Telecommunication networks of Milan: analyzing its representation as a time evolving network

Aurel Prósz[1] *

[1]ELTE TTK, Scientific Modelling Computer Lab, 16.05.2019

A lot of real world systems in nature, like society, the Internet and many biological phenomenons can be described as a network, which evolves in time. By modelling the structure of these one can understand, predict, and optimize the behaviour of dynamic systems. My aim with this project is to study these kind of networks by first analyzing smaller toy networks, then reconstructing the telecommunication network of the city of Milan. The resulting model is a temporal, undirected, weighted network, with a time resolution of 100 milliseconds. During my project I successfully identified periodic events in the daily and weekly periods of communication exchange between the provinces and located outlier events. I also experimented with 3D visualization techniques to present my results in a different, and more meaningful way.

Network Science | Network Dynamics | R programming | Advanced data visualization

Introduction. Almost all real world networks are evolving in time, since usually they are constructed from a smaller to a bigger network. This evolution can be manifested as adding or removing nodes or links over time. Common examples of time evolving networks are social and communication networks, the internet, and transportation networks. It is increasingly important to study time evolving networks, because they represents many crucial parts of our lives.

My project aims to study time evolving networks on real life-examples, like the time dependent weighted network of mobile phone calls in northern Italy. This article covers the whole process of my project during the first and second term of the laboratory. In the first part I introduce the tool sets and datasets I used to obtain my results. In the second part I present the data cleaning and visualizations of several dynamic networks. In the last part I show the results of the temporal simulations on the networks. Last, but least I show the first results of 3D visualization of scatter plots which are used for deeper data understanding.

Note that I use the word network and graph, node and vertices interchangeably. (1)(2)

In Table 1 I present my work and progress plan, and in Figure 1 the time schedule of the project.

Personal motivation and my contribution to the topic. During my studies I encountered problems and systems which could be represented as a static network to study its properties better. However, in certain cases there was a need to infer how this network would evolve in time, and how would this evolution change its attributes and what could it mean for the underlying process as well. A problem I recently encountered involves gene-gene interaction networks, which could potentially evolve in time. I expect after I finish my report I acquire the necessary tools and means to tackle this problem. During my project I developed an end-to-end data processing and visualization workflow, which means I downloaded the raw data, pre-processed it, did the simulations and statistical inferring, and also the visualization and interpretation. My work could be considered novel in a way, that I did not find any previous dynamic animation of the time evolution of the mentioned Milan network, and the planned 3D visualization is also a underrepresented topic in network science.

Materials and methods. To study the attributes of time evolving networks I explored the following data repositories:

- The Colorado Index of Complex Networks (ICON) (3)
- OpenConnectome (4)
- Netflix dataset (5)
- DIMES (6)
- GEO (7)
- Mobile phone calls in northern Italy (8)

From these sources I selected the following networks to study their details further:

- **Collaborations between French Gothic illuminated manuscript workshops between 1260 and 1320.**: Between 1260 and 1320, a distinctly French approach to manuscript illumination reached its pinnacle in works produced for royal and aristocratic patrons in Paris. Scholarship on these manuscripts and the workshops that produced them, however, has not yet fully contextualized them with regard to contemporaneous developments on the peripheries of the region. This dataset describes the interaction between the workshops. (9)
- **Hypertext 2009 (conference) dynamic contact network**: This dataset was collected during the ACM Hypertext 2009 conference, where the SocioPatterns project deployed the Live Social Semantics application. Conference attendees volunteered to wear radio badges that monitored their face-to-face proximity.

The dataset published here represents the dynamical network of face-to-face proximity of 110 conference attendees over about 2.5 days. (10)

- **Media network**: A network consisting of collaborations of media players in New York in time. It is an useful network to study edge weights in time.(11)

- **E.coli transcription network**: Transcriptional regulation networks in cells orchestrate gene expression. In this network the nodes are operons, and each edge is directed from an operon that encodes a transcription factor to an operon that it directly regulates. The data was generated from the *Escherichia coli* bacteria. (12)

- **Milan phone calls dataset**: This dataset provides information regarding the directional interaction strength between the city of Milan different areas based on the calls exchanged between Telecom Italia Mobile users. The directional interaction strength between the area A and the area B is proportional to the number of calls issued from the area A to the area B. (8)

During the analysis I used two tools: Gephi and an R package called NetworkDynamics. They both offer a simple way to analyse and visualize the temporal networks, which could be harder using the packages of Python.

Details of the multi-source dataset of urban life in the city of Milan. In the recent years the almost universal adoption of mobile phone services exponentially increased the data generated by its users and this data can be used to provide new insights into socio-technical systems. There is an emerging field of new research areas which use these huge datasets to extract and analyze human mobility patterns, social interactions, predicts socio-economic indicators, and the spreading of diseases.
In this project I used the telecommunications interactions data regarding only the city of Milan. The city is divided into 1000 squares of about 235 × 235 meters, and the call intensity data is aggregated in these patches. This grid is projected with the WGS84 (EPSG:4326) standard.
The call intensity values are composed of the incoming and outgoing calls of Telecom Italia's customers, which corresponds of only the 34% of all customers in Milan. Also the missed calls are not represented in this dataset. The data has been collected over two months, from November 1st, 2013 to January 1st, 2014. This time interval contains two important events: the Christmas holiday, and the New Years Eve, which I hypothesize could be observed in the data.
The data is structured in the following way:

- **Time interval:** The beginning of the time interval expressed as the number of millisecond elapsed from the Unix Epoch on January 1st, 1970 at UTC. The end of the time interval can be obtained by adding 600000 milliseconds (10 minutes) to this value.

- **Square id1:** the id of the square of the Milano GRID which is the origin of the interaction.

- **Square id2:** the id of the square of the Milano GRID which is the destination of the interaction.

- **Directional Interaction Strength:** the value representing the directional interaction strength between the Square id1 and the Square id2. This value is proportional to the number of calls exchanged between callers located in the Square id1 and receivers located in the Square id2.

Due to the very large volume of the dataset (over 80 GB), I randomly sampled 50 squares from the total of 1000 squares, representing the 5% of the total data for my biggest analysis.

