P, Simões A. D2.2 - Conceptual framework (RESOLUTE deliverable)

Faturechi, R., Miller-Hooks, E. (2014) Measuring the performance of transportation infrastructure systems in disaster: a comprehensive review. J. Infrastruct. Syst., 04014025-1.

Fiedler M., (1973), Algebraic connectivity of graphs, Czechoslovak Mathematical Journal, 23, 298–305.

Gaitanidou E, Bellini E, Ferreira P. D3.5 – European Resilience Management Guidelines (RESOLUTE deliverable)

Gaitanidou E., Tsami M. D3.7 – ERMG adaptation to UTS (RESOLUTE deliverable)

Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., Renso, C., Rinzivillo, S., and Trasarti, R. Unveiling the complexity of human mobility by querying and mining massive trajectory data. Very Large Database 20, 5 (2011).

Giannotti, F., Pedreschi, D. Mobility, Data Mining and Privacy, Springer, 2008.

Gehl, P., Wang, M., Taalab, K., D'Ayala, D., Medda, F., et al. (2016) Use of multi-hazard fragility functions for the multi-risk assessment of road networks, 1st International Conference on Natural Hazards & Infrastructure: INCONHIC 2016, Jun 2016, La Canée, Greece. Conference proceedings, 2016, 1st International Conference on Natural Hazards & Infrastructure Conference proceedings

Girvan M., Newman M.E.J, (2002), Community Structure in Social and Biological Networks, Proceedings of National Academy Science, USA, 99, 7821–7826.

Grifoni A., Costantini C., Cigheri S. D4.5 – Integration Framework Implementation (RESOLUTE deliverable)

Hémond, Y., Robert, B. (2010) Evaluation of the consequences of road system failure on other critical infrastructures. Int. J. Crit. Infrastruct. 6, 1–16.

Hsieh, C.-H., Feng, C.-M., 2014. Road network vulnerability assessment based on fragile factor interdependencies in spatial-functional perspectives. Environ. Plan. A 46, 700–714.

Jenelius, E., Koutsopoulos, H.N. (2013) Travel time estimation for urban road networks using low frequency probe vehicle data. Transp. Res. Part B 53, 64–81.

Jenelius, E., Mattsson, L.-G. (2012) Road network vulnerability analysis of area-covering disruptions: a grid-based approach with case study. Transp. Res. Part A 46, 746–760.

Jin, J.G., Teo, K.M., Odoni, A.R. (2015) Optimizing Bus Bridging Services in Response to Disruptions of Urban Transit Rail Networks. Transportation Science.

Johansson, J., Hassel, H., Cedergren, A., 2011. Vulnerability analysis of interdependent critical infrastructures: case study of the Swedish railway system. Int. J. Crit. Infrastruct. 7, 289–316.

Kalayathankal, S., Singh, G., 2010. A fuzzy soft flood alarm model. Mathematics and Computers in Simulation. 887-893.

Kepaptsoglou, K., Karlaftis, M.G. (2009) The bus bridging problem in metro operations: conceptual framework, models and algorithms, Public Transp 1: 275–297.

Khademi, N., Balaei, B., Shahri, M., Mirzaei, M., Sarrafi, B., Zahabiun, M., Mohaymany, A. (2015) Transportion network vulnerability analysis for the case of a catastrophic earthquake. Int. J. Disaster Risk Reduct. 12, 234–254.

Lang, B., Poppe, T., Minin, A., Mokhov, I., Kuperin, Y., Mekler, A., Liapakina, I. (2008) Neural Clouds for Monitoring of Complex Systems. Optical Memory and Neural Networks. 183-192.

Liao, L. C., Jiang, X. H., Zou, F. M., Tsai, P. W., and Deng, Y. L. A method of latent semantic information mining for trajectory data. In 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP) (Sept 2015), pp. 353-353.

Luxburg U., (2007), A Tutorial on Spectral Clustering, Statistics and Computing, 17, 1-32.

Mattsson, L.G., Jenelius, E. (2015) Vulnerability and resilience of transport systems – A discussion of recent research, Transportation Research Part A 81 (2015) 16–34.

Nanni, M., and Pedreschi, D. Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems 27, 3 (Nov. 2006), 267-289.

Ng, A. Y., Jordan, M. & Weiss, Y. (2001) On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. 14, 849–856.

Otsuka, T., Torii, Y., Ito, T. Anomaly detection algorithm for localized abnormal weather using low-cost wireless sensor nodes (2014). 7th IEEE International Conference on Service-Oriented Computing and Applications.

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J. A., Pelekis, N., Theodoridis, Y., and Yan, Z. Semantic trajectories modeling and analysis. ACM Computing Surveys 45, 4 (2013).

Pelekis, N., Kopanakis, I., Kotsifakos, E., Frentzos, E., Theodoridis, Y. (2009) Clustering trajectories of moving objects in an uncertain world, in: ICDM, 2009, pp. 417–427.

