D6.1 PRELIMINARY REPORT ON NETWORK ANALYSIS

Grant Agreement:	833635
Project Acronym:	ROXANNE
Project Title:	Real time network, text, and speaker analytics for combating organised crime
Call ID: **Call name:**	H2020-SU-SEC-2018-2019-2020, Technologies to enhance the fight against crime and terrorism
Revision:	V1.0
Date:	01 June 2020
Due date:	01 June 2020
Deliverable lead:	LUH
Work package:	WP6
Type of action:	RIA

Disclaimer

The information, documentation and figures available in this deliverable are written by the "ROXANNE - " Real time network, text, and speaker analytics for combating organised crime" project's consortium under EC grant agreement 8833635 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

Project co-funded by the European Commission within the H2020 Programme (2014-2020)		
Nature of deliverable:	R	
Dissemination Level		
PU	Public	☒
CO	Confidential, only for members of the consortium (including the Commission Services)	☐
EU-RES	Classified Information: RESTREINT UE (Commission Decision 2015/444/EC)	☐
* R: Document, report (excluding the periodic and final reports) DEM: Demonstrator, pilot, prototype, plan designs DEC: Websites, patents filing, press & media actions, videos, etc. OTHER: Software, technical diagram, etc.		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Revision history

Revision	Edition date	Author	Modified Sections / Pages	Comments
V0.1	16 March 2020	Tuan-Anh Hoang [L3S]	All	Original draft
V0.2		Francesco Calderoni [UCSC]	relevant passages on criminal networks	
V0.3	15 Apr 2020	Nikos Nikolaou [ITML]	inputs on several sections (3.2, 3.4.1, 4)	
V0.4	11 May 2020	Tuan-Anh Hoang [L3S]	All	Restructure
V0.5	15 May 2020	Francesco Calderoni [UCSC]	On-going work and future direction	
V0.6	18 May 2020	Nikos Nikolaou [ITML]	Preparation for integration On-going work and future direction	
V0.7	19 May 2020	Tuan-Anh Hoang [L3S]	Preparation for integration On-going work and future direction	
V0.8	21 May 2020	Stefano Guastamacchia [UCSC]	All	Few suggestions on mentioned bibliography (especially sections 2.1.1 and 2.1.2), minor typos
V0.9	26 May 2020	Erinc Dikici [SAIL]	All	Review, correction of typos
V1.0	01 June 2020	Tuan-Anh Hoang [L3S]	All	Revise according to reviewer's comments
V1.0	01.June 2020	Sergej zerr [L3S]	All	Final checks, proofread and formatting
V1.0	01 June 2020	Petr Motlicek [IDIAP]	All	Proof-reading
V1.0	01 June 2020	Johan Rohdin [BUT]	All	Proof-reading

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Executive summary

This deliverable D6.1 is the preliminary report on network analysis is in reference to tasks T6.3: Multilayer and cross network structural analysis; T6.4: Multilayer and cross network behavioral analysis; T6.5: Latent subnetwork detection; and T6.6: Subnetwork shrinking. Its purpose is to provide information on methods and systems for network analysis and their application in analyzing criminal networks. It also describes the systems and methods developed and researched on, and the analysis of datasets collected and produced within ROXANNE. It contains a reviewed and modified version of some ideas presented in the Grant Agreement as well as in the deliverable D4.1: Overview and analysis of lawfully intercepted and publicly available data, and D5.1: Initial speech/text/video technologies.
This deliverable is a living document; it will be updated as the project progresses, and as new datasets, data types or technologies are made available to the consortium. Updates and extensions to this document will also be covered in D6.3: Analysis of criminal networks (due M24), and D6.4: Systematic comparison of criminal networks (due M33).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

1. Introduction

1.1. Background

The objectives of WP6 are given in the ROXANNE Grant Agreement as follows:

- Build a framework to extract static and dynamic representations of criminal networks from multimodal sources such as video streams or geo-locations, to support the process of investigation through modeling criminological concepts, correlation analysis across different events, and providing statistics in supporting SID, SLT, VA related activities from WP5 and vice versa;
- Develop a statistical interface between SLT/metadata/VA and network analysis;
- Investigate, develop and evaluate methods for the network(s) based identification, detection, and mining of related individuals along with their activity patterns;

This WP6 will, therefore, take as input the interactions and relations extracted by technologies developed in WP5, construct networks from the input, perform analysis to help end-users to get insights from the data that are suggestive for their inspection work, as well as to provide feedback for improving the performance of the technologies in WP5.

1.2. Purpose of the document

The primary purpose of this document is described in the ROXANNE Grant Agreement [1], as to provide an "initial report and systems on network analysis". Along these lines, this document is produced with the efforts in the tasks "T6.3: Multilayer and cross-network structural analysis", "T6.4: Multilayer and cross-network behavioral analysis", "T6.5: Latent subnetwork detection" and "T6.6: Subnetwork shrinking", whose details are given in Section 2.

The members of the consortium participating in WP6 provide the information regarding datasets that they are aware of or have access to and that they could use in ROXANNE. They also provide information regarding methods and systems that they have been using for network analysis, particularly for the analysis of criminal networks.

Besides T6.3- T6.6, the other tasks in ROXANNE which this deliverable has relations to are:

- T6.1: Fusion of information from component technologies for network analysis
- T6.7: Systematic assessment of the contribution of SLT and relation analysis to criminal NA, and
- T2.1: Collection of end-user requirements

1.3. Document structure

This document is organized as follows: Section 2 briefly describes families of methods that have been used for network analysis. We present the intuition, principle, technical considerations, and potential application in ROXANNE for each method. In this section, we present an overview of existing systems for network analysis in general and criminal network analysis in particular. Section 3 describes the systems that have been developing within ROXANNE. In Section 4, we give some initial results of applying these systems on ROXANNE's datasets. Section 5 describes our activities for integrating these systems into ROXANNE's platform. Finally, we describe the on-going work and draw some future directions for WP6 in Section 6.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

2. Basics of network analysis technologies

Briefly described, network analysis is the process of uncovering hidden patterns regarding the behavior and relations among individuals in networks through the use of a wide range of computational and statistical methods. Examples of those patterns are the distribution of relations among the individuals, the underlying factors that determine the relations, and cohesive groups of individuals with dense relations, etc. These methods have been applied in various domains of daily life, including economics, biology, and sociology, and particularly security and criminology. In this section, we present the methods most closely related to ROXANNE and an overview of existing systems and software that provide those methods.

2.1 Methods for network analysis

We summarize in the following subsections the existing methods for network analysis functionalities that are particularly relevant to ROXANNE's mission. These functionalities include social influence analysis, community detection, link prediction, outlier detection, and network embedding. For each of the functionalities, we shall present the objectives, the potential applications in ROXANNE, and the typical existing methods.

2.1.1 Social influence analysis Objective

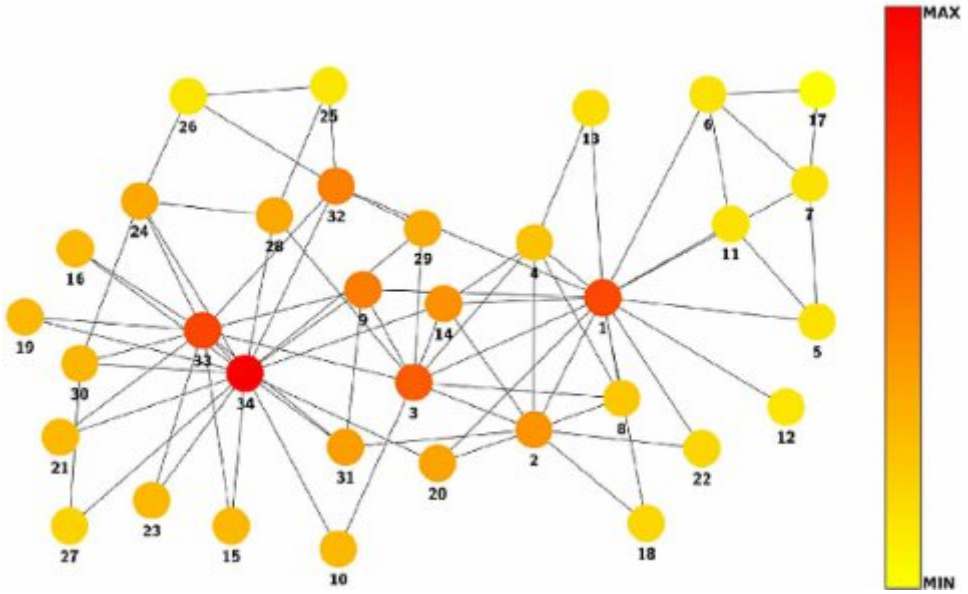


Figure 1. Example of social influence score measured for individuals in Zachary's karate club network[1],[2]

Objective: This task aims to quantify the influence that individuals have over the others within a social network. In some contexts, this task is also called influential individual identification. Here, within the ROXANNE project, where the domain is already determined and well understood, we focus on the overall, global influence of each individual within the network. Note that, in open domain settings, the influence can also be explicitly quantified to the domain or pairwise

[1] Zachary, Wayne W., An Information Flow Model for Conflict and Fission in Small Groups, 1977, Journal of Anthropological Research, p. 452-473, Vol 33, n 4

[2] Image taken from Alvarez-Socorro, A. J., G. C. Herrera-Almarza, and L. A. González-Díaz. "Eigencentrality based on dissimilarity measures reveals central nodes in complex networks." Scientific reports 5 (2015): 17095.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

relations. In this task, we would like to assign each individual a relative importance score, as demonstrated in Figure 1, that measures its influence compared to that of other individuals.

Application in ROXANNE: Inquiry on the most influential individuals in a criminal group has been one of the leading drivers of network analysis applications to organized crime. Most studies apply the centrality measures (a family of measures aiming at assessing how central, i.e. essential or influential, a node is in a network) to criminal organizations.

Using centrality measures in criminal networks poses several challenges:

- The analysis is just as good as the input data is[3]. Most existing studies rely on co-offending, criminal intelligence, telephone calls, or meeting data. This data is inevitably influenced by the investigations' legal and operational context, which changes across countries and agencies. As a consequence, academic studies and law enforcement "raw" data rely on a non-representative subsample of the relations among the suspects.
- The types of relations analyzed determine the network topology. some studies showed that analyzing the same gang through co-offending, criminal intelligence, and telephone contact data yielded substantially different networks[4].
- There is strong evidence that criminals may take active measures to avoid detection, such as avoiding telephone use, prefer secure communication channels[5],[6] (either through technology or completely avoiding technology). Furthermore, criminals may also avoid explicit references and languages: they may use jargon, codes, pseudonyms and similar tactics.
- Research has sometimes shown that there is a difference between actual status within a criminal group (e.g. position in the formal hierarchy) and network centrality. The difference may be the result of organizational arrangements aimed at protecting the most important criminals and leaving in the most visible and vulnerable positions, middle-level managers[7],[8] .

The above challenges must be considered carefully in the implementation of ROXANNE. The type of input data does inevitably determine the possibility and extent of the analysis of influential nodes. Nevertheless, the analysis of criminal networks relies on a set of now popular centrality measures, which will also be implemented in ROXANNE.

Methods: Typically, individuals' influence in a general network can be measured by the following metrics, each has its own intuition and thus quantifies a particular aspect of the individuals' importance within the network.

- **Degree**[9],[10]: this is the number of relations or links to other individuals that the targeting individual involves in. Intuitively, this metric can be interpreted as the immediate likelihood of the targeting individual in monitoring or sensing whatever is flowing through the network (such as an exchange of some information)
- **Closeness**[11],[12]: intuitively, this metric measures how close the targeting individual is to all other individuals. Mathematically, it is computed based on aggregating the length of the shortest paths between the targeting

[3] Morselli, Carlo. Inside criminal networks. Vol. 8. New York: Springer, 2009.

[4] Rostami, Amir, and Hernan Mondani. "The complexity of crime network data: A case study of its consequences for crime control and the study of networks." PloS one 10.3 (2015).

[5] Calderoni, Francesco, and Elisa Superchi. "The nature of organized crime leadership: criminal leaders in meeting and wiretap networks." Crime, Law and Social Change 72.4 (2019): 419-444.

[6] Agreste, Santa, et al. "Network structure and resilience of Mafia syndicates." Information Sciences 351 (2016): 30-47.

[7] Morselli, Carlo. "Assessing vulnerable and strategic positions in a criminal network." Journal of Contemporary Criminal Justice 26.4 (2010): 382-392.

[8] Calderoni, Francesco. "The structure of drug trafficking mafias: the 'Ndrangheta and cocaine." Crime, law and social change 58.3 (2012): 321-349.

[9] Easley, David, and Jon Kleinberg. Networks, crowds, and markets. Vol. 8. Cambridge: Cambridge university press, 2010.

[10] Freeman, L. C. (1979). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

[11] Easley, David, and Jon Kleinberg. Networks, crowds, and markets. Vol. 8. Cambridge: Cambridge university press, 2010.

[12] Freeman, L. C. (1979). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

individual and all other individuals in the network. In the purest form, it is the average of paths' length, and thus only valid in a connected network where there is always at least one path between any pair of individuals. In more general settings, it is computed by the harmonic mean of the paths' length with the convention that if there is no path between a pair of individuals, then that pair has zero contribution to the mean.

- **Betweenness**[13],[14]: this metric quantifies the number of times the targeting individual lies on the shortest path between pairs of other individuals. Intuitively, this measures the likelihood that the targeting individual plays as a bridge for and thus controls the communication between other individuals.

- **Pagerank score**: inspired by the ranking of webpages in search engines[15], this metric measures the importance of the targeting individual by counting the weighted vote it receives from other individuals. Here each in-link is considered as a "vote", and the weight is determined by the intuition that important individuals are more frequently voted by other important individuals. Mathematically, this leads to a recursive equation whose solution can be effectively found by using power iteration methods[16].

- **Authority and Hub scores**: similar to Pagerank score, hub and authority scores also originated in search engine research[17]. These are two scores that quantify the hub and authoritativeness of individuals in a network. Intuitively, a hub is one that points to many other authoritative individuals, while the authoritative one should be pointed to by many other hubs. In other words, hubs know where the flows on the networks start from, while authoritative individuals are where the flows initiate. Again, this leads to a recursive equation whose solution can be effectively found by using power iteration methods.

The above list of methods is not exhausted in the sense that there are several variants of these methods, e.g. the Katz centrality[18] is an extension of Degree measure. However, considering that their intuition are already covered by the listed methods, we opt to leave them out.

Evaluation: As totally unsupervised and there is often no ground-truth available in reality, social influence analysis methods can mostly be evaluated qualitatively and indirectly. The qualitative evaluation includes more in-depth analysis of top scored individuals and manually examining the correlation and influence (if any) between their behaviors and the behaviors of other individuals interacting with them[19]. The indirect evaluation includes using the individuals' influence score as a feature for predicting influence-related behaviors in the future. For example, an influential individual once adopts some item, would affect others' decision in adopting similar items[20].

[13] Easley, David, and Jon Kleinberg. Networks, crowds, and markets. Vol. 8. Cambridge: Cambridge university press, 2010.