Network layout and centrality measures. The Milan telecommunication data can be represented as a network, where the squares of the Milano GRID correspond to the nodes, and the Directional Interaction Strength to the edges and weights. To make the analysis more simple I omitted the edge weight information, but handled the network as a directional one.
I visualized the networks using the so called force-directed layout algorithms, which are graph drawing algorithms, based on only the graph structure itself. The most straightforward force-directed algorithm uses repulsive forces between nodes and attractive forces between adjacent nodes. This is a physical model, and it converges into a certain energy minima, which makes the layout of the network visually pleasing. These kind of algorithms are good if one does not have any prior knowledge about the logical layout of a graph, as in my case (13)(14).
To embed a graph with the simplest force-directed layout algorithm we replace the edges with springs, and the vertices with rings. Then the vertices are placed in an initial layout. Following the 1989 algorithm of Kamada and Kawai there are no separate attractive and repulsive forces between pairs of vertices, but instead if a pair of vertices is (geometrically) closer/farther than their corresponding graph distance the vertices repel/attract each other.
According to this model the ideal length of a spring between vertices i and j is $l_{i,j} = Lxd_{i,j}$, where $d_{i,j}$ corresponds the shortest path distance between vertex i and vertex j and L is the desirable length of a single edge in the display. Kamada and Kawai suggest that $L = \frac{L_0}{max_{i<j} d_{i,j}}$ where L_0 corresponds to the length of a side of the display area and $max_{i<j} d_{i,j}$ is the diameter of the graph. The strength is then defined as

$$k_{i,j} = \frac{K}{d_{i,j}^2}$$

here K is a constant.
In this way, by treating the problem as localizing p_n particles in 2D Euclidean space, we can write the energy function, what we aim to minimize:

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} k_{i,j} \left(|p_i - p_j| - l_{i,j}\right)^2$$

The algorithm solves this problem in $O\left(n^3\right)$ time and requires $O\left(n^2\right)$ storage for pairwise vertex distances. Despite

these drawbacks the algorithm still produces a simple and intuitive definition of a "good" graph layout: A graph layout is good if the geometric distances between vertices closely correspond to the underlying graph distances.
To quantify the structural attributes and its changes of the network in time, I calculated different centrality measures to describe the higher level structures, and the density of the network. In my analysis I calculated these measures to each nodes, and then normalized all of them with the theoretically maximal value. This technique can be summarized with the following equation:

$$C^*(G) = \sum_{i \in V(G))} \left| \underset{v \in V(G))}{max}(C(v)) - C(i) \right|$$

where $C^*(G)$ is the value of the centrality measure on the graph G, and $C(v)$ is the measure calculated at each node.
Here I shortly describe each of them. Note that there are many more centrality scores, and I selected only the most common ones (15) (16).
Betweenness centrality: Betweenness centrality is basically the measure of the extent to which a given node lies on paths between other vertices. It can be defined as:

$$b_i = \sum_{s,t} w^i_{s,t} = \sum_{s,t} \frac{v^i_{s,t}}{v_{s,t}}$$

where $v_{s,t}$ is the total number of shortest paths from node s to node t and $v_{s,t}$ is the number of those paths that pass through v.
Degree centrality: The degree centrality measures how each node is influential in such a way, that how many edges they have. Let $deg(v)$ be the degree of a node, and $H = n^2 - 3n + 2$ be the connected graph that maximizes the following quantity (with $v*$ being the node with highest degree centrality in the original network. Then the degree centrality of graph G corresponds to:

$$C(G) = \sum_{i=1}^{n} \frac{C(v*) - C(v_i)}{H}$$

Note that this form of the equation is already normalized to the maximal value.
Closeness centrality: Closeness centrality is calculated by the sum of the length of the shortest paths between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes. Let $d_{i,j}$ the distance between two nodes, N the total number of nodes, then the closeness centrality of a node is:

$$C(v_i) = \frac{N}{\sum_j d_{i,j}}$$

Eigenvector centrality: This centrality measure determines how influential a node is in the network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Given the adjacency matrix of the network, which is $A = a(i,j)$, the relative centrality score of vertex v can be written as

$$x_v = \frac{1}{\lambda} \sum_j t \in M(v) x_j = \frac{1}{\lambda} \sum_j t \in G A x_j$$

Then with rearrangement we can write it as an eigenvector equation:

$$Ax = \lambda x$$

The eigenvector which belongs to the greatest eigenvalue will give the centrality scores for each nodes, where the v^{th} component of the eigenvector corresponds to the relative centrality score of the vertex v in the network.

Edge formation rate. A simple metric which quantifies how many new edges are added to the network in a given Δt time. One would assume that during prime hours, or during a certain localized event in the city this measure would be high.

Sankey diagram. In my project I visualized the Sankey diagram of the temporal networks as a tool to visualize the proximity of each node during the time evolution defined by the geodesic distance between the nodes. The geodesic distance is defined by the distance between two nodes in a network which is the number of edges in a shortest path connecting them.

Exploring 3D visualization tools. There are multiple arguments supporting true 3D visualization of a dataset, including the fact, that this way we can observe patterns in higher dimensions, than we could when we analyze data in lower one (for example a 2D scatter plot versus a 3D).
To create a true 3D image, one must render a view of an object from two slightly different perspective, similar how the human eyes see the world from a Δd distance from each other. If one can render a scene with this exact parameter, the obtained two image can be converted to a 3D picture or movie formatted as the following to approaches(17)(18)(19):

- **Anaglyph 3D:** It achieves a 3D effect by color coding the image for the right and the left eye (usually cyan and red), and the stereoscopic image can be perceived by a similarly colored glasses.
- **Active shutter 3D system:** This technology works in a way, that special "active" glasses periodically blocks the view of one of the viewers eyes, while the display screen shows only the part of the video for the corresponding eye. This periodic blocking effect is so fast, it does not effect the viewers experience.

Results

Installing software and tools. In the beginning of my project I had to make sure that I choose the right tools for further analysis, and to be able to exchange data between them. Therefore I chose a medium-sized biological network, the E.Coli transcription network, where the nodes are operons, and each edge is directed from an operon that encodes a transcription factor to an operon that it directly regulates.(12) This network was stored in a long format in a text file, which I was able to import into Gephi, Cytoscape, NetworkX, and the NetworkDynamics R package. I plotted the network with the various tools using different layouts, and I exchanged the data between them using the file format, which is a comprehensive and easy-to-use file format for graphs *graphml*. It consists of a language core to describe the structural properties of a graph and a flexible extension mechanism to add application-specific data. I evaluated each of the toolkits if they are easy to use, scriptable and supports temporal network visualization. In the end, according to these requirements, I chose the NetworkDynamics package for the rest of my projects, because it satisfies the condition above, and the R language support a lot of other visualization tools.