Reggiani, A. (2013) Network resilience for transport security: some methodological considerations. Transp. Policy 28, 63–68.

Reggiani, A., Nijkamp, P., Lanzi, D. (2015) Transport resilience and vulnerability: the role of connectivity. Transp. Res. Part A 81, 4–15.

Renso, C., Spaccapietra, S., and Zimanyi, E., Eds. Mobility Data: Modeling, Management, and Understanding. Cambridge University Press, 2013.

Schaeffer S.E., (2007), Graph Clustering (survey), Computer Science Review, 1, 27-64.

Shi, J. & Malik, J. 2000 Normalized cuts and image segmentation. IEEE T. Pattern Anal. 22, 888–905.

SONG, X.; ZHANG, Q.; SEKIMOTO, Y.; SHIBASAKI, R.; YUAN, N.; XIE, X.. A Simulator of Human Emergency Mobility Following Disasters: Knowledge Transfer from Big Disaster Data. AAAI Conference on Artificial Intelligence, North America, 2015

Trasarti, R., Rinzivillo, S., Pinelli, F., Nanni, M., Monreale, A., Renso, C., Pedreschi, D., and Giannotti, F. Exploring real mobility data with M-atlas. Machine Learning and Knowledge Discovery in Databases (2010), 624-627.

Vugrin ED, Warren DE, Ehlen MA, and Camphouse RC, A Framework for Assessing the Resilience of Infrastructure and Economic Systems, 2010.

Wachowicz, M., Ong, R., Renso, C., and Nanni, M. Finding moving flock patterns among pedestrians through collective coherence. IJGIS 25, 11 (2011).

Wang Y., Guo J., Currie G., Ceder A.A., Dong D. Brendan Pender "Bus Bridging Disruption in Rail Services With Frustrated and Impatient Passengers", IEEE Transactions on Intelligent Transportation Systems, 15(5), 2014 – 2023, 2015

Wang Y, Zheng Y, and Xue Y. (2014). Travel time estimation of a path using sparse trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 25–34.

Watling, D., Balijepalli, N.C. (2012) A method to assess demand growth vulnerability of travel times on road network links. Transp. Res. Part A 46, 772–789.

Wilson DT, Hawe GI, Coates G, Crouch RS (2013), Scheduling response operations under transport network disruptions, Proceedings of 10th International Conference on Information Systems for Crisis Response and Management

Worton, K.E. (2012) Using socio-technical resilience frameworks to anticipate threat. In: 2012 Workshop on Socio-Technical Aspects on Security and Thrust (STAST), pp. 19–26.

Xiao, X., Zheng, Y., Luo, Q., and Xie, X. (2010). Finding similar users using category-based location history. In Proceedings of the 18th Annual ACM International Conference on Advances in Geographic Information Systems. ACM, 442–445.

Yazdani A., Jeffrey P., (2012), Water distribution system vulnerability analysis using weighted and directed network models, Water Resources Research, 48.

Yuan J., Zheng Y., Zhang L., Xie X., and Sun G. (2011b). Where to find my next passenger? In Proceedings of the 13th International Conference on Ubiquitous Computing. ACM, 109–118.

Yuan J, Zheng Y, Xie X, and Sun G. (2013a). T-Drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Transaction on Knowledge and Data Engineering 25, 1 (2013), 220–232.

Yuan NJ, Zheng Y, Zhang L, and Xie X. (2013b). T-Finder: A recommender system for finding passengers and vacant taxis. IEEE Transaction on Knowledge and Data Engineering 25, 10 (2013), 2390–2403.

Zhang, J., Song, B., Zhang, Z., Liu, H., 2014. An approach for modeling vulnerability of the network of networks. Physica A 412, 127–136.

Zheng, Y. (2015) Trajectory Data Mining: an Overview, ACM Transactions on Intelligent Systems and Technology, 6(3) Article 29

Zheng Y, Liu Y, Yuan J, and Xie X. (2011a). Urban computing with taxicabs. In Proceedings of the 13th International Conference on Ubiquitous Computing. ACM, 89–98.

Zheng, Y., and Xie, X. Learning Location Correlation from GPS Trajectories. 2010 Eleventh International Conference on Mobile Data Management, 49 (2010), 27-32.

Zimmerman R., (2004), Decision-making and the vulnerability of interdependent critical infrastructures, CREATE Report, Center for Risk and Economic Analysis of Terrorism Events, University of Southern California, Los Angeles, California.

Zimmerman R., Restrepo C.E., (2006), The next step: quantifying infrastructure interdependencies to improve security, International Journal of Critical Infrastructures, 2, 215-230.

Zou, Z., Xiao, Y., & Gao, J. (2013). Robustness analysis of urban transit network based on complex networks theory. Kybernetes, 42(3), 383-399.