[14] Freeman, L. C. (1979). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

[15] Sergey, Brin. "The anatomy of a large-scale hypertextual Web search engine." Computer networks and ISDN systems 30.1 (1998): 107-117

[16] Brinkmeier, Michael. "PageRank revisited." ACM Transactions on Internet Technology (TOIT) 6.3 (2006): 282-301.

[17] Kleinberg, Jon M. "Authoritative sources in a hyperlinked environment." Journal of the ACM (JACM) 46.5 (1999): 604-632.

[18] Katz, Leo. "A new status index derived from sociometric analysis." Psychometrika 18.1 (1953): 39-43.

[19] Tang, Jie, et al. "Social influence analysis in large-scale networks." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009.

[20] Goyal, Amit, Francesco Bonchi, and Laks VS Lakshmanan. "Learning influence probabilities in social networks." Proceedings of the third ACM international conference on Web search and data mining. 2010.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

2.1.2 Community detection

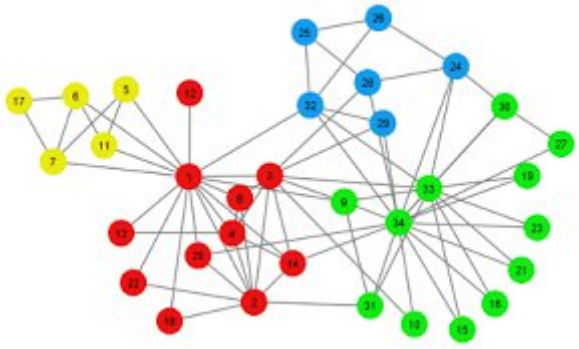


Figure 2. Example of communities of individuals in Zachary's karate club network[21]: individuals having the same color belong to the same community[22]

Objective: Very often, individuals within a network tend to form communities. As illustrated by the example in Figure 2, each community is a cohesive group of individuals whose intra-community interaction is more dense and frequent than their interaction with the rest of the network. However, in most cases, this community structure is hidden as communities are not well defined, or individuals do not publicly reveal their community membership. In this task, we aim to uncover those hidden structures using interactions among network individuals and their associated information, e.g., attributes and meta-data.

Application in ROXANNE: While community analysis has rapidly grown to be a popular field in network research in general, applications to criminal networks are still very rare. Research has shown that community analysis can hardly distinguish real-life subgroups in a criminal network (e.g. gangs, cliques, families, clans). For example, a study inquired whether the network communities in a communication network among mafia offenders corresponded to the clan subdivision within the offenders. Results showed that communities only partially overlapped with criminal clans. While some communities clearly identify mafia clans, other communities comprised multiple clans for remarkably different reasons: some clans were the offspring of others and community analysis reflected this separation process; other clans were clashing and conflicting and community analysis detected the intensified interaction between opposing groups[23].

Notwithstanding these issues, investigators may benefit from community detection methods when analysing the patterns of interaction among entities in a criminal network. While the human eye and mind can hardly identify cohesive subgroups, the clustering of social relations is an essential element of society's (and criminal groups') organization.

ROXANNE will deploy several established methods of community detection to identify cohesive subgroups of individuals/entities who interact more often among them than with other individuals in the network. Investigators will be presented with the results and will be able to focus the attention on the interactions within and between the communities. This could generate additional insight into the social organization of criminal groups and provide added value to the law enforcement agencies.

Methods: Generally, existing community detection methods[24] can be grouped into: (i) individual-centric methods, (ii) group-centric methods, (iii) network-centric methods, and (iv) hierarchy-centric methods. Individual-centric methods focus on identifying communities whose members satisfy some certain properties, e.g., the degree or reachability from other members. Group-centric methods emphasize each community as a whole: each community has to satisfy certain properties without zooming into the individual level, e.g., the density of relations or interaction among community

[21] Zachary, Wayne W., An Information Flow Model for Conflict and Fission in Small Groups, 1977, Journal of Anthropological Research, p. 452-473, Vol 33, n 4

[22] Image taken from http://www.ludowaltman.nl/slm/

[23] Calderoni, Francesco, Domenico Brunetto, and Carlo Piccardi. "Communities in criminal networks: A case study." Social Networks 48 (2017): 116-125.

[24] Tang, Lei, and Huan Liu. "Community detection and mining in social media." Synthesis lectures on data mining and knowledge discovery 2.1 (2010): 1-137.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

members. Network-centric methods focus on partitioning the network into several sub-networks; each is then considered as a community. Lastly, the hierarchy-centric methods aim to construct a hierarchical structure of communities within the network.

Parameter-wise, community detection methods can also be grouped into: (a) parametric methods - which require some input parameters, e.g., the number of desired communities, or the thresholds for intra-community degree and interaction/relation density; and (b) non-parametric methods - which do not require any input parameter and identify communities in an unsupervised manner.

In the following, we summarize typical methods of each aforementioned groups which are closely relevant and most applicable in ROXANNE

- **K-clique based method**[25]: this is an individual-centric and parametric method. It requires an input number K, which is often set in the range 3 to 5, and returns overlapping communities, i.e., each individual can be a member in more than one community. Here, each community is identified by unifying adjacent K-cliques in the network. A K-clique[26] is a fully connected sub-network, and two K-cliques are adjacent if they have K-1 individuals in common.

- **Modularity maximization**[27],[28]: this is a group-centric and non-parametric method. It works by partitioning the input network into sub-network to maximize the difference between the observed intra- sub-network interactions/relations and their expectation. It follows the intuition that communities should have much more internal interactions/relations than cross-communities ones. This leads to a NP-Hard problem that can be practically solved by greedy algorithms[29].

- **Spectral clustering**[30]**:** this is a group-centric and parametric method. It requires the desired number of communities K. It works by first performing spectral analysis[31] on the Laplacian matrix [32]of the network so as to construct a feature vector for each individual, then performing clustering on these feature vectors to obtain clusters and communities of individuals. The latter step can be done using any clustering method, e.g., K-means[33].

- **Matrix Factorization**[34]**:** this is again a group-centric and parametric method. It requires the desired number of communities K and returns overlapping communities. It works by first performing rank-K approximation[35] of the adjacency matrix[36] of the network using non-negative factorization[37] or singular value decomposition[38], then

[25] Gergely Palla, Imre Derényi, Illés Farkas1, and Tamás Vicsek, Uncovering the overlapping community structure of complex networks in nature and society Nature 435, 814-818, 2005,

[26] Luce, R.D. (1950). "Connectivity and generalized cliques in sociometric group structure," Psychometrika 15: 169-90.

[27] Newman, Mark EJ. "Modularity and community structure in networks." Proceedings of the national academy of sciences 103.23 (2006): 8577-8582.

[28] M. E. J Newman 'Networks: An Introduction', page 224 Oxford University Press 2011.

[29] Clauset, Aaron, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks." Physical review E 70.6 (2004): 066111.

[30] Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems. 2002.

[31] Seary, Andrew J., and William D. Richards. "Spectral methods for analyzing and visualizing networks: an introduction." Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers. 2003.

[32] Estrada,Ernesto, and Kight, Pilip A., A first Course in Network Theory, Oxford University Press, 2016

[33] https://en.wikipedia.org/wiki/K-means_clustering

[34] Yang, Jaewon, and Jure Leskovec. "Overlapping community detection at scale: a nonnegative matrix factorization approach." Proceedings of the sixth ACM international conference on Web search and data mining. 2013.

[35] https://en.wikipedia.org/wiki/Low-rank_approximation

[36] Harary Frank, Graphs and Matrices, SIAM Review, Vol 9, n 1, p.83-90, 1967

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

identifying the communities based on the obtained rank-K matrices: an individual *i* belongs to community *k* if the value of the *k*-th element of the i-th row (or column, depends on the method) exceeds a certain threshold.

- **Mixed-membership models**[39]: this is another group-centric and parametric method. It requires the desired number of communities K and returns soft communities: each individual belongs to each community with a certain probability. It works by assuming a stochastic model for generating observed interactions/relations among the individuals from their community membership and then learning the community membership by fitting the model to the observed data.

- **Hierarchical clustering methods:** these are hierarchy-centric methods which are either parametric or non-parametric. They can work by top-down or bottom-up approaches. Top-down methods work by iteratively dividing the (sub-)networks into smaller sub-networks to maximize a certain measure, e.g., the modularity of the obtained sub-networks[40]. Bottom-up methods, in contrast, work by iteratively merging smaller sub-networks into bigger ones so as to maximize a certain network-level measure, e.g., the overall modularity of all obtained sub-networks[41].

Certainly, the above list of methods is not exhaustive. There are other methods that leverage side information in the input data to improve the detection accuracy. For example, several methods have been proposed for incorporating individuals' attributes and interactions/relations to detect communities[42]. However, considering their computational complexity and the availability of datasets at the current phase of the project, we opt to leave them out of the list for now. We will investigate these methods in the later phase of the project when more data is available.

Evaluation: Similar to other clustering tasks, there is often no ground-truth information for community detection and the methods can mostly be indirectly. The indirect evaluation includes using some metrics for measuring the cohesiveness of the obtained communities, e.g., intra- and inter-community density, clustering coefficient, and conductance[43]. In very few cases that the ground-truth information is available, then the methods are evaluated by their ability in recovering the ground-truth communities, and often measured by metrics like precision, recall, and F-scores[44].

[37] https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

[38] https://en.wikipedia.org/wiki/Singular_value_decomposition

[39] Airoldi, Edoardo M., et al. "Mixed membership stochastic blockmodels." Journal of machine learning research 9.Sep (2008): 1981-2014.

[40] Newman, Mark EJ. "Finding community structure in networks using the eigenvectors of matrices." Physical review E 74.3 (2006): 036104.

[41] Newman, Mark EJ. "Fast algorithm for detecting community structure in networks." Physical review E 69.6 (2004): 066133.

[42] Bothorel, Cécile, et al. "Clustering attributed graphs: models, measures and methods." Network Science 3.3 (2015): 408-444.

[43] Harenberg, Steve, et al. "Community detection in large-scale networks: a survey and empirical evaluation." *Wiley Interdisciplinary Reviews: Computational Statistics* 6.6 (2014): 426-439.

[44] Yang, Jaewon, and Jure Leskovec. "Defining and evaluating network communities based on ground-truth." Knowledge and Information Systems 42.1 (2015): 181-213.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

2.1.3 Link prediction

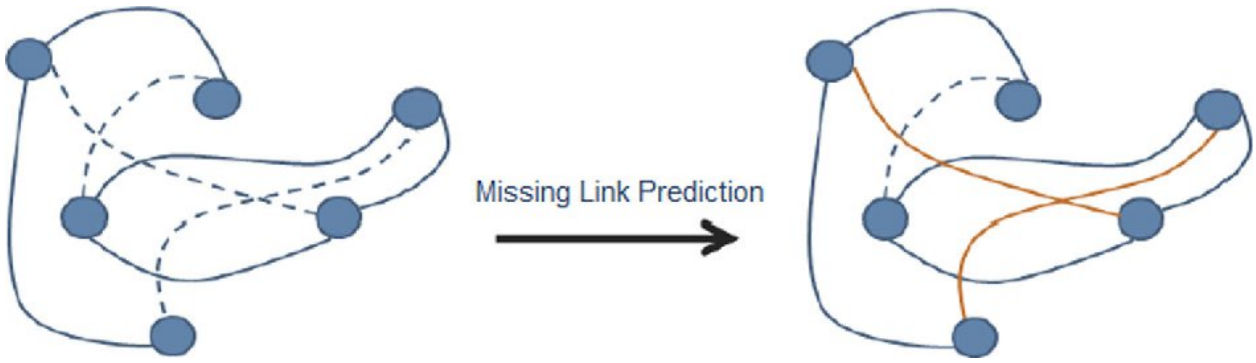


Figure 3. Example of missing links (dashed lines on the left network) and the predictions to recover them (red lines on the right network)[45]

Objective: In reality, people interact and communicate with each other through many channels that are not always observable. Hence, many interactions and relations among individuals in a network are hidden or not observed. Moreover, the network evolves as new interactions and relations are constantly added to the network. This is due to the revelation of hidden interactions and relations in the past, the availability of new data sources, and mostly, as individuals forming new relations or interacting with new other individuals. As illustrated by the example in Figure 3, in this task, we aim to uncover these missing or hidden, unobserved interactions and relations in the network, as well as to predict the most probable ones to be formed in the near future. Specifically, given an individual *u* and a number k, we would like to identify top k other individuals that are not connected with *u* but might or would have interactions/relations with *u*.

Applications in ROXANNE: Criminal investigations inevitably collect incomplete information about a group. Compared to the universe of suspects' interactions with other people, the law enforcement agencies do typically gather only a portion of these. Also, criminals have normal lives with interactions with family, friends, partners, colleagues. And these interactions take several forms, which can hardly be exhaustively mapped by law enforcement.

Furthermore, the investigative process is also designed to filter information and focus only on useful evidence. For example, only a part of telephone traffic or conversations from a criminal's telephone will offer useful evidence. Prosecutors may select only a part of this evidence for the pre-trial detention or the trial.

The process of "skimming" of information during a criminal investigation may pose the risk that relevant links may be lost or be missing. Studies in the field showed that the criminal investigation process is quite consistent in maintaining information about the main suspects in a case, whereas information on marginal suspects may be filtered out and lost[46].

Link prediction methods, increasingly popular in general network research, can assist in identifying possible missing links. ROXANNE will deploy a set of algorithms to measure the probability that two nodes are connected. This may assist investigators by providing possible leads on connections not (yet) observed in the investigation, and that may request specific actions (e.g. increased surveillance on two individuals due to the high probability of a relation between them).

Methods: In principle, all link prediction methods work by assigning a score(u, v) to pairs of nodes (u, v), then choosing the top scored pairs as predicted links for u[47]. Here the score is computed based on the input network, and often measures proximity or similarity between u and v in some sense. Generally, existing methods differ from each other by the

[45] Image taken from Haghani, Sogol, and Mohammad Reza Keyvanpour. "A systemic analysis of link prediction in social network." Artificial Intelligence Review 52.3 (2019): 1961-1995.

[46] Berlusconi, Giulia, et al. "Link prediction in criminal networks: A tool for criminal intelligence analysis." PloS one 11.4 (2016).

[47] Liben-Nowell, David, and Jon Kleinberg. "The Link-Prediction Problem for Social Networks." Conference on Information and Knowledge Management (CIKM'03). 2003.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

similarity used. In the following we summarize the typical similarities found in literature that are most relevant and applicable in ROXANNE.

- **Jaccard similarity**: this measures the probability that both *u* and *v* have past connections with individual f, for a randomly selected individual f that set of individuals that either *u* or *v* have have past connections with. Mathematically, it is determined by the ratio between the number of individuals that *both u* and *v* have observed interactions/relations with and the number of those that *either u* and *v* have observed interactions/relations with[48].

- **Adamic Adar similarity[49]**: this is a variant of Jaccard similarity where the similarity is now determined as a weighted count of the individuals that *both u* and *v* have observed interactions/relations with. Here each individual is weighted by the inversion of the number of other individuals that it has observed interactions/relations with. That is, a popular individual that has connections with many other individuals contributes less to the similarity, and an individual *f* that has past connections with only *u* and *v* contribute the most. It is reasonable as, in this case, *f* would most likely introduce *u* and *v* to each other.