Visualizing various temporal networks and calculating centrality measures. During the first part of my project I downloaded the networks from online repositories and visualized them using Gephi and the NetworkDynamics R package. First I plotted the smaller networks, and without the time evolution part, then I gradually constructed bigger networks. Next I analyzed the networks by letting them evolve in time and also calculated the edge formation rate and the centrality measures. By doing this I could infer attributes of the dynamics of the underlying processes, like periodicities which could be connected to specific events. In Figure 2. and Figure 3. I show the visualization of two of the smaller networks with two different methods, one which involves multiple parameter which can be visualized as node colors or size, and the other is the so called photo reel visualization of a temporal network, which shows multiple snapshots of the same network in time. In Figure 4. and Figure 5. the results of the calculation of the metrics for the conference network can be seen, which shows the edge formation rate and the betweenness centralization during the conference event of the underlying dataset. The results clearly shows periods, where no interactions occurs between the participants, like when they attend a lecture for example.
This part of my project was about experimenting with different networks, and see if I can calculate the centrality, and other measures which could describe the Milan dataset in the same way, and they are going to be applied to the Milan dataset in much bigger scale.

Analysis of the telecommunication network of Milan

After I experienced how the NetworkDynamics package works on smaller networks, I started to analyze the telecommunication network of Milan. I was interested in observing periodic events in the nodes of the network, and also in the calculated centrality, and other measures, which describes the network as a whole. I hypothesized I could see the Christmas and New Years Eve holidays in the time series, as outlier events. Also trends, like daily, and weekly periods are expected to be present in the data. Last, but not least I compared the time evolution of the network to a randomized one, to see the differences in their structure. One would expect that in a randomized network there should not be so many complex structures, as in the original one.

My network analysis method was the following: First I downloaded the data from the (20) repository. Then I sampled n random nodes with the *grep* command to obtain a smaller network, which I can handle with R. Next I performed data cleaning, and merging with the file containing the metadata on the geo-location of the nodes. After the data was in an appropriate format I converted into a network data type with NetworkDynamics, and performed the visualizations, map embedding, and calculated the measures. All of the measures were decomposed with the *TTR* package into a trend, seasonal, and random part (in an additive way), where the trend corresponded to the weekly variations, and the seasonal to the daily ones (21). I applied simple moving averaging to the obtained time series, to smooth the data, with $n = 20$, or $n = 3000$, depending if I analyzed the 5 or the 50 nodes network.

Exploring and interpreting the Milan telecommunication network on 1 day of data with 100 nodes. The Milan telecommunication network is a huge dataset, therefore it makes it difficult to analyze. Because of this at first I only included one day of data (1st of November, 2013), of 100 nodes, and only the edges which edge weight was bigger than the median of all edge weight. By modelling the first day as a temporal network I could identify periodic events and peak hours, as one can see in Figure 6. and Figure 7. Next I quantified how randomly chosen node-pairs correlate with each other and how they are getting closer to each other during the day, and also I identified outlier phone intensity events on a histogram of all the edge weights. Also I draw a Sankey diagram and a heatmap of the time evolution of the time series (Figure 8 and 9) for the specific nodes, which can be used for forecasting into the future (for example by taking the highly correlated nodes, and by measuring one of them, predict the other). By acquiring the metadata of the dataset I could embed the network on a map. Projecting the data like this, one can correlate the features with other geographic indications, like internet traffic usage, weather data, and so on.

Exploring and interpreting the Milan telecommunication network on 2 months of data with 5 nodes. There are outlier events and periodic phenomenons which can be observed only in the 2 month period of the data. First I selected only the nodes, which interact with themselves, which represent the inter-communication in a given square of the GRID. I made an interactive plot (see the *media* folder from the supplemental data to access to interactive plots and movies), where I plotted all of the time series of the activities of these nodes, hence one can compare with each other. Daily and weekly periods can be easily determined in this plot. Next I decomposed the time series into a

trend, seasonal, and random series, where two additional information can be determined. The first is that each day there are peak hours in telecommunication from 8 a.m. to noon, and then till the evening. The second is that during the workday the activity is higher than in the weekends.
I also located the two trivial outlier event. The first is the Christmas holiday around 24th of December, the second is the New Years Eve. There is also a non-trivial event, which is in the 4th of November, the National Unity and Armed Forces Day, which is a public holiday from its inception in 1919, it marks the anniversary of the ratification of the Armistice of Villa Giusti between Italy and Austria-Hungary. These events can be observed when I plotted two different node time-series of the edge formation rate in Figure 11, which was also decomposed into a seasonal, trend, and random components in an additive way. The weighted edge formation rate of the nodes and the node pairs can be seen in Figure 10, for the interactive version of the plot please refer to the Github repository of the project. Note that each of the time series of the nodes have also a spatial dependency, so if for example there is a football game in a stadium somewhere in Milan, the increased activity is present only on that region if one only analyze the intra-communication in each square of the GRID. This spatial dependency is a very interesting topic, and could be a topic of a future project.

Centrality measures were calculated in 60 days of data with 50 nodes. I embedded the 50 randomly selected nodes to the map of Milan, which can be seen in Figure 18.
In Figure 19-21 one can see the decomposed centrality measures on the whole time scale for 50 nodes. The best measure, which can be used to identify the weekly variations was the degree and the eigenvector centralities, which gave very similar results, but the betweenness centrality captured less of this information.

Comparison of the measures if the real network with randomized networks with the 5 nodes subset of the data. I randomized the original dataset in a way to compare its attributes to random networks, where the higher level structures are absent because of the randomization process. I performed the randomization with edge randomization which takes a measurement at time t and randomly change the source and destination nodes of that link, but not allowing self-loops to be created. This way the network is randomized in each time step independently. There are more randomization techniques, which preserve the degree of the nodes, and would be interesting to extend my analysis with these as well in the future.
First I calculated all of the metrics on a randomized network in time, as I did previously with the original data. Then I repeated the whole randomization 50 times, and took the mean and median of the metrics, and visualized the calculated time series with the one standard deviation error rate for the mean, and 25% and 75% quantiles for the median. This way one can compare the results of the randomization and infer the distribution of the metrics. In Figure 12 one can see that there are times when the mean or the median of the randomized network's metrics follow the course of the original one, but there are significant differences between them.
In Figure 13 and 14 I compared two structure of the original network, when the degree centrality is high, but the betweenness centrality is low, and the opposite case as well, to obtain a bigger picture what is happening when these metrics are far from each other, and why the randomized network is different from the original one.
Regarding the distribution of the metrics of the randomized network I plotted a histogram from an arbitrary time point, which shows non-Gaussian distribution in Figure 15. I also compared the median with the mean in Figure 16, and the standard deviation - quantiles difference metrics in Figure 17. In the case of mine when the correlation is low between these values it indicates a non-Gaussian distribution of the metrics, as validated before.

Results of the exploration of 3D visualization tools

To test the capabilities of the 3D converting methods I experimented with two objects. One can be seen in Figure 23 and the second is a result of an n-body simulation which consist of 50 bodies, and their respective X, Y, Z coordinates, and a snapshot of this data can be seen in Figure 24.
In Figure 22. I attempted to visualize the projected anaglyphic visualization of the Iris dataset with the *Tourr* R package. To render objects in 3D I used the *mplot3d* python package, which allows to obtain pictures from two slightly different angles. I used this package to create objects and I rotated them with 1 degree of angle, and later combined these two movies into one movie in both anaglyph and shutter formats. The shutter movie can be viewed with the *mplayer* software.

Next I automatically converted my existing plots with online tools from 2D to anaglyph 3D, but the results were far behind my expectations. The next step was to create objects with *mplot3d* and render them from different angles, then combine and convert the video to 3D in *StereoMovie Maker* software. In Figure 23 can be seen such a converted 3D object, which can be observed with appropriate glasses.
My last work within the 3D visualization tasks was to visualize the n-body dataset in 3D. The results can be seen in Figure 24 and a video was uploaded to the *media* folder.

Discussion and future perspectives. I summarized my work plan and progress in the Supplementary Note 1 part. I managed to finish most of the goals of the project during the semester regarding the network analysis parts. However I planned to achieve more in 3D visualization, but the material turned out to be harder to learn and implement than I expected during the planning of the project.
To summarize my achievements I planned and implemented an end-to-end network analysis pipeline, which yielded non-trivial discoveries in the dataset, and can be applied

to other data types as well. From the simple input-output handling tasks to the more complex centrality measures calculation of an evolving network I performed various analysis, interpretations, and visualizations. First I showed how I can transfer data between various tools, then visualized smaller networks. From the conference network I successfully identified periodicities, which I interpreted as events, like lectures between brakes, which corresponds to low activity time-frames between people.
By analyzing the daily variations of the Milan dataset, I verified the findings of (8) about the daily activity and measurable habits of the inhabitants of the city. By the additive decomposition of the related activity of the nodes on longer time frames I identified daily and weekly seasonal and trend elements, which shows anomalous days like the Christmas and New Years Eve periods. I also identified one of the holidays of Italy, which was unexpected and one of my main findings.

I experimented with various 3D visualization techniques to find patterns in the data more easily and in a more natural way. I successfully converted a simple simulation with the *StereoMovie Maker* and *mplot3d* software and package to anaglyph and shutter 3D. However these tasks were not fully completed due to time shortages, but these results open up a lot of new ideas to experiment in the future.

Software availability. All my codes and files which I used during my project can be found at this url:
https://github.com/Paureel/time_evolving_networks. The structure of the repository follows:

- **codes:** Scripts used for the evaluation and exploration of the dataset. *milan_explore_5nodes.R* corresponds to the 5 nodes, *milan_explore_50nodes.R* to the 50 nodes analysis. Other scripts can be found here as well. Use the command from *preprocess_milandata.txt* to extract the nodes from the raw Milan dataset.

- **network_files:** Folder containing the network files generated during the earlier parts of the project.

- **media:** Interactive plots and videos from the project.

ACKNOWLEDGEMENTS
Firstly, I would like to express my sincere gratitude to my advisor Prof. Péter Pollner for the continuous support of my project, especially the help with the 3D visualization tools. Secondly, I thank my fellow lab-mate András Biricz for preparing the n-body simulation data for me.

Bibliography

1. Petter Holme. Temporal networks. In *Encyclopedia of Social Network Analysis and Mining, 2nd Edition*. 2018. doi: 10.1007/978-1-4939-7131-2_42.
2. Loet Leydesdorff. "betweenness centrality" as an indicator of the "interdisciplinarity" of scientific journals. *Journal of the American Society for Information Science and Technology*, 58, 07 2007. doi: 10.1002/asi.20614.
3. https://icon.colorado.edu/#!/, .
4. http://openconnecto.me/graph-services/download/, .
5. http://academictorrents.com/details/9b13183dc4d60676b773c9e2cd6de5e5542cee9a, .
6. http://www.netdimes.org/new/, .
7. https://www.ncbi.nlm.nih.gov/geo/, .
8. Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex 'Sandy' Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. In *Scientific data*, 2015.
9. http://www.brepols.net/Pages/ShowProduct.aspx?prod_id=IS-9782503581057-1, .
10. http://www.sociopatterns.org/datasets/hypertext-2009-dynamic-contact-network/, .
11. https://github.com/kateto/R-igraph-Network-Workshop-NetSciX, .
12. http://www.weizmann.ac.il/mcb/UriAlon/e-coli-transcription-network.
13. http://philogb.github.io/blog/2009/09/30/force-directed-layouts/, .
14. http://cs.brown.edu/people/rtamassi/gdhandbook/chapters/force-directed.pdf, .
15. https://www.sci.unich.it/~francesc/teaching/network/betweeness.html, .
16. https://www.geeksforgeeks.org/, .
17. https://docplayer.net/60759767-3-d-stereoscopic-plots-from-history-to-r.html, .
18. M. B. Islam, L. Wong, K. Low, and C. Wong. Aesthetics-driven stereoscopic 3-d image recomposition with depth adaptation. *IEEE Transactions on Multimedia*, 20(11):2964–2979, Nov 2018. ISSN 1520-9210. doi: 10.1109/TMM.2018.2820324.
19. https://phys.org/news/2009-09-shutter-3d-technology-hdtv.html, .
20. https://dandelion.eu/datagems/SpazioDati/telecom-mi-to-mi/resource/.
21. https://otexts.com/fpp2/components.html.