- **Preferential attachment similarity[50],[51]**: this metric follows the "rich-get-richer" phenomenon[52] in network evolution. That is, the likelihood that *u* will connect with *v* in the future is proportional to the popularity of the two individuals. Mathematically, in its simplest form, this similarity is determined by the product between the number of other individuals that *u* has observed interactions/relations with and that number of *v*

- **Resource allocation similarity[53]**: this metric is proposed to measure the resources that flow from *u* to *v* through other individuals that *u* has connections with. Mathematically, it has very similar form with the Adamic Adar similarity except that each individual is now weighted by the number of other individuals that it has observed interactions/relations with.

- **Soundarajan-Hopcroft similarity[54]**: this is a variant of Adamic Adar similarity which is enhanced by using community structure in the input network. It follows the intuition that the more community membership the two individuals have in common the more likely they form some connection. Mathematically, the score is determined by the number of individuals that *both u* and *v* have observed interactions/relations with plus the number of communities they both belong to.

The methods listed above are purely based on interactions/relations among the individuals in the network. Obviously, they can be improved by also taking into consideration the individuals' attributes and behaviors when computing the similarities. There is also a number of works leveraging this approach, e.g., the tensor factorization method[55]. Again, considering their computational complexity and the availability of datasets at the current phase of the project, we opt to leave them out of the list for now and will investigate these methods in the later phase of the project when more data is available.

[48] https://en.wikipedia.org/wiki/Jaccard_index

[49] Adamic, Lada A., and Eytan Adar. "Friends and neighbors on the web." Social networks 25.3 (2003): 211-230.

[50] Barabâsi, Albert-Laszlo, et al. "Evolution of the social network of scientific collaborations." Physica A: Statistical mechanics and its applications 311.3-4 (2002): 590-614.

[51] M. E. J. Newman. Clustering and preferential attachment in growing networks. Physical Review Letters E, 64(025102), 2001.

[52] Easley, David, and Jon Kleinberg. Networks, crowds, and markets. Vol. 8. Cambridge: Cambridge university press, 2010.

[53] Lü, Linyuan, and Tao Zhou. "Link prediction in complex networks: A survey." Physica A: statistical mechanics and its applications 390.6 (2011): 1150-1170.

[54] Soundarajan, Sucheta, and John Hopcroft. "Using community information to improve the precision of link prediction methods." Proceedings of the 21st International Conference on World Wide Web. 2012.

[55] Dunlavy, Daniel M., Tamara G. Kolda, and Evrim Acar. "Temporal link prediction using matrix and tensor factorizations." ACM Transactions on Knowledge Discovery from Data (TKDD) 5.2 (2011): 1-27.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Evaluation: The performance of link prediction methods are often evaluated by their ability in recovering the true links that are purposely hidden. That is, for given individuals, we first hide some small portion of their links, and then use the prediction methods to recover the hidden links. The performance is often measured by metrics like precision and recall at top ranked predicted links[56].

2.1.4 Outlier detection

Objective: Generally, in a large dataset, an outlier is defined as a data point which is very *different* from the rest of the dataset[57]. Here, the difference is often determined by some statistical measures that are specifically tailored to the context and domain of the dataset. Those outliers often contain useful information on irregular elements or strange behavior of the system or population that the dataset describes. In network data, outliers include individuals that exhibit abnormal attributes or interactions/relations with other individuals, as well as strange behaviors that are not often performed by certain individuals. In this task, we would like to identify such individuals and behaviors. Precisely, given an input network, we aim to produce anomaly ranks for individuals in the network and their behaviors, then output the top ranked ones.

Applications in ROXANNE: Outlier detection methods have been widely applied for fraud detection[58], crime investigation[59], and most recently for spotting terrorist activities[60]. These studies show that outlier detection approach has a potential for identifying suspicious individuals and interactions that the investigators should pay more attention to. On one hand, this would form a warning system detection and prevention of criminal events. On the other hand, it would also reduce manual effort of the investigators by prioritizing the most potentially more suspicious cases.

Methods. In principle, efficient outlier detection methods require domain knowledge and understanding of the input datasets. Since each method relies on a particular statistical measure that is specifically designed to the context and domain of the data, there are no universal methods that work practically well on all the datasets. For general network data, the following approaches are widely adopted in existing methods[61].

- **Feature-based approach**: methods in this approach work by first extracting the feature vector for each individual in the network, and then identifying the outliers by invoking some multivariate outlier detection, e.g., using Mahalanobis distance[62]. Here the features are handcrafted based on network structures[63], individuals' attributes[64], or the temporal dynamic of the network[65].

- **Proximity-based approach**: methods in this approach work by first extracting proximity or similarity among individuals, and then returning as outliers the individuals that are least similar to others. Typical similarities for

[56] Liben-Nowell, David, and Jon Kleinberg. "The Link-Prediction Problem for Social Networks." Conference on Information and Knowledge Management (CIKM'03). 2003.

[57] Charu C. Aggarwal, Outlier Analysis, Springer, Cham, 2017

[58] Bolton, Richard J., and David J. Hand. "Statistical fraud detection: A review." Statistical science (2002): 235-249.

[59] Lin, Song, and Donald E. Brown. "An outlier-based data association method for linking criminal incidents." Decision Support Systems 41.3 (2006): 604-615.

[60] Wang, Pei-Chi, and Cheng-Te Li. "Spotting Terrorists by Learning Behavior-aware Heterogeneous Network Embedding." Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019.

[61] Akoglu, Leman, Hanghang Tong, and Danai Koutra. "Graph based anomaly detection and description: a survey." Data mining and knowledge discovery 29.3 (2015): 626-688.

[62] Charu C. Aggarwal, Outlier Analysis, Springer, Cham, 2017

[63] Akoglu, Leman, Mary McGlohon, and Christos Faloutsos. "Oddball: Spotting anomalies in weighted graphs." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2010.

[64] Perozzi, Bryan, and Leman Akoglu. "Scalable anomaly ranking of attributed neighborhoods." Proceedings of the 2016 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2016.

[65] Akoglu L, Faloutsos C (2010) Event detection in time series of mobile communication graphs. In: Proceedings of army science conference

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

simple networks include the ones that are mentioned in Section 2.1.3. Certainly, these similarities can be vastly enhanced by incorporating individuals' attributes and the network evolution information[66].

- **Community-based approach**: methods in this approach work by first finding communities in the input network, then spotting as outliers the individuals or the interactions/nodes that span the communities. That is, outliers are defined as the ones that do not firmly belong to any particular community. For simple networks, in the first step -- community detection, methods mentioned in Section 2.1.2 can be used. Again, these methods can also be improved significantly by incorporating individuals' attributes[67] and the network's dynamicity[68]

The above list of approaches re-emphasizes that the performance of any outlier detection method vastly depends on the insights obtained from inspecting and analysing the input datasets. Given the unavailability of appropriate datasets at the current phase of the project, we have decided to put more effort into other network analysis functionalities than on this task. It should be more reasonable to investigate these aforementioned methods more comprehensively in the later phase of the project when more data is available.

Evaluation: Again, as totally unsupervised and there is often no ground-truth available in reality, outlier detection methods can mostly be evaluated qualitatively. The qualitative evaluation includes in-depth manual examination the detected individuals and behaviors, and comparing them with the rest of the network. Another typical technique for evaluating the methods is using synthetic datasets with injected outliers. That is, given a network with mostly no outlier, some individuals and interactions are added or changed into outliers. The detection methods are then evaluated by their ability in detecting the injected ones, and their performance is often measured by metrics like precision and recall at top-ranked outliers[69].

2.1.5 Network embedding

Objective: Most network analysis functionalities, particularly the ones mentioned in the previous subsections, comprise a feature engineering step to convert each element (i.e., individual or interaction/relation) into a vector that represents the element. Those vectors, also called feature vectors, had been traditionally handcrafted using domain knowledge or comprehensive statistical analysis. Hence, the feature engineering step is highly expensive or time-consuming, as it requires much domain expertise. Network embedding is a recently proposed approach for addressing this shortcoming[70]. As depicted in Figure 4, this approach allows us to learn the feature vectors without or with very little supervision. Precisely, given an input network, and optionally, some supervision information (e.g., roles or labels for some small proportion of individuals in the network), node embedding methods can automatically produce feature vectors, now also called embedding vectors, for each individual in the network.

[66] Manzoor, Emaad, Sadegh M. Milajerdi, and Leman Akoglu. "Fast memory-efficient anomaly detection in streaming heterogeneous graphs." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.

[67] Perozzi, Bryan, and Leman Akoglu. "Discovering communities and anomalies in attributed graphs: Interactive visual exploration and summarization." ACM Transactions on Knowledge Discovery from Data (TKDD) 12.2 (2018): 1-40.

[68] Baingana, Brian, and Georgios B. Giannakis. "Joint community and anomaly tracking in dynamic networks." IEEE Transactions on Signal Processing 64.8 (2015): 2013-2025.

[69] Akoglu, Leman, Hanghang Tong, and Danai Koutra. "Graph based anomaly detection and description: a survey." Data mining and knowledge discovery 29.3 (2015): 626-688

[70] Cui, Peng, et al. "A survey on network embedding." IEEE Transactions on Knowledge and Data Engineering 31.5 (2018): 833-852.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

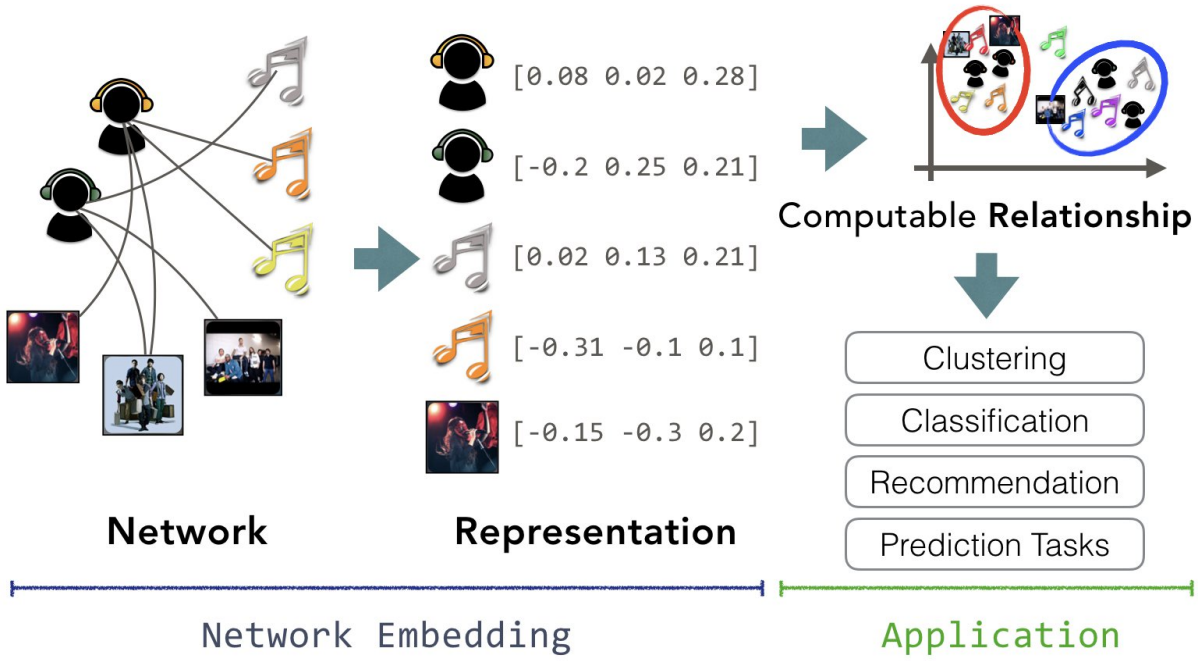


Figure 4. Node embedding framework and its applications[71]

Applications in ROXANNE: Within ROXANNE, the application of network embedding methods is two-fold. On the one hand, these embedding vectors would help us to perform other network analysis functionalities more effectively. Those vectors have been shown to be able to significantly improve the performance of individual classification, community detection, and link prediction[72]. On the other hand, the vectors would also be used as a feedback information of network analysis components (WP6) to help improving the performance of the speech and text analysis components (WP5) within ROXANNE's pipeline systems.

Methods: Network embedding has been attracting much interest from researchers of many fields due to its usefulness in a wide range of applications. There has been a number of embedding methods proposed for networks found in different domains and contexts. In the following, we summarize the methods that are most closely related and applicable in ROXANNE.

- **Factorization - based methods**[73]: these methods work by first forming some matrix for representing the input network (e.g., the adjacency matrix[74]), then performing a low-rank approximation[75] on the obtained matrix. This approximation can be done by using, e.g., non-negative matrix factorization[76], or singular value decomposition[77]

- **Random walk - based methods**: inspired by pioneering works in embedding of words in text documents, these methods work by first conducting a number of fixed-length random walks on the input network, then considering

[71] Image taken from https://github.com/chihming/awesome-network-embedding

[72] Cui, Peng, et al. "A survey on network embedding." IEEE Transactions on Knowledge and Data Engineering 31.5 (2018): 833-852.

[73] Liu, Xin, et al. "A general view for network embedding as matrix factorization." Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 2019.

[74] Harary Frank, Graphs and Matrices, SIAM Review, Vol 9, n 1, p.83-90, 1967

[75] https://en.wikipedia.org/wiki/Low-rank_approximation

[76] https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

[77] https://en.wikipedia.org/wiki/Singular_value_decomposition

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

each path as a sentence and each individual as a word, and lastly invoking some word embedding model, e.g., *word2vec* model[78], for learning the embedding vector of the individuals. Methods in this approach differ from each other by the techniques used for conducting the random walks, e.g., by first order [79],[80] or second order[81] Markov processes[82].

- **Deep neural network - based methods**: these methods work by employing an encoder-decoder multilayer neural network for learning the individuals' embedding vectors in an autoencoder[83] manner. They differ from each other by the structure of the neural networks used, e.g., either a multilayer perceptron network[84], or a multilayer convolutional network[85],[86].

Most of the methods cited above are purely based on the structure of the input network. Nevertheless, they can be easily extended to incorporate more individuals' and interactions/relations' attributes. There are also a number of works on the incorporating of such information, e.g., individuals' roles in the network[87], or the heterogeneity of the interactions/relations[88].

Evaluation: As suggested in Figure 4, the effectiveness of the embedding methods are mostly evaluated by the performance of their embedding vectors in the downstream tasks. Other aspects in evaluating these methods include the scalability and the robustness[89].

2.2 Systems for network analysis

In this section we present a brief overview of existing systems and frameworks for manipulating, visualizing and performing network analysis functionalities. We focus on open-source and publicly available ones so as to increase their re-usability and reproducibility in ROXANNE and reduce the implementation cost. We start by reviewing generic systems and frameworks for open-domain network analysis. We then review the ones that are specifically developed for networks in the crime domain.

[78] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

[79] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.

[80] Tang, Jian, et al. "Line: Large-scale information network embedding." Proceedings of the 24th international conference on the world wide web. 2015.

[81] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.

[82] https://en.wikipedia.org/wiki/Markov_chain

[83] https://en.wikipedia.org/wiki/Autoencoder

[84] Wang, Daixin, Peng Cui, and Wenwu Zhu. "Structural deep network embedding." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.

[85] Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." Advances in neural information processing systems. 2016.