Supplementary Note 1: Workplan and figures

In the next pages I summarize my work plan and include the figures generated during the project.

Task name	Task description
Literary Research - Temporal networks	Read papers about time evolving networks and 3D visualization.
Installing softwares and tools	Install softwares and libraries for network analysis and simulation. Experiment with data exchange between the installed tools.
Temporal networks: visualizations	Experiment with different representations of time evolving networks.
Calculating centrality measures	Calculate the centrality measures for an evolving network from a network data repository.
Exploring 3D visualizations	Create a 3D, interactive demo of time evolving networks in the 3D Virtual Reality Lab.
Exploring the Milano telco. dataset	Explore the time dependent weighted network of mobile phone calls in northern Italy.
Interpreting the Milano telco. dataset	Find basic repetitive patterns (daily, weekly etc.), and locate outlier events.
Compare real networks with randomized ones	Create randomized versions of the above networks. Determine the significant differences from random behavior.

Table 1. The work and progress plan. I summarized all of the tasks of the project in this table.

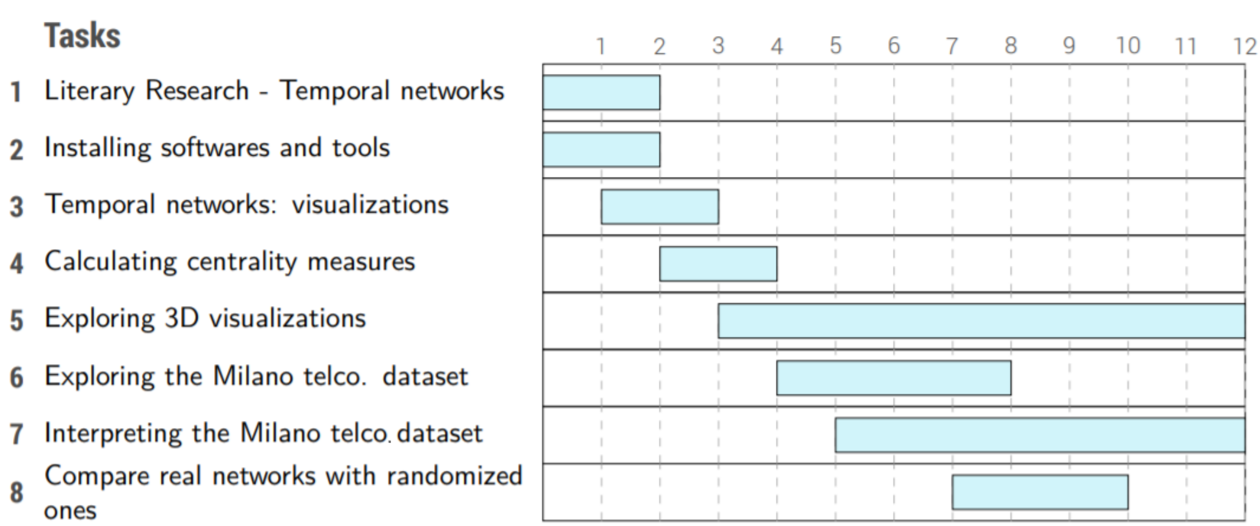

Fig. 1. Time schedule. In this figure I present my time schedule for each of the tasks. The numbers indicate the weeks of the semester.

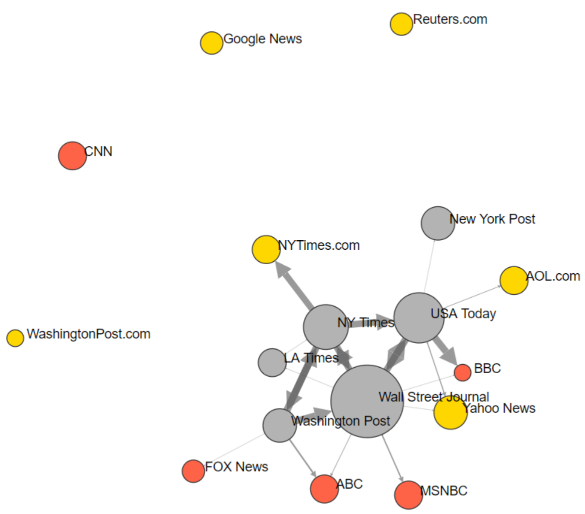

Fig. 2. Snapshot of the media network. I froze the time evolution of the directed, weighted media network, and visualized using the NetworkDynamics R package. It shows that multiple parameters can be visualized using this package, like edge weight and node size, which are proportional in this figure to the values of these parameters.

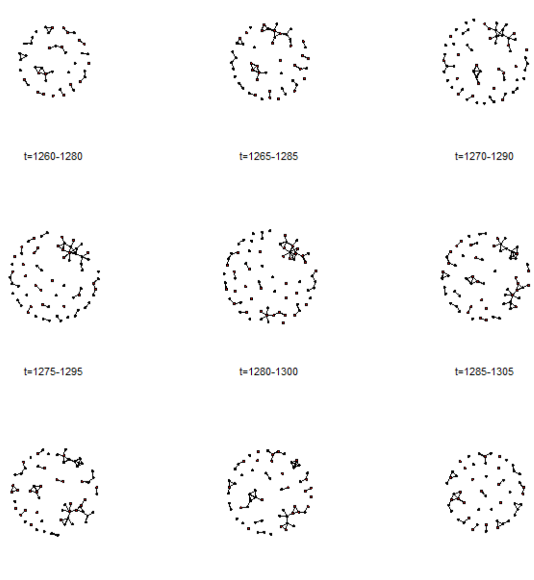

Fig. 3. Photo reel visualization of the manuscript network. The time evolution of a small network can be easily visualized as a photo reel.

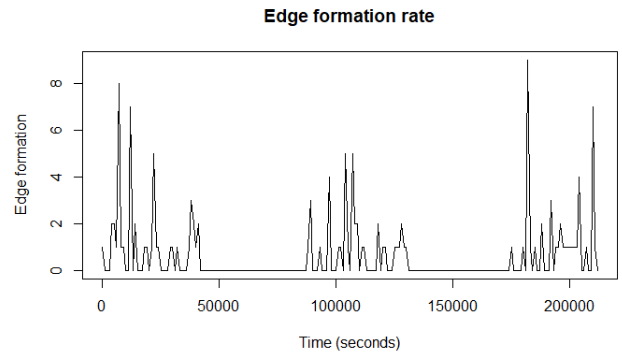

Fig. 4. Edge formation rate of the conference network. This plot shows the edge formation rate of the first day of the conference network. The zero values could be considered events like presentations when people could not interact with each other.

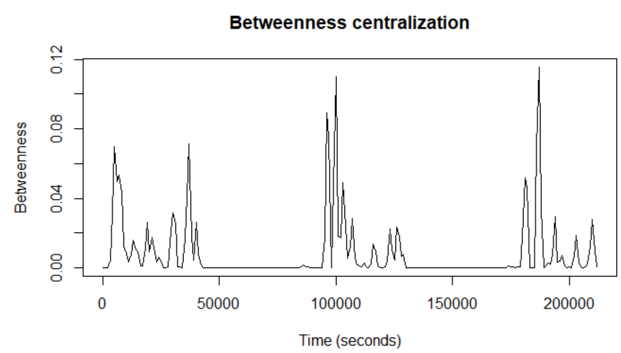

Fig. 5. Betweenness centralization of the conference network. This plot shows the betweenness centralization of first day of the conference network. The explanation is similar to the edge formation rate for the same network.

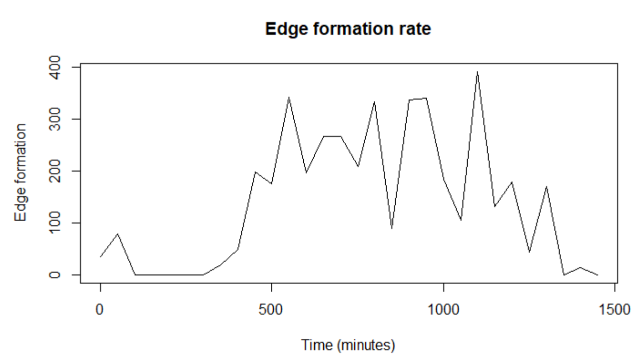

Fig. 6. Edge formation rate of the Milano network for 1 day of data. This plot shows the edge formation rate of the first day of the Milano network. One can observe that during daylight there is a heightened intensity of telecommunication processes.

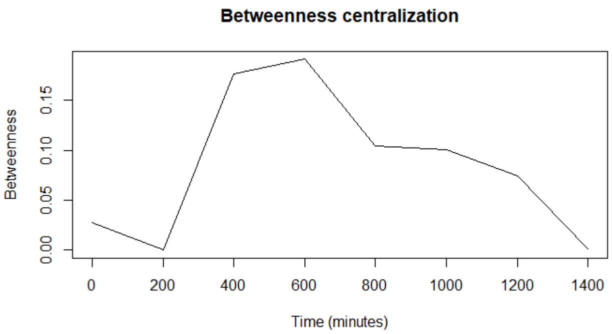

Fig. 7. Betweenness centralization of the Milano network for 1 day of data. This plot shows the betweenness centralization of first day of the Milano network. The explanation is similar to the edge formation rate for the same network.

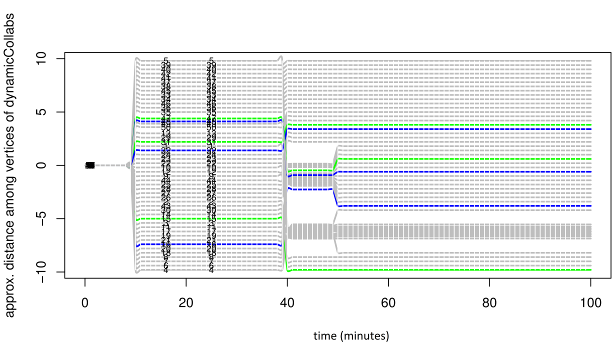

Fig. 8. Sankey diagram of the Milano network. This plot shows how far the nodes are apart from each other (how many nodes takes to travel from one to the other) in the first part of the first day of the Milano dataset. When groups of nodes form, their communication intensity correlate well with each other, hence the closeness.

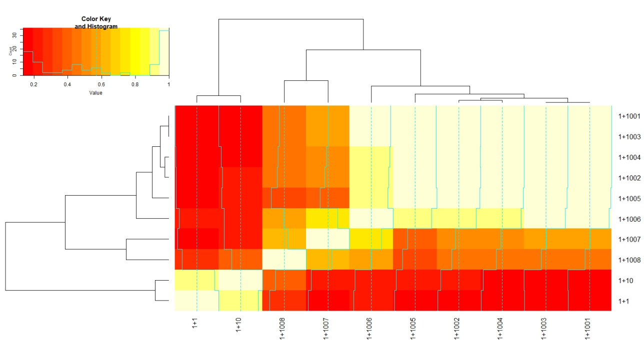

Fig. 9. Correlation heatmap of node pairs of the Milano network chosen randomly. The correlation heatmap shows how node-pairs behave similarly. A dendogramm was added to group the similar node-pairs together. The result can be used to forecast the activity levels of specific nodes by measuring the value of the nodes, which are highly correlated with the selected one.

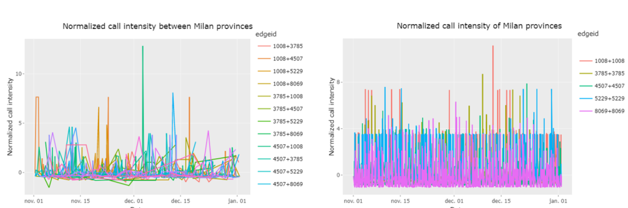

Fig. 10. Time series of the weighted edge formation rate of the nodes and the node pairs. An interactive version can be found in the repository of the project, and can be used to explore the correlations between the series.

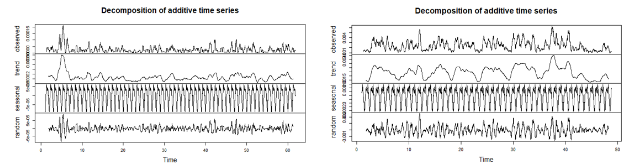

Fig. 11. Additive decomposition of the weighted edge formation rate of node 3785 (left) and node 1008 (right). On the right one can observe the national holiday at fourth of November, and the Christmas holiday at the right. The difference could be explained the different geographic locations.