[86] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).

[87] Ahmed, Nesreen K., et al. "Learning role-based graph embeddings." arXiv preprint arXiv:1802.02896 (2018).

[88] Dong, Yuxiao, Nitesh V. Chawla, and Ananthram Swami. "metapath2vec: Scalable representation learning for heterogeneous networks." Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017.

[89] Dai, Quanyu, et al. "Adversarial network embedding." *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

2.2.1 Generic analysis systems

Programming libraries and frameworks: As network data is prevalent in daily contexts, there have been a number of programming libraries and frameworks developed for providing network analysis functionalities. Among those libraries and frameworks, the typical ones are as follows.

- **NetworkX**[90]: is a free, scalable Python package for the creating, manipulating, and analyzing networks. This package provides a wide range of functions for computing centrality measures, community detection, and link prediction.
- **Zen**[91]: is a free Python package for the creating, manipulating, and visualizing networks. This package also provides some functions for network structural analysis, e.g., modularity based community detection
- **igraph**[92]: a free package with versions for multiple programming languages. Similar to Zen, this package is mainly for creating, manipulating, and visualizing networks
- **graph-tool**[93]: is a free, efficient Python package for manipulating and performing statistical analysis on networks. This package provides some functions for community detection using mixed-membership and hierarchical methods.
- **NetworkKit**: another free package with C++ and Python versions. This package provides a wide range of functions for community detection and link prediction.
- **Stanford's SNAP**[94]: another free package with C++ and Python versions for creating, manipulating and analyzing networks. This package focuses on functions for computing individuals' centrality measures and community detection.
- **JUNG**[95]: a free Java package for analyzing and visualizing networks. This package provides some functions for computing centrality measures and community detection.
- **Karate Club**[96]**:** a free Python package that is built on top of NetworkX for unsupervised learning on networks. This package provides a wide range of functions for community detection and network embedding.
- **Pytorch BigGraph**[97]: the Python package recently released by Facebook for performing network embedding at very large scales.

Software: Similarly, there are also compact softwares that are dedicatedly developed for network analysis. The notable ones are as below:

- **NodeXL**[98]: a network analysis and visualization software package for Microsoft Excel with the free option for basic usage. This software provides functions for computing centrality measures.

[90] https://networkx.github.io/
[91] http://zen.networkdynamics.org/
[92] https://igraph.org/
[93] https://graph-tool.skewed.de/static/doc/index.html
[94] https://snap.stanford.edu/index.html
[95] https://jrtom.github.io/jung/
[96] https://karateclub.readthedocs.io/en/latest/
[97] https://github.com/facebookresearch/PyTorch-BigGraph
[98] https://nodexl.com/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- **Gephi**[99]: an open-source Java program for manipulating, visualizing, and analyzing networks. This software provides functions computing centrality measures and community detection.
- **Tulip**[100]: a free C++ program for visualizing and analyzing networks. This software provides some basic functions for community detection

Commercial software: Companies and industries have also been developing commercial softwares for working with network data. The following are currently notable ones in the market.

- **Palantir Gotham**[101]**:** a framework for integrating, visualizing, and analyzing data from multiple sources. For network analysis in particular, this framework provide functions for social influence analysis and community detection
- **Sentinel Visualizer**[102]: a software for visualizing and analyzing networks with functions for social influence analysis and community detection
- **Visallo**[103]**:** a software for visualizing networks with functions for link and path discovery
- **Vizkey**[104]**:** a software for manipulating and visualizing networks with functions for community detection and link and path discovery

2.2.2 Criminal network analysis systems

The following software platforms are the State-of-the-Art of criminal analysis systems that are free to acquire, are licensed and already used by a lot of LEAs mostly in the USA.

- **IBM i2 Analyst's Notebook**[105]: a commercial software for pattern analysis in structured and unstructured data using visualisation. This software provides some basic functions for social influence analysis
- **DataWalk**[106]**:** a framework for integrating and analyzing data from multiple sources with social influence analysis functions.
- **Linkurious**[107]**:** a framework for integrating and analyzing data from multiple sources with functions for outlier detection
- **Siren**[108]**:** a framework for visualizing data from multiple sources with functions for link prediction
- **ClueMaker**[109]**:** a framework for manipulating and visualizing networks with functions for link and path discovery

[99] https://gephi.org/
[100] https://tulip.labri.fr/TulipDrupal/
[101] http://www.palantir.com/palantir-gotham/
[102] http://www.fmsasg.com/
[103] https://www.visallo.com/
[104] http://www.vizkeyweb.com/?lang=en/
[105] https://www.ibm.com/products/analysts-notebook
[106] https://datawalk.com/
[107] https://linkurio.us/
[108] https://siren.io/
[109] https://cluemaker.com/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- **Link Explorer[110]:** a software for visualizing and analyzing networks with functions for social influence analysis
- **Maltego[111]:** a software for constructing and visualizing networks from multiple data sources with social influence analysis

Additionally, there are also other softwares and programming packages for forensic and security that make use of networks for representing data. Examples are

- **Crimestat**[112]: a spatial statistics program for the analysis of crime incident locations. CrimeStat is Windows-based and interfaces with most desktop GIS programs.
- **CrimeCenter**[113]: a complete case management system designed by law enforcement experts.
- **COPLINK[114]**: a data sharing and crime analytics platform that is designed to help LEAs solve crimes faster by providing tactical, strategic and command-level access to vast quantities of seemingly unrelated data.
- **crimelinkage[115]**: a package to help crime analysts and researchers with tasks related to crime linkage. This package specifically includes methods for criminal case linkage, crime series identification and clustering, and suspect identification.

From the discussion in Section 2.2, we observe that the existing systems and frameworks, both free/open-sources and commercial ones, may provide some but not all the necessary functionalities, as mentioned in Section 2.1. This gives us more incentives to develop an open-source framework for providing end-users with all these functionalities. We would also like the framework to be highly extensible and adaptable to increase its usage in different contexts.

[110] https://www.xanalys.com/products/link-explorer/
[111] https://www.maltego.com/
[112] https://www.icpsr.umich.edu/CrimeStat/
[113] https://crimecenter.com/
[114] https://forensiclogic.com/coplink/
[115] https://cran.r-project.org/web/packages/crimelinkage/index.html

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

3. Network analysis technologies in ROXANNE

In ROXANNE, the network construction and analysis components will be developed in conjunction with the speech/text/video technologies in WP5, and these components will be integrated into the common frameworks developed in WP7. Precisely, as depicted in Figure 5, our components will take as input the output from WP5' components, which are JSON objects that describe the interactions among individuals that are extracted from raw data. Using data fusion technologies, we construct the networks from these interactions. Next, components for network analysis and visualization will be used for working with the constructed networks. The result from these components can be sent to end-users, through some interfaces within WP7, or sent back to WP5 as reinforcement feedback for its components. For example, community detection may help determine how specific language is propagated among members of a clique and between one clique to another, and other network measures may help to improve the performance in speaker identification[116].

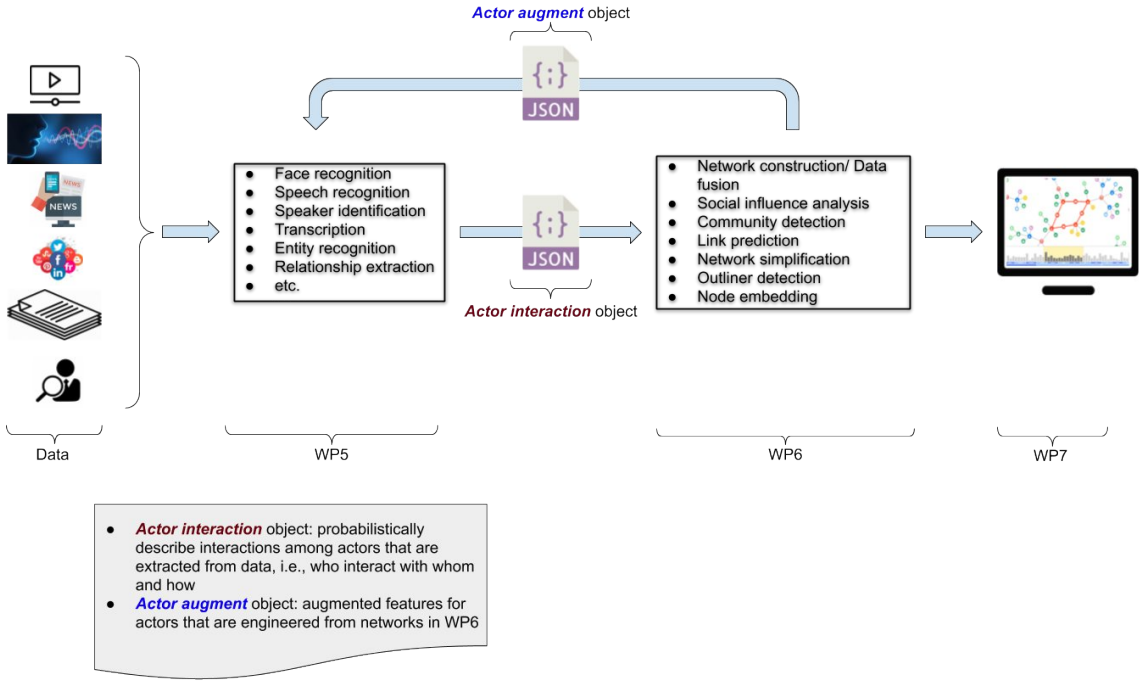


Figure 5. WP6's components within ROXANNE's information flow

We now describe the systems that have been developing in WP6 in detail. We adopt an agile approach to developing these systems. That is, we leverage existing open-source frameworks and publicly released implementations for network analysis methods to quickly build up the first versions of the systems that are able to provide expected functionalities. These first versions would help us to gain useful feedback from partners and end-users while iteratively and gradually improve the system's performance and user experience.

[116] Gao, Ning, et al. "Leveraging side information for speaker identification with the Enron conversational telephone speech collection." 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

3.1 Data fusion framework

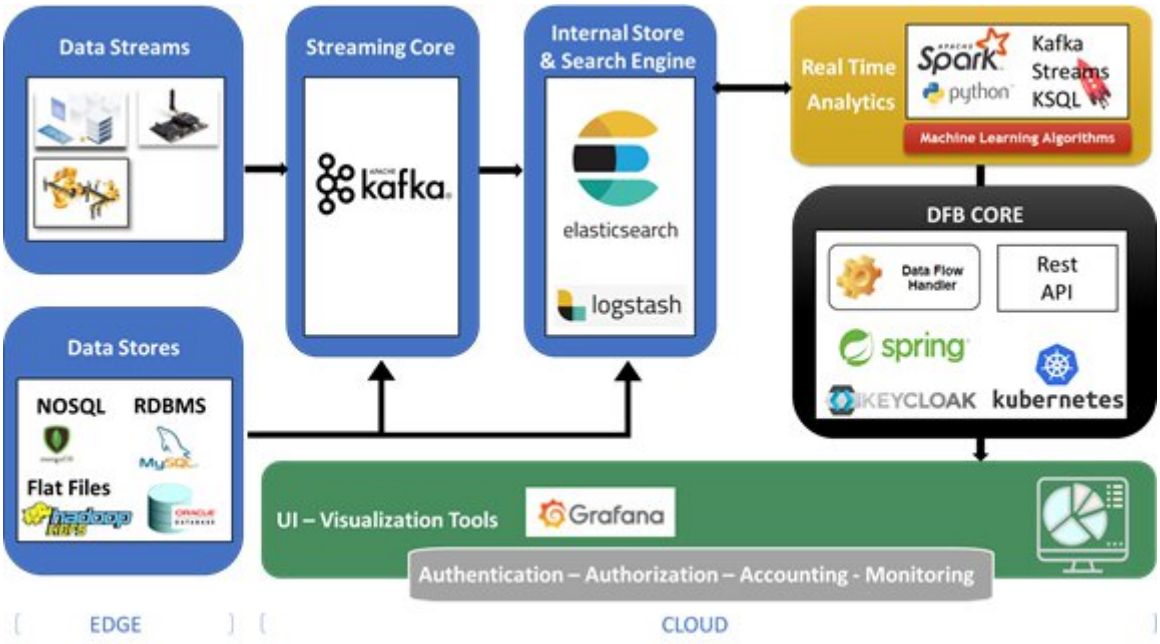
The data fusion framework in ROXANNE builds upon the Data Fusion Bus[117] (DFB) platform which enables organizations in developing, deploying, operating and managing a big data environment with an emphasis on real-time applications. It combines the features and capabilities of several big data applications and utilities within a single platform.

The key capabilities of DFB are:

- Real-time monitoring and event-processing, semantic fusion of events not coinciding in time.
- Data aggregation from heterogeneous data sources and data stores.
- Real-time analytics, offering ready to use Machine Learning algorithms for classification, clustering, regression, anomaly detection.
- An extendable and highly customizable Interface (REST API[118] and Web app) for configuring analytics, manipulation and filtering. It also includes functionality for managing the platform.

DFB will be used in ROXANNE for:

- Aligning data streams for time and granularity: Event (e.g. landmark reference, Name reference) - time merge of different Kafka streams
- Providing granularity by creating aggregations in live or historical datasets when required (min, max, average, distinct values, etc.)
- Providing ML support to network and relation analysis activities of WP6: (i) off-line creation of ML models based on historical data, (ii) real-time clustering and classification of events, (iii) persistent storage of metadata and (iv) exporting real-time results to other services through Kafka streams.



[117] https://www.itml.gr/data-fusion-platform

[118] https://en.wikipedia.org/wiki/Application_programming_interface

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Figure 6. Data Fusion Bus (DFB)

The main building blocks of DFB to be used in ROXANNE are:

- Data Analytics: DFB supports batch processing and stream processing with Kafka Streams & KSQL[119], Apache Spark[120] and Spark Streaming[121].
- DFB Core: responsible for providing business logic to DFB and managing all the data flows. DFB Core exposes a REST API for configuring and running ML related tasks.

As illustrated in Figure 6, our DFB relies on Kafka for streaming data input/output and Elasticsearch[122] for persistent storage of data and metadata.

3.2 Network analysis system

We develop our components for network analysis and integrate them internally to form a system according to the structure shown in Figure 7. Specifically, our system has the following four main components

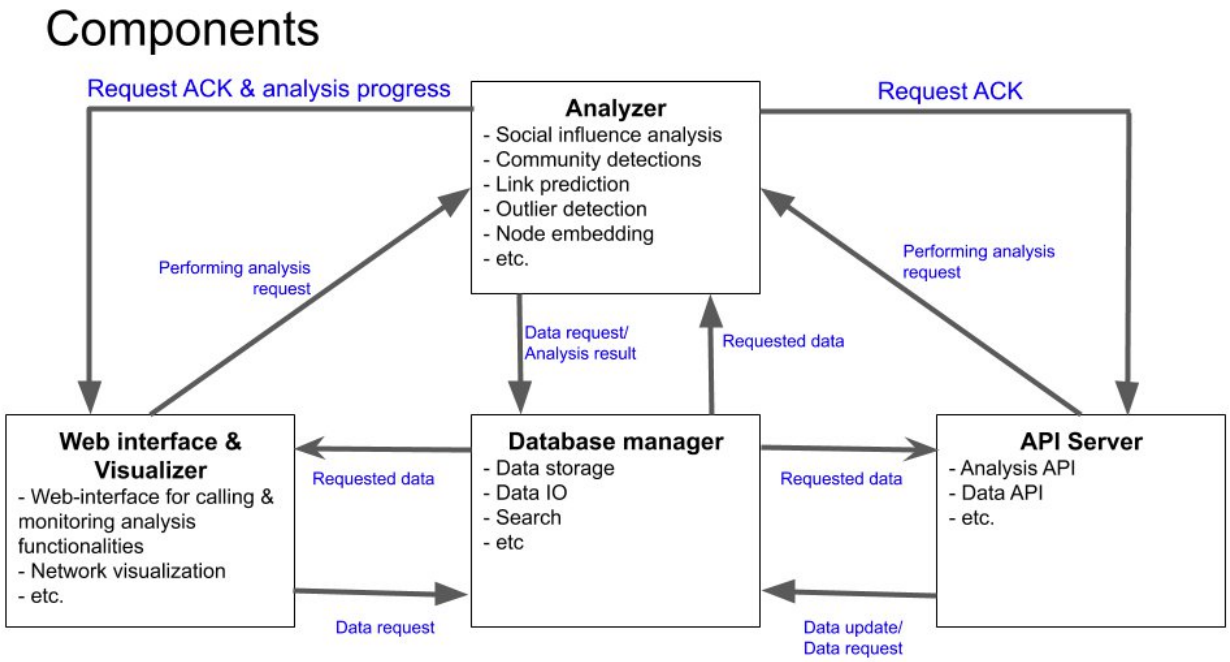


Figure 7. The internal structure of WP6's components

- The **Analyzer**: which is in charge of providing the necessary analysis functionalities as discussed in Section 2.1. This component receives analysis request from the **Web interfaces & Visualizer** or the **API Server** component, queries for the corresponding data from the **Database manager**, performs the requested analysis, and lastly returns the analysis results to the requester

[119] https://kafka.apache.org/documentation/streams/
[120] https://spark.apache.org/
[121] https://spark.apache.org/streaming/
[122] https://www.elastic.co/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- The **Web interfaces & Visualizer**: a component mainly for end-users to visually examine networks. a web-based application that allows end-users to import, manipulate**,** and perform analysis functionalities on networks. This component receives users' requests through web interfaces, then requests for the corresponding data from the **Database manager** and the corresponding analysis from the **Analyzer**, and lastly visualizes the analysis results on web interfaces.

- The **API Server**: a component that allows our system to be integrated easily to any other larger systems. This server provides APIs for constructing and analyzing networks similar to the **Web interfaces** & **Visualizer** except that the communication with this component is now performed through the API calls. That is, this component receives users' requests through API calls, then requests for the corresponding data from the **Database manager** and the corresponding analysis from the **Analyzer**, and lastly returns the analysis results through some network communication channel.

- The **Database manager**: is basically a graph database management system that provides the functionalities for storage, manipulating, and querying network data.

For implementing the system, we opt to use Python as the primary programming language. This is mainly to help us to fully leverage existing open-source and free packages and framework for network analysis, as discussed in Section 2.2. In the following subsections, we describe in detail the implementation and current status of each component.

3.2.1 Database manager

We have investigated a number of open-source frameworks for graph database management and others for heterogeneous data as well. We determined to use **JanusGraph**[123] for implementing the **Database manager** component. This decision was made due to the scalability and flexibility of **JanusGraph** as well as the wide range of query functionalities that it supports. Particularly, **JanusGraph** can work with various data storage schemes and frameworks, e.g., Apache HBase, Google Cloud Bigtable, etc., and provides search capabilities like full-text via widely used frameworks like Elasticsearch or Apache Lucene. Moreover, it also supports **Gremlin**[124] -- the network-specific query language.

Additionally, for flexibility and to increase the usage of our system when deployed in contexts with not too large datasets or with limited infrastructure, we also implement an in-memory database manager that provides all necessary functions for the system. This in-memory database manager is designed to work smoothly and transparently with other components.

3.2.2 Analyzer

Our analyzer is built on top of a wide range open-source and free programming libraries and frameworks as discussed in Section 2.2.1. Currently, this component provides the following network analysis functionalities:

- **Social influence analysis**: with options for all the measures discussed in Section 2.1.1, i.e., closeness, betweenness, pagerank and authority

- **Community detection**: with with options for all the methods discussed in Section 2.1.2, i.e., K-clique, modularity maximization, spectral, and hierarchical clustering methods

- **Link prediction**: with with options for all the methods discussed in Section 2.1.3, i.e., Jaccard, Adamic Ada, preferential attachment, resource allocation and Soundarajan-Hopcroft methods

[123] https://janusgraph.org/

[124] https://tinkerpop.apache.org/gremlin.html

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- **Network embedding**: with with options for most of the methods discussed in Section 2.1.5 matrix factorization, Deepwalk, Node2vec, and LINE methods

For the remaining time of the project, we will investigate and integrate into this component functions for performing outlier detection as discussed in Section 2.1.4. Those functions require methods to be more specifically tailored for the domain. Hence, as mentioned earlier, we leave it for the later phase of the project when more dataset is available.

Equivalently, we will constantly improve the performance of all the functionalities by investigating and adapting state-of-the-art methods and others that will be developed as the project progresses.

3.3.3 API server

For implementing the API server, we adopt the **Flask**[125] web framework. Compared to its alternatives, e.g., django or Bottle, and this framework is considered more simple to use and with more explicit workflow, so as to make our implementation easier to maintain.

Currently, our API server provides the following functions:

- **User authentication**: to authenticate users and get token for using subsequent API calls
- **Moderating**: to manage users and their permission on data
- **Data querying**: to query data from existing database
- **Network manipulating**: to create and manipulate networks
- **Analysis**: to perform analysis on an existing network in database or a provided one

In the future, we will continue to add more API functions to the components. These functions are defined and developed based on requirements from both end-users and other components in ROXANNE.

3.3.4 Web interfaces and Visualizer

Similar to the API server, for implementing the Web interfaces and Visualizer, we also adopt **Flask**[126] web framework. Additionally, for the visualization of networks, we employ the **Dash**[127] package. This package is built on top of Flask and has been widely used for developing highly interactive interfaces for visualizing data.

Currently, our visualizer provides the following functions:

- **Data importing**: to import networks from database or local files, and to select sub-network surrounding a set of individuals
- **Network visualization**: display the selected (sub-) networks, to show/hide the selected individuals of certain types, to highlight relations of selected individuals
- **Analysis**: to perform social influence analysis, community detection, or link prediction and display the results. For example, in Figure 8, we show the result of social influence analysis on a network in which the individuals' darkness represents the importance score. Similarly, in Figure 9, we show the result of community detection on the same network where its color denotes each individual's community. Lastly, in Figure 10, we show the link prediction result for an individual in the network where predicted links are drawn in red.

[125] https://flask.palletsprojects.com/en/1.1.x/
[126] https://flask.palletsprojects.com/en/1.1.x/
[127] https://dash.plotly.com

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

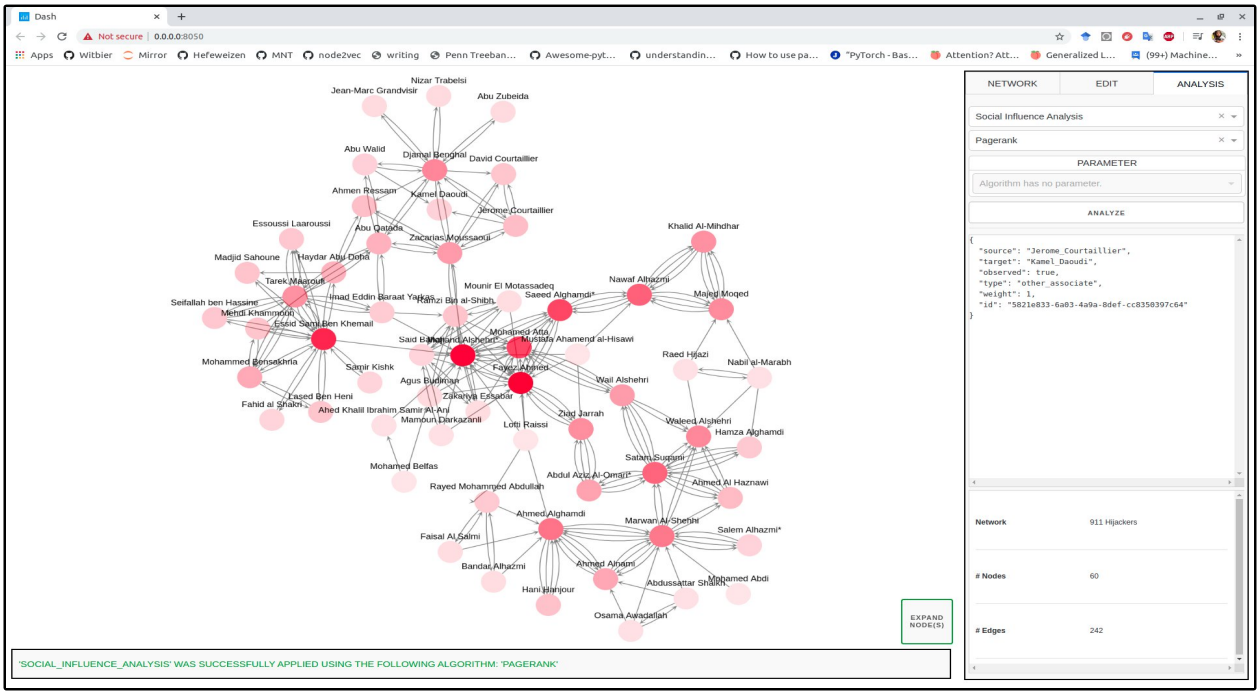


Figure 8. Example visualization of social influence analysis: darker nodes have higher importance score

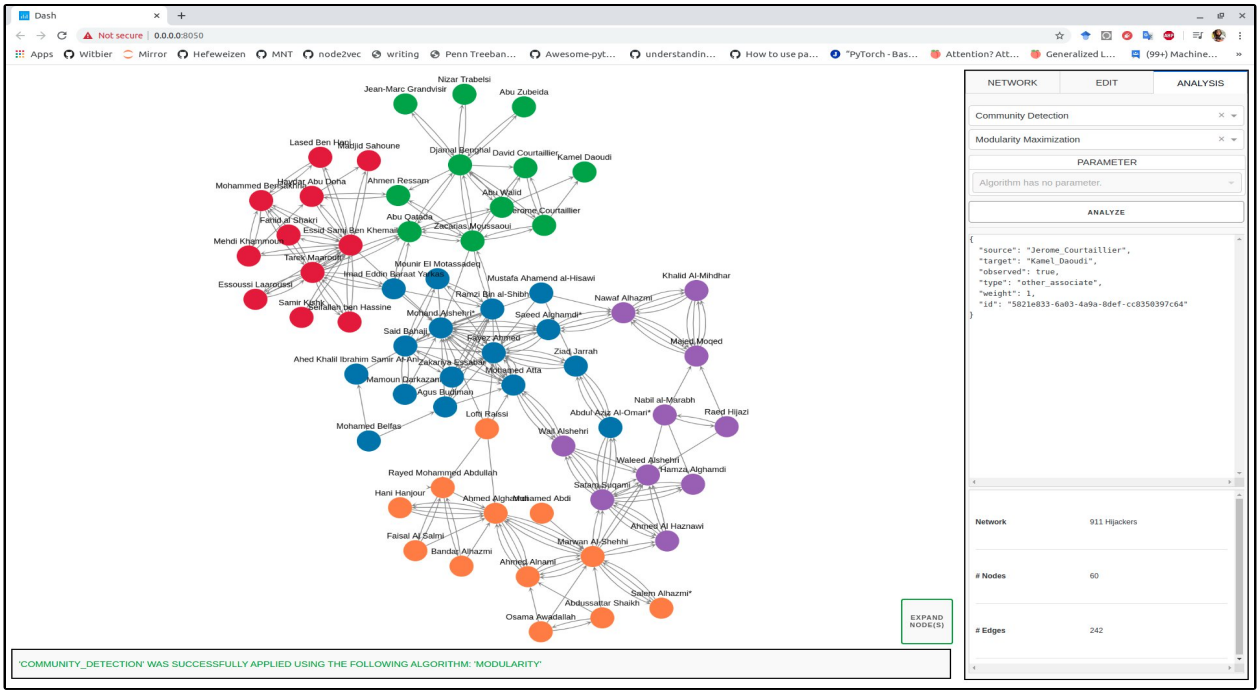


Figure 9. Example visualization of community detection visualization: individuals of the same color belong to the same community

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

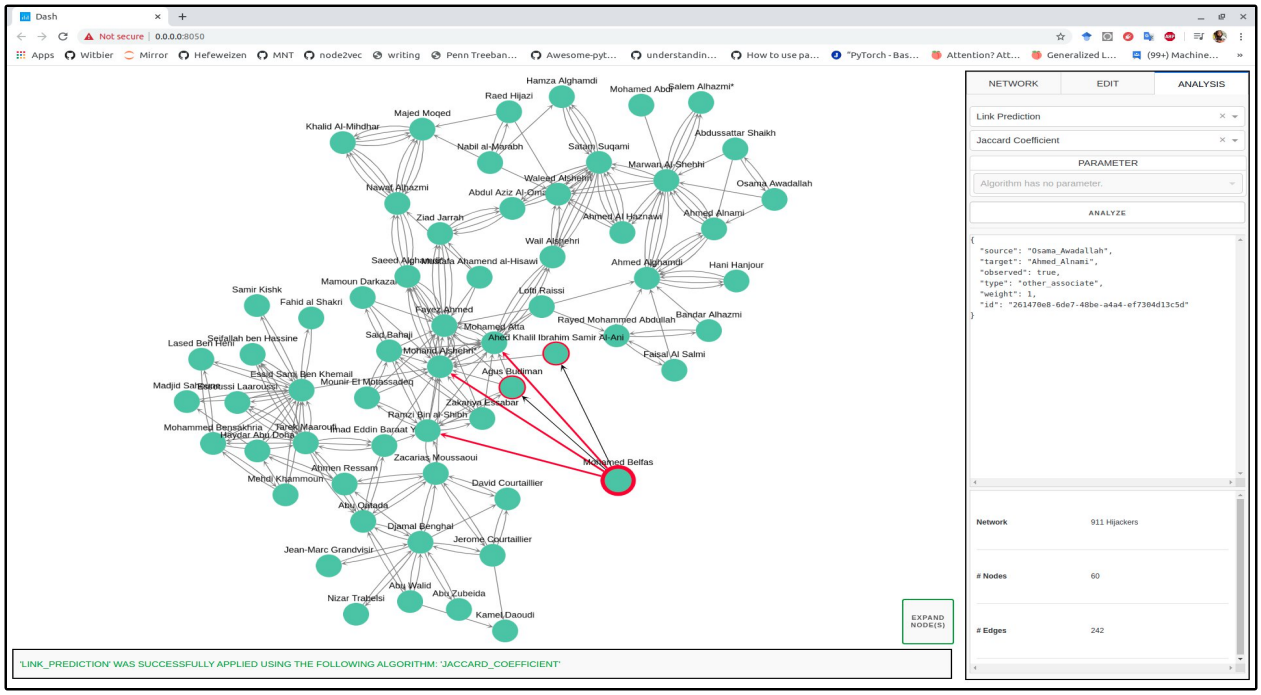


Figure 10. Example visualization of link prediction visualization: predicted links are in red

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

4. Case studies on ROXANNE datasets

As real datasets are not yet available at the current phase of the project, while the simulated data generation work is in progress (as detailed in the deliverables D4.1 Overview and analysis of lawfully intercepted and publicly available data), we have investigated a wide range of public datasets for developing and testing the network analysis system described above. This is conducted in conjunction with WP4's tasks on inventorying data needed for the whole ROXANNE project. Currently, the following datasets were collected and have already integrated into the system.