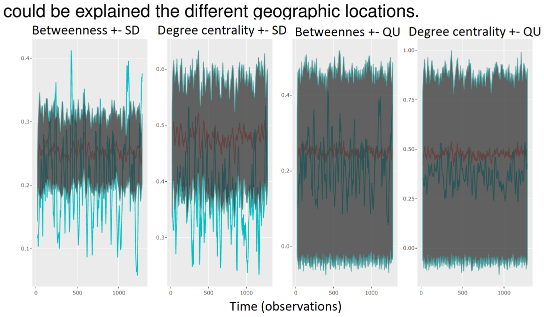

Fig. 12. Summary of the metrics of the original and the randomized networks In the first and second plots one can see the original betweenness and degree centralities compared to the mean of the randomized ones, with the standard deviations indicated. On the third and fourth plots are the same metrics, but compared with the median values and 25% and 75% quantiles. One can observe parts of the time series where the peaks are highly correlated with the original data, but also when there are not, as expected. Taking the quantiles results in a wider range of the values, due to the data not being normally distributed.

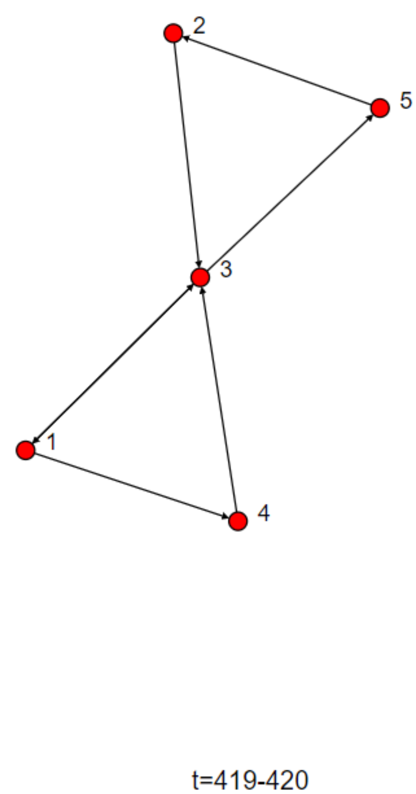

Fig. 13. The network structure when the betweenness is high, but the degree centrality is low compared to each other. The reason behind this that nearly all of the nodes are easily reached.

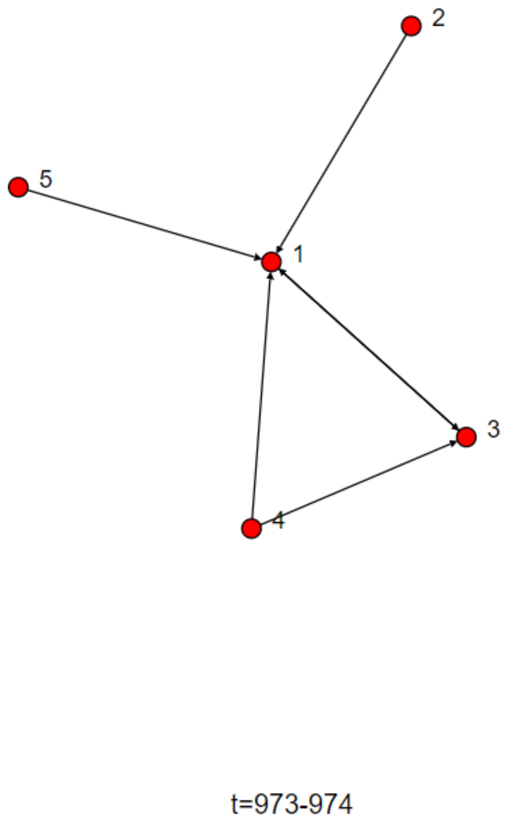

Fig. 14. The network structure when the degree centrality is high, but the betweenness is low compared to each other. The reason behind this that there are nodes which cannot be reached, and there is one central node with high node degree.

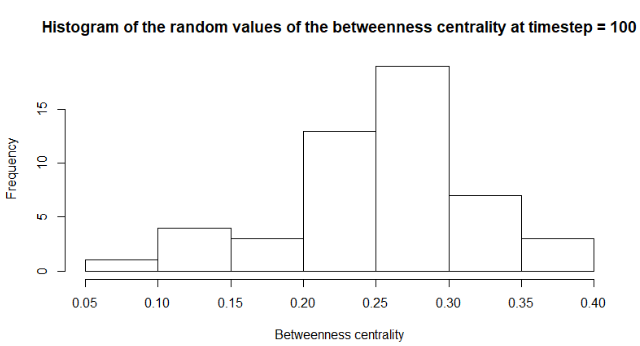

Fig. 15. Histogram of the betweenness centralities of the random networks at an arbitrary time One can see in this histogram that the calculated random centralities are not normally distributed.

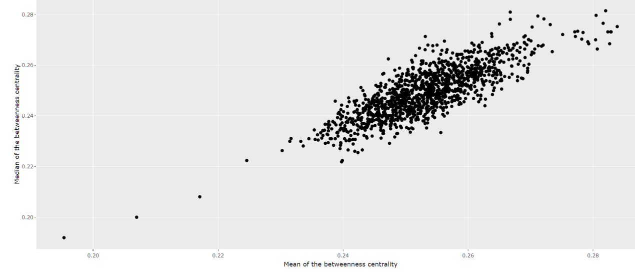

Fig. 16. The mean of the betweenness centrality versus the median of it The correlation is rather high of these two measures when compared.

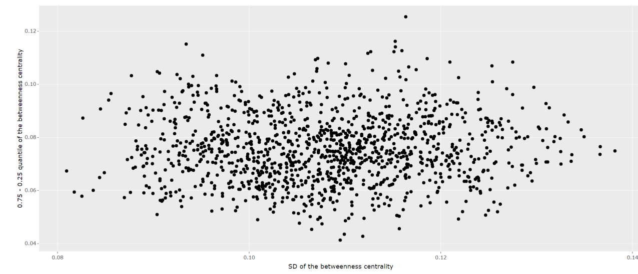

Fig. 17. The standard deviation of the betweenness centrality versus the difference between the 75% and the 25% of it The correlation is nearly zero, which can be explained by the highly varying behaviour of the underlying distributions.