- **Covert networks**: this is a collection of past cases' networks that are collected and made publicly available by researchers from the University of Manchester[128]. Here, we focus on a subset of the collection where the network among the involved individuals is available along with some of their attributes (e.g., name, gender, role, etc). These include the networks of Madoff fraud[129] case and 911 attack[130]. The advantage of these datasets is that they refer to criminal activities of various types. At the same time, however, the relational information is limited to contacts and other forms of interactions. There is no audio/video material that can be exploited for extraction of additional information in ROXANNE WP5
- **CSI** networks: this is a collection of conversational networks among characters selected episodes of the Crime Scene Investigation[131] series. Detailed information of this dataset is given in Section 4 of the deliverable D5.1 Initial speech/text/video technologies. Partners in WP5 have run extensive experiments on this dataset and helped to extract "who talked to whom" information from the videos. Using this information, we have constructed a network among 14 main characters of an episode (Season 1, Episode 7) with 60 links, in which each link means its two characters have been involved in some conversation. Another similar network is also extracted from the transcripts (as published by researchers in Edinburgh University[132]) with 30 characters of the same and 107 links.

- **Enron** network: a dataset among a group of employers in Enron who have exchanged some email or involved in some phone calls. Similarly, detailed information of this dataset is given in Section 4 of the deliverable D5.1 Initial speech/text/video technologies. Preprocessing steps in WP5 have helped to extract from this dataset a network of 69 employers and 350 communication links (i.e., email or phone call) among them.

- **NIST** datasets: consists of two clusters of the phone call network dataset that was published under NIST Speaker Recognition and Evaluation and acquired by the ROXANNE partners. These two networks have 2044 nodes & 5866 links and 354 nodes & 468 links, respectively.

In the following, we give some case studies for demonstrating the usage of our systems when applied on CSI and Enron networks. We omit **NIST** networks in these studies as they do not contain meta information for nodes involved that allows us to conduct an evaluation at this phase of the project.

4.1 Madoff fraud network

This is the network that depicts the finance flows in the Madoff case[133]. The individuals in the network are the involved financial institutions and firms and the links are money flows. There are 61 individuals and 61 links in this dataset.

[128] The whole collection can be downloaded from https://sites.google.com/site/ucinetsoftware/datasets/covert-networks

[129] https://en.wikipedia.org/wiki/Madoff_investment_scandal

[130] https://en.wikipedia.org/wiki/September_11_attacks

[131] https://en.wikipedia.org/wiki/CSI:_Crime_Scene_Investigation

[132] Frermann, Lea, Shay B. Cohen, and Mirella Lapata. "Whodunnit? crime drama as a case for natural language understanding." Transactions of the Association for Computational Linguistics 6 (2018): 1-15.

[133] https://en.wikipedia.org/wiki/Madoff_investment_scandal

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

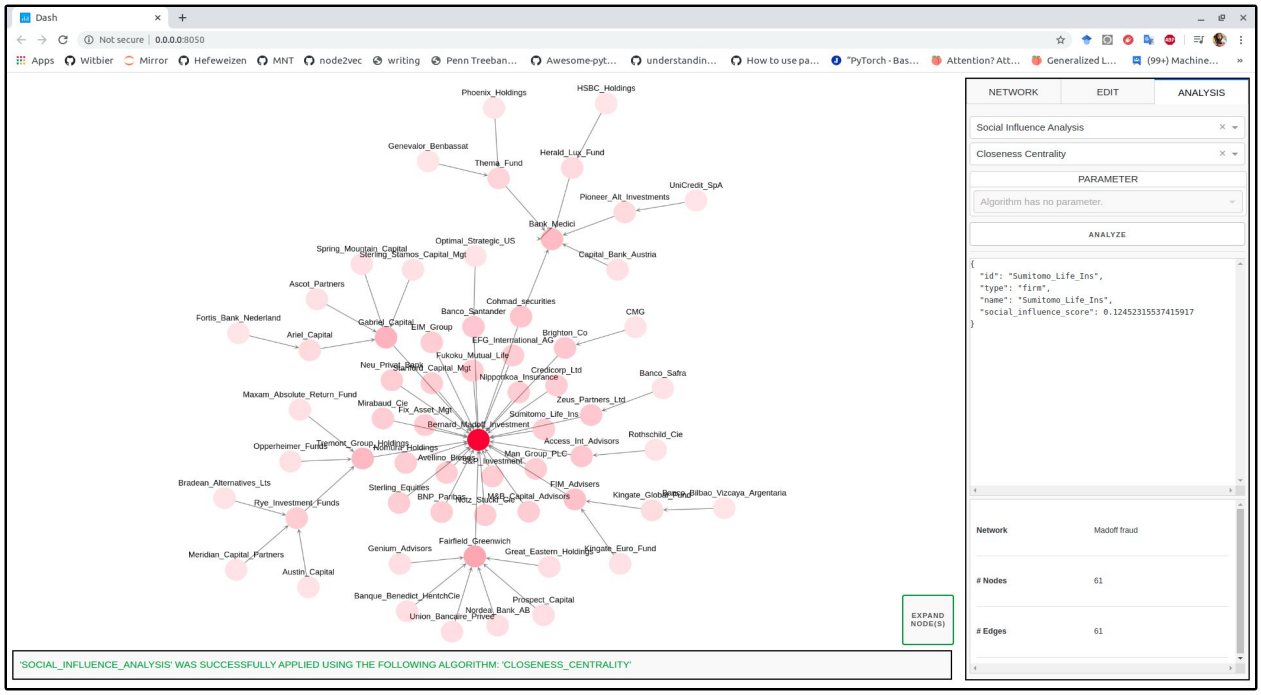


Figure 11. Social influence analysis on a Madoff fraud network using closeness measure

Figure 11 shows the visualization of social influence analysis on the network using the closeness measure. It can be observed from the figure that *Bernard_Madoff_Investment* is the most important individual in the network by this measure. That is reasonable and highly expected as this firm is the center that receives money credited from the rest of the network, and hence should be closest to all other individuals.

4.2 CSI network

Figure 12 shows the visualization of social influence analysis on a CSI network using the betweenness measure. It can be observed from the figure that *grissom* is the most important individual in the network by this measure. That is reasonable and highly expected as he is the leader of the investigation team and should be in the center of most conversations among the characters in the series

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

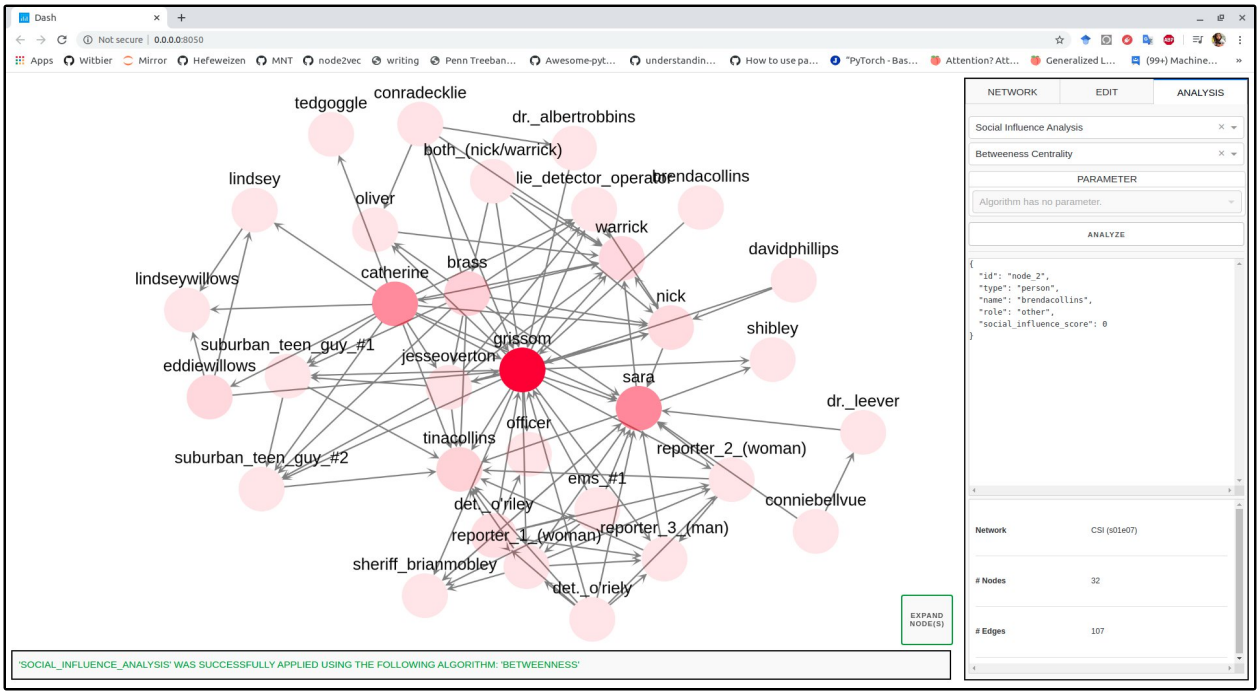
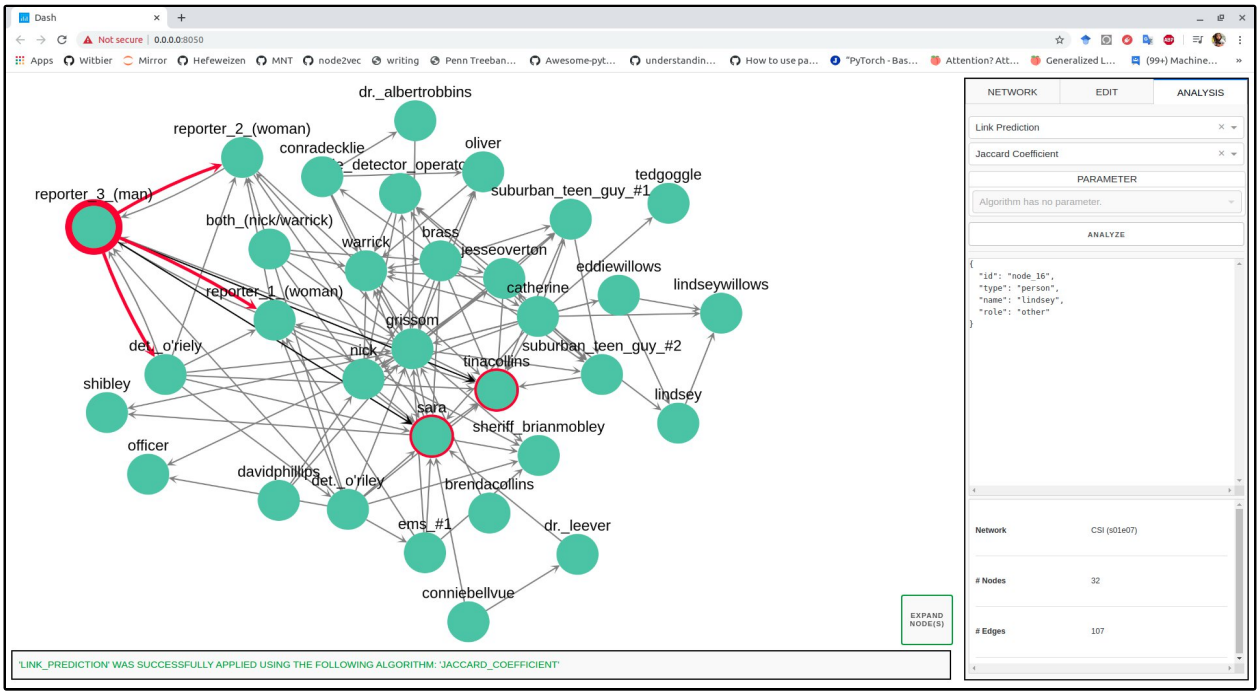


Figure 12. Social influence analysis on a CSI network using betweenness measure

Figure 13 shows the visualization of link prediction on a CSI network for *reporter_3* where the individual is predicted to have latent links with *reporter_1*, *reporter_2* and *det_o'riely*[134]. That is reasonable and highly expected that reporters have some unobserved interactions/relations when they are involved in the same business. It is also expected that the reporters have some connection with the detective for getting more information regarding their business.



[134] https://csi.fandom.com/wiki/Ray_O%27Riley

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

Figure 13. Link prediction on a CSI network

4.3 Enron network

Figure 14 shows the visualization of social influence analysis on the Enron network using the pagerank measure. It can be observed from the figure that *brian*[135]*, bill, jeremy,* and *jeff*[136] are the most important individuals in the network by this measure. That is reasonable and highly expected as they are all high level managers and should be reported to by other important individuals of the company.

Figure 15 shows the visualization of community prediction on the Enro. *nuke, brendan, muse, mike, muke, hats*, and *mt* are reasonably grouped into a single community as they are densely connected while there's mostly no connection with the rest of the network. Similarly, the same observation is applied *for sue, bob, gibran* and *neal*

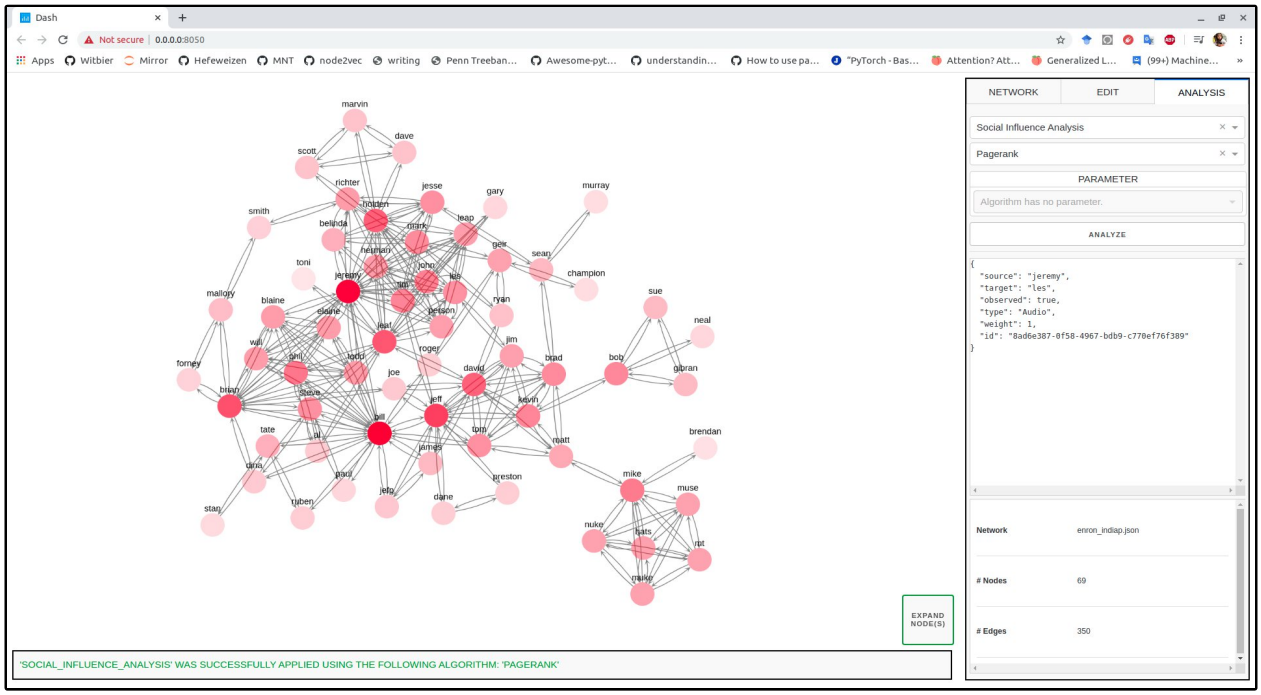


Figure 14. Social influence analysis on Enron network using pagerank measure

[135] https://www.linkedin.com/in/bkripley/

[136] https://en.wikipedia.org/wiki/Jeffrey_Skilling

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

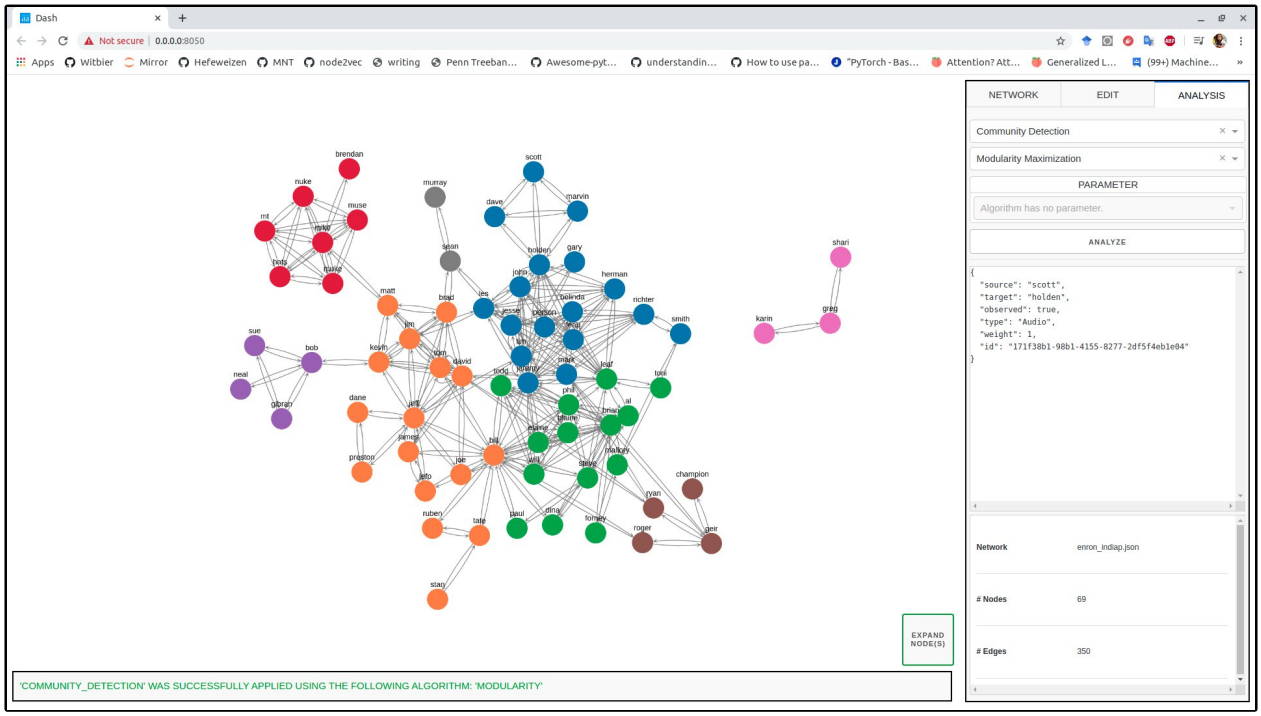


Figure 15. Community detection on Enron network

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635. No part of this document may be used, reproduced and/or disclosed in any form or by any means without the prior written permission of the ROXANNE project partners.

5. Preparation for the integrations

We now briefly describe our modules to be integrated into the ROXANNE platform. The technologies for the integration were determined through the discussion led by WP7 in M4-M5 of the project. In the following sections, we provide the important implementation of our modules. For more detail on the integration aspects, please refer to the respective WP7 deliverables.

The data fusion bus: DFB combines the features and capabilities of several big data applications and utilities within a single platform. However, in ROXANNE some of the key modules (such as Elasticsearch and Kafka) are already provided and are not required within DFB.

For the integration of DFB in the ROXANNE pipeline, orchestrated by Nifi, a dockerized version of DFB has been created, comprised of the following components:

- **DFB core**: a rest API for invoking the DFB functionalities. Since the exact functionalities required in ROXANNE are not defined yet, a docker image with basic functionalities has been prepared.
- **DFB analytics Engine**: the engine is based on Apache spark. For the integration, we prepared a custom docker image and configuration for invoking spark in the existing Kubernetes cluster on demand.

The network analysis systems: The first version of our network analysis system has already been released in the form of Docker[137] modules. These modules can work as all-in-one components, so as they can easily be integrated into ROXANNE's pipeline systems as well as to any other end-users' systems. These following two modules are ready-to-use and will been periodically updated

- **API Service module**: this module includes the API server, the analyzer, and the data manager, and provides APIs for network analysis functionalities as described in Section 3.2
- **Network visualizer**: this module includes web interface, visualizer, the analyzer, as well as the the data manager, and provides web-based interfaces for visualizing and and performing analysis on network as described in Section 3.2

We have also provided in the appendix of this deliverable the detailed documentation for setting up and usage of these modules.

[137] https://www.docker.com/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

6. On-going work and Future directions

Lastly, we describe our on-going work and sketch some directions for future work within WP6 or related to ROXANNE in general. These include (i) developing and enhancing our systems for network constructions and analysis, as well as on integrating these systems into ROXANNE platform; (ii) research on the embedding of uncertain networks, and (iii) systematic evaluation of systems and components developed in WP6 and ROXANNE as a whole in analyzing criminal networks. In the following subsections, we provide more information for these works.

6.1 System building

Data Fusion Bus: ITML is working on the adaptation of the Data Fusion Bus (DFB) platform to be used as the main element to the fusion procedures and, at the same time, to support and be included in the high-level architecture of ROXANNE platform. In parallel to the data (speech, text, and video) analysis carried out in WP5, DFB is adapted in light of the sequential development of data streams alignment and data points resolution. Regarding aligning data streams, ITML is examining the following methods/approaches on: (a) events (e.g. landmark reference, Name reference) - time merge of different Kafka streams, transform to Spark data frames and (b) the creation of aggregations in Spark data frames when required (min, max, average, distinct values, etc.). In sequence-based on event-time merge, datasets must be aligned and merged. In that direction, ITML is examining possible approaches such as: (a) deep learning approach (from free text, feature extraction, clustering, and classification), (b) to run clustering algorithms on text-based datasets, based on term frequency-inverse document frequency (automatic categorization of documents) and (c) to produce dimensionality reduction in large data frames.

Network analysis components: LUH is working on investigating and integrating more analysis functionalities to the **Analyzer**. In particular, we focus on leveraging and adapting state-of-the-art methods, as well as on developing new methods as the project progresses. In parallel, we are adding more **APIs** to the **API server**, as well as more features and interactions to the **Web interfaces** and **Visualizer**. Examples of such interactions are allowing end-users to add manual correction and labeling on the network. These will enable our systems to leverage the supervision from end-users for improving the analysis performance.

Integration into ROXANNE pipelines: AEGIS is working on integrating the Network analysis API service to the Forensics Visualization Toolkit[138] (FVT) that will act as the user interface to the ROXANNE platform. To this end, support for network diagrams was activated, including retrieval of network-related data, colour-coding, and the basic network management features such as base node selection, change of depth level, node repositioning, zoom-in/out and horizontal/vertical scrolling among others. Figure 16 shows the FVT interface showing an example network with community detection applied to the Enron database.

[138] https://aegisresearch.eu/solutions/advanced-visualization-toolkit/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

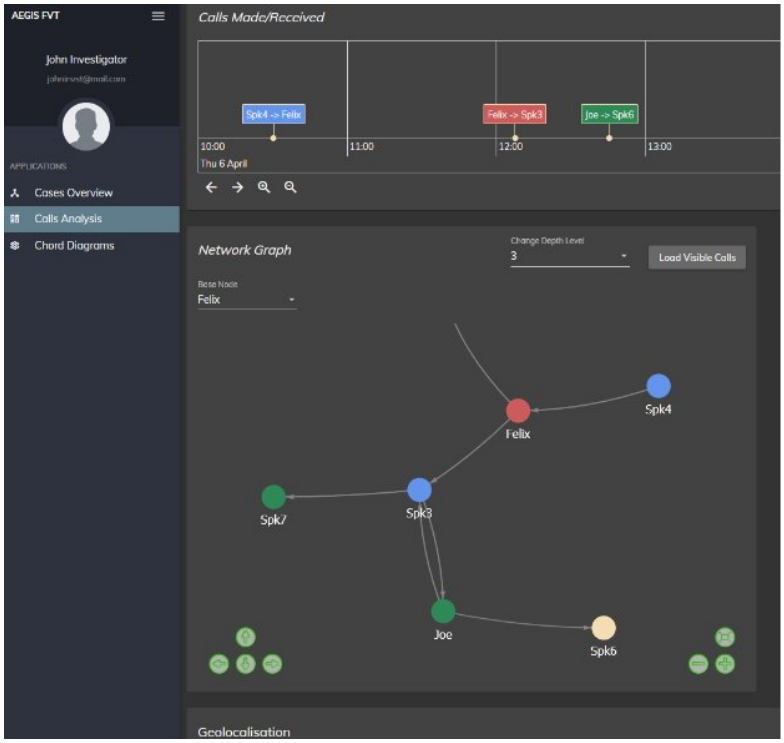


Figure 16. Community detection on Enron network

6.2 Embedding of Uncertain Networks

Due to their advantages in improving the performance of other network analysis functionalities and the relaxation of the domain expertise requirements, we put more effort into investigating and adapting network-embedding methods and developing new methods that are more suitable for the use in ROXANNE's pipelines. In particular, given the diversity of ROXANNE's data (e.g., data collected from multiple sources, multimodality, and the multilingual challenges, etc.) that may lead to high cost for any domain expertise requirement, and it is more reasonable to explore unsupervised and semi-supervised approaches for analyzing the data.

Furthermore, the input networks for our network analyzer are constructed from the output of speech and text analyzers developed in WP5, which often comes with uncertainty. For example, most speech recognition methods return the recognized utterance with some confidence scores that measure the uncertainty in the recognition process. This uncertainty in speech and text analysis results in uncertainty in the constructed networks. However, to the best of our knowledge, the network's uncertainty has not yet been appropriately considered and handled by existing network-embedding methods. There are only very few previous works on addressing part of this issue, e.g., the URGE model.

Taking these considerations into account, we have determined to investigate in developing novel methods for the embedding methods for uncertain networks. Potentially these methods can help us further to improve the performance of other network analysis functionalities and also to be able to provide more useful feedback to components and technologies developed in WP5. Our plan for this work would include the following phases

- Examining existing state-of-the-art network embedding methods when applied for uncertain networks
- Adapting and extending these methods for working with uncertain networks based on the findings of the previous steps.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

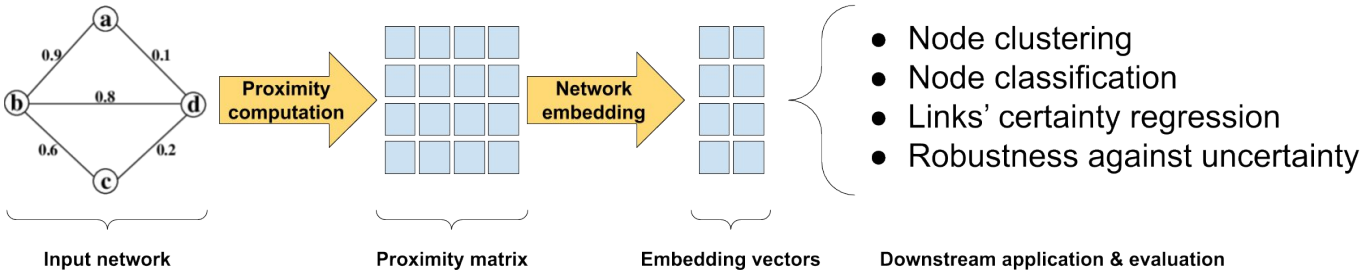


Figure 17. Framework for evaluating embedding methods on uncertain networks

Evaluation framework: In the first phase, we examine the methods according to the framework shown in Figure 17. That is, we first compute the proximity among nodes in the input network, then feed the obtained proximity matrix into different network embedding methods, and evaluate the quality of their learned embedding vectors learned through downstream applications: i.e., node clustering or community detection, node classification, links' certainty regression. We also examine the robustness of these methods when dealing with uncertainty in the input networks.

Proximity computation: This is the first step to deal with the uncertainty in the input network. This step's objective is to convert the input network into a deterministic one that is suitable for subsequent embedding methods. We will consider different methods for computing the proximity among nodes, including:

- Origin: no other computation is added, the probability associated with each link of the input network is considered as the edge's weight
- Expected Jaccard similarity: the proximity of a pair of nodes is defined by the expected value of the pairs across all the possible realizations of the input network

Methods to be examined: We will examine a wide range of network embedding methods, including:

- Matrix factorization methods: e.g., Nonnegative matrix factorization[139], and Singular value decomposition[140]
- Conventional node embedding methods:, e.g., DeepWalk[141], LINE[142], and Node2Vec[143]
- Graph Auto-encoder methods: e.g., DNGR[144], SDNE[145], and VGAE[146]
- Methods tailored for uncertain networks: e.g., URGE[147]

[139] Liu, Xin, et al. "A general view for network embedding as matrix factorization." *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019.
[140] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.
[141] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.
[142] Tang, Jian, et al. "Line: Large-scale information network embedding." Proceedings of the 24th international conference on the world wide web. 2015.
[143] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.
[144] Cao, Shaosheng, Wei Lu, and Qiongkai Xu. "Deep neural networks for learning graph representations." *Thirtieth AAAI conference on artificial intelligence*. 2016.
[145] Wang, Daixin, Peng Cui, and Wenwu Zhu. "Structural deep network embedding." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.
[146] Kipf, Thomas N., and Max Welling. "Variational graph auto-encoders." arXiv preprint arXiv:1611.07308 (2016).
[147] Hu, Jiafeng, et al. "On embedding uncertain graphs." *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Dataset: In addition to ROXANNE's datasets, we will use the following public datasets for examining the methods

- PPI datasets[148]: those are protein interaction datasets with ground truth labels are complexes (i.e., clusters) of proteins
- Synthetic datasets: those are simulated datasets that are generated by injecting uncertainty[149] into networks with ground truth labels for nodes' community or categories.
- **Preliminary results:** Figures 18, 19, 20, and 21 show the performance of the matrix factorization methods on PPI datasets. Here the performance is measured by the F1 scores obtained by the methods' learned embedding vectors in recovering ground truth complexes, i.e., community detection. In the figures, the legends are in the form of *proximity computation method, matrix factorization method.* For example, *matrix_S_EJS, lsnfm* means the running of least square nonnegative matrix factorization method on the proximity matrix obtained by expected Jaccard similarity.

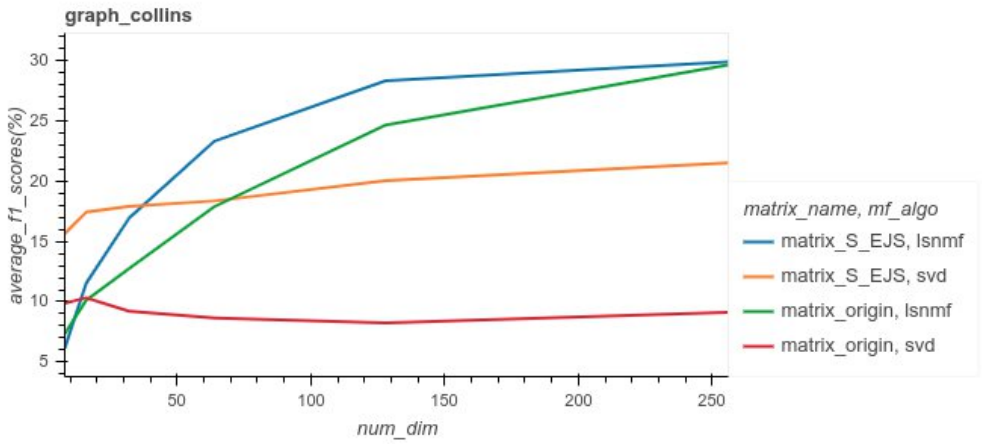


Figure 18. Performance of factorization methods on Collins dataset

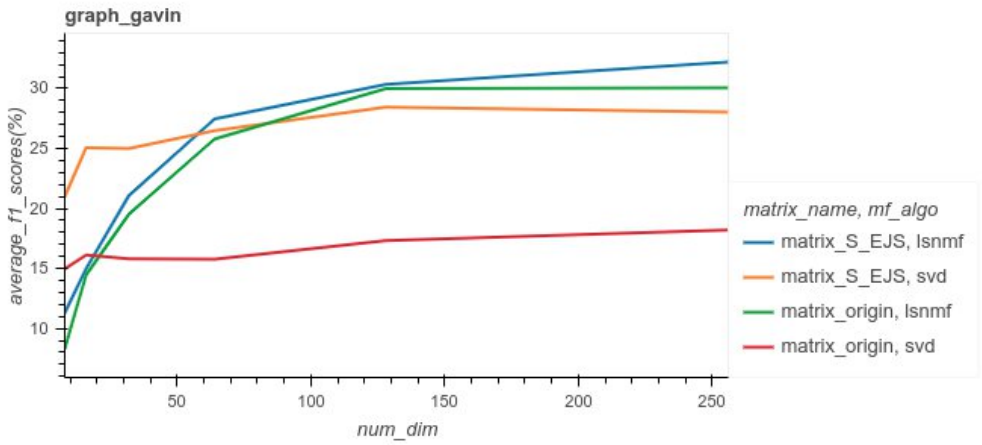


Figure 19. Performance of factorization methods on Gavin dataset

[148] Yang, Zhi Hao, et al. "Integrating PPI datasets with the PPI data from biomedical literature for protein complex detection." *BMC medical genomics* 7.S2 (2014): S3.

[149] Gionis, Paolo Boldi Francesco Bonchi Aristides, and Tamir Tassa. "Injecting Uncertainty in Graphs for Identity Obfuscation." *Proceedings of the VLDB Endowment* 5.11 (2012).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

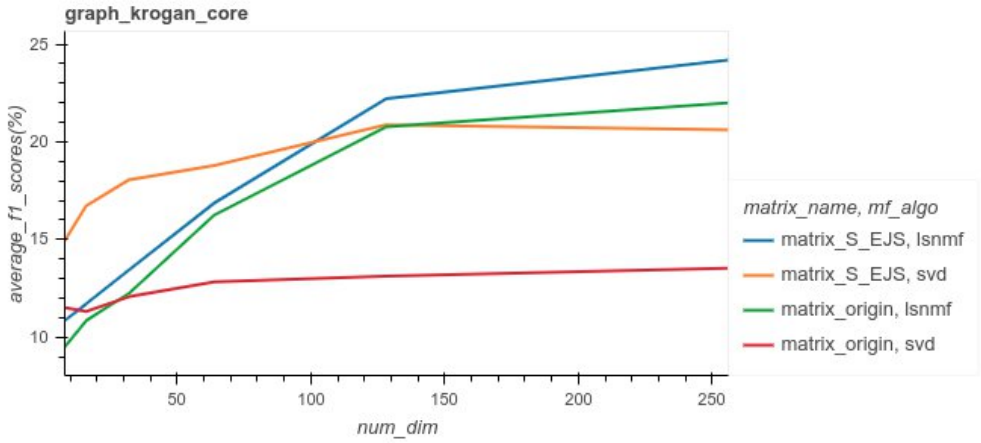


Figure 20. Performance of factorization methods on Krogan_core dataset

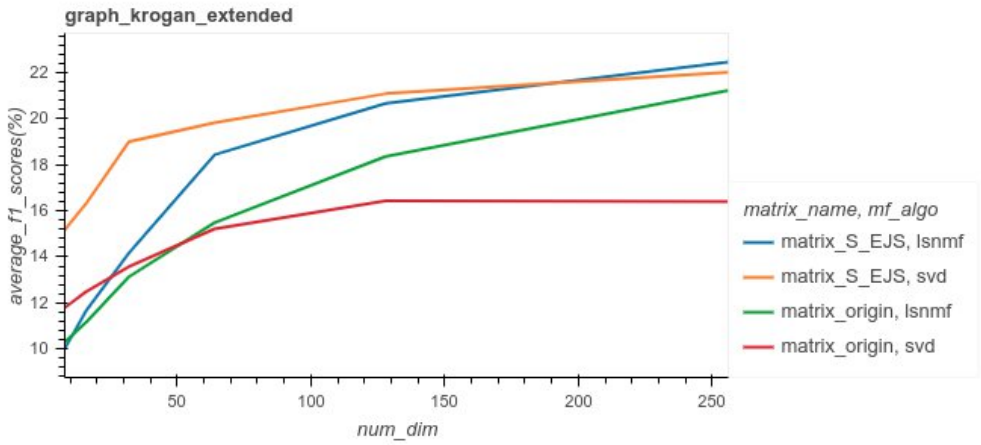


Figure 21. Performance of factorization methods on Krogan_extended dataset

The above figures clearly show that the performance of the methods is generally increased when we increase the dimension of the embedding vectors. It is also expected that this increment is saturated when after a certain value of the number of dimensions.

Our next steps will be evaluating all aforementioned methods in all the tasks listed in Figure 17. Findings from these evaluations would help us to devise more appropriate methods for uncertain networks.

6.3 Systematic evaluation

UCSC Transcrime's task in WP6 is dedicated to systematically assessing the added value of the tools and methods developed by ROXANNE (T6.7 and D6.4, due in M 33). This will be performed by comparing criminal networks extracted/derived through traditional approaches (contact logs, telephone contacts, meetings) with the networks enriched by additional information from the project's results. Given the peculiarities of each criminal network, at present, the assessment should be conducted on a relatively large sample of real criminal networks.

At the present stage of the project, the consortium is focused on the development of the platform and on identifying possible analytical methods. Until real criminal cases will be processed through the ROXANNE platform, UCSC Transcrime will be unable to perform the systematic assessment. In the meanwhile, UCSC Transcrime will continue to contribute to the partners' activities to ensure that the approaches developed take into account:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

- The realistic conditions of criminal investigations from a comparative perspective. While several LEAs are part of the consortium, ROXANNE aims to develop approaches and tools which may benefit a wide audience of agencies and stakeholders from multiple jurisdictions

- The type of operational problems investigators face as well as the delicate issues relating evidence and assumptions in criminal justice. Applications of network analysis to criminal justice need to consider that algorithms and tools often have a margin of error. In the context of criminal investigations, the protection of individuals' rights and freedoms is essential. Errors induced by imprecise algorithms should be avoided - if possible - or investigators should be clearly informed about assumptions and choices based on algorithms. Ideally, most of the findings should be reviewed by criminal investigators, who may decide whether to follow the indications of ROXANNE's tool or dismiss it.

- The evidence from criminological literature on networks, crime and particularly organized crime.

Furthermore, UCSC Transcrime will contribute to the general development of the analyses of WP6 in cooperation with the other partners, offering specific insight based on its expertise in the area of crime in general and of the analysis of criminal networks in particular. UCSC will contribute to the analysis of the NIST, CSI, and Enron datasets in collaboration with other partners. The contribution will ensure that the tools developed in ROXANNE are addressing relevant issues in the analysis of criminal networks as well as addressing the needs of law enforcement agencies from a variety of jurisdictions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Appendix A: Documentation for the API Server

Setup

Install Docker on Linux: Please follow the instruction link: DigitalOcean tutorial
Run Docker image from git repository:

Step 1: to login to git repository, run command:
docker login git.l3s.uni-hannover.de:8443

Step 2: to pull the latest version, run command:
docker pull git.l3s.uni-hannover.de:8443/ROXANNE/wp6_network_and_relation_analysis/docker/ROXANNE-l3s-api:latest

Step 3: to run the Docker image, run command:
docker run -p 5000:80 -p 5050:8080 git.l3s.uni-hannover.de:8443/ROXANNE/wp6_network_and_relation_analysis/docker/ROXANNE-l3s-api:latest

Calling the APIs

In the following we describe the parameters required by the main APIs provided by this component. For a clear representation, we demonstrate the API calls through examples in Python

First we need to import necessary packages and declare a request:

```
# import necessary packages
from pathlib import Path
from yarl import URL
import requests
from requests_toolbelt import MultipartEncoder

# Declare a request
base_url = URL("http://127.0.0.1:5000/") # by default the API server work on port 5000
request_headers = {
    "Content-Type": "application/json",
    "Accept": "application/json"
}
```

Authenticating API

```
# get the authenticating tokens
login_result = requests.post(
    base_url / "login",
    headers=request_headers,
    json={"username": "admin", "password": "pass1234"}
)
token_response = login_result.json()
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
# memorize the token for the subsequent calls
request_headers["Authorization"] = "Bearer " + token_response["access_token"] # Add token to headers
```

Moderating API

```
# getting all users
get_users_result = requests.get(
  base_url / "admin/users",
  headers=request_headers
)

# updating a user
update_user_result = requests.put(
  base_url / "admin/update/carl",
  headers=request_headers,
  json={
    "password": "SecreterP1234",
    "role": "Access"
  }
)

# deleting user
delete_user_result = requests.delete(
  base_url / "admin/delete/carl",
  headers=request_headers
)
```

Data querying API

```
# Create new dataset
create_result = requests.post(
   base_url / "query/dataset",
   headers=request_headers,
   json={
      "network": "new_network",
      "description": "Created right now"
   }
)


# Delete dataset
create_result = requests.delete(
   base_url / "query/dataset",
   headers=request_headers,
   json={
      "network": "new_network"
   }
)
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
# getting sub-network surrounding a set of nodes
pull_result = requests.post(
    base_url / "query/pull",
    headers=request_headers,
    json={
        "node_ids": ["Yayha_al_Azari", "Abu_Ubayda_al_Masri", "Abu_Mohammad_al_Anzari",
"Omar_al_Shishani"],
        "network": "bbc_islam_groups",
        "params": {}
    }
)

# adding nodes and edges
push_result = requests.post(
    base_url / "query/push",
    headers=request_headers,
    json={
        "nodes": {
            "data": [
                {
                    "id": "Billy",
                    "properties": {
                        "type": "murderer",
                        "name": "Billy",
                        "surname": "Herrington"
                    }
                }
            ],
            "params": {}
        },
        "edges": {
            "data": [
                {
                    "source": "Billy",
                    "target": "Omar_al_Shishani",
                    "properties": {
                        "type": "murdered",
                        "weapon": "knife"
                    }
                }
            ],
            "params": {}
        },
        "network": "bbc_islam_groups"
    }
)
```

Analysis API

```
# get available analysis methods
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

```
info_result = requests.get(
    base_url / "analysis/info",
    headers=request_headers
)
info_result.json()


# analyze network in database
analysis_result = requests.get(
    (base_url / "analysis/analyze").with_query({
        "network": "bbc_islam_groups",
        "task": "social_influence_analysis",
        "options": '{"method": "authority", "parameters": {}}'}),
    headers=request_headers
)


# analyzing a network contained in a local file
filepath = Path("../datasets/preprocessed/moreno_crime.json")
form_data = MultipartEncoder(fields={
  "payload": (filepath.name, open(filepath, "rb"), "application/json")
})
upload_result = requests.post(
    (base_url / "analysis/analyze").with_query({
        "task": "link_prediction",
        "options": '{"method": "preferential_attachment", "parameters": {"community_detection_method":
"modularity"}}'}),
    headers={
        "Content-Type": form_data.content_type,
        "Accept": "application/json",
        "Authorization": "Bearer " + token_response["access_token"]
    },
    data=form_data
)
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

Appendix B: Network visualizer

Setup

Install Docker on Linux: Please follow the instruction link: DigitalOcean tutorial
Run Docker image from git repository:

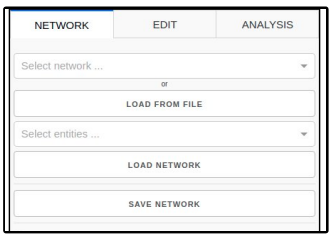
Step 1: to login to git repository, run command:
docker login git.l3s.uni-hannover.de:8443

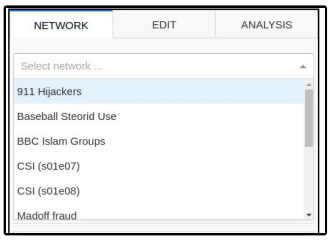
Step 2: to pull the latest version, run command:
docker pull git.l3s.uni-hannover.de:8443/ROXANNE/wp6_network_and_relation_analysis/docker/ROXANNE-l3s-visualization:latest

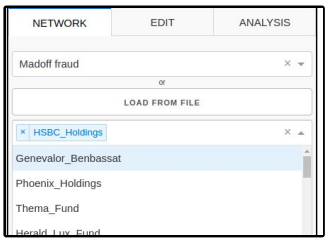
Step 3: to run the Docker image, run command:
docker run -p 8050:8050 git.l3s.uni-hannover.de:8443/ROXANNE/wp6_network_and_relation_analysis/docker/ROXANNE-l3s-visualization:latest

Network Loading

- Select network to load from '**Select Network**' dropdown or '**LOAD