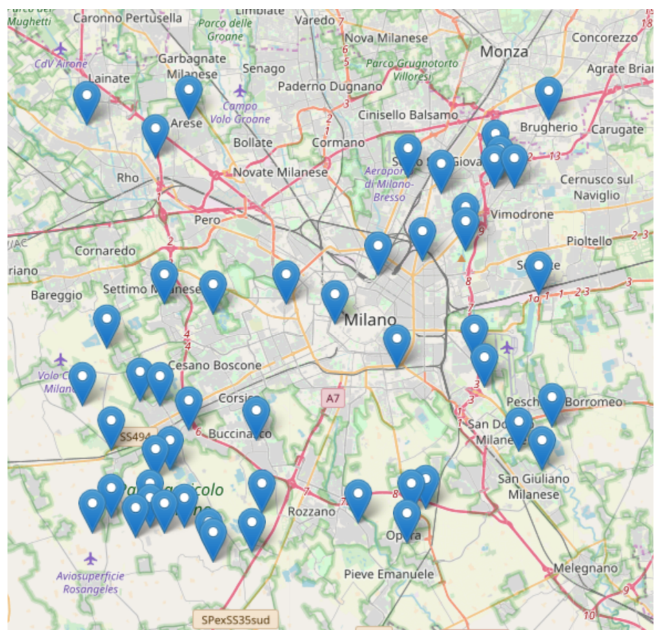

Fig. 18. Geographic embedding of the Milano network for the 50 nodes set I took the metadata regarding the nodes to embed the nodes on a leaflet map.

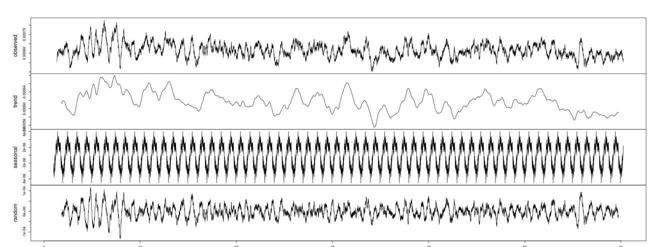

Fig. 19. Additive decomposition of the betweenness centrality on the 2 month data of the 50 nodes. It is harder to distinguish weekly trends compared to the edge formation rate, but daily periods can be observed easily.

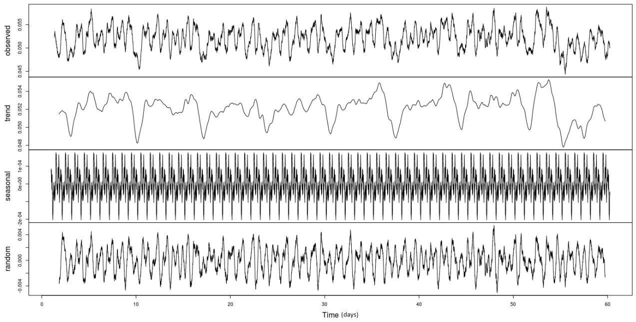

Fig. 20. Additive decomposition of the degree centrality on the 2 month data of the 50 nodes. The weekly trends can be observed easily.

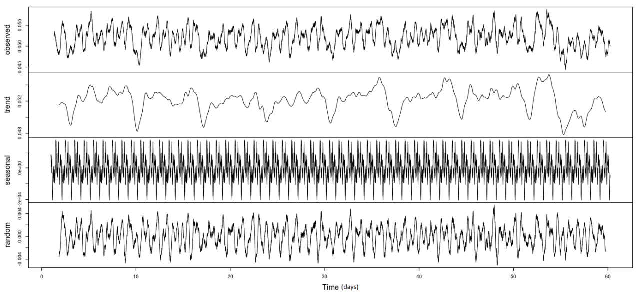

Fig. 21. Additive decomposition of the eigenvector centrality on the 2 month data of the 50 nodes. Nearly identical results were obtained compared to the degree centrality.

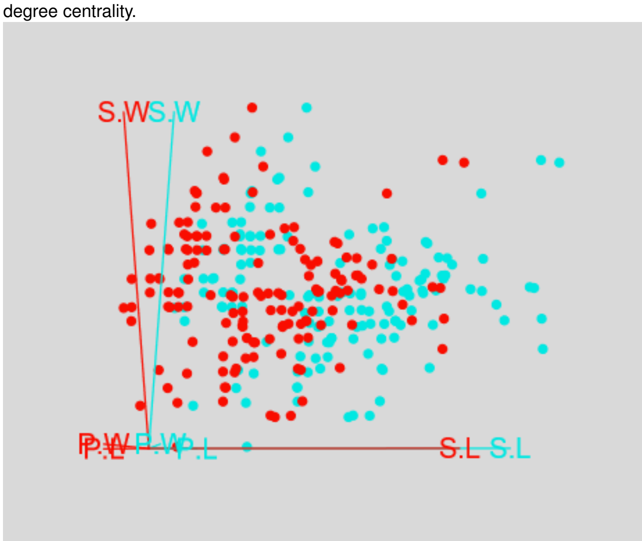

Fig. 22. Anaglyph visualization of the Iris dataset This is the first attempt towards true 3D visualization of a simple dataset. I processed the Iris dataset with PCA technique, and visualized the first 3 PC. Next I converted the results with the *Tourr* package into anaglyph form.

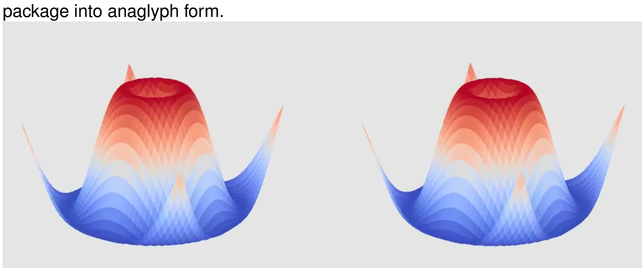

Fig. 23. A test surface rendered by two slightly different angles. The surface corresponds to the $f(X,Y) = sin(\sqrt{X^2+Y^2})$, and with the appropriate tool can be converted into anaglyph and shutter 3D formats.

Fig. 24. Sample image from the n-body simulation 3D rendering. The simulation was rendered from a slightly different angles, and was converted to shutter 3D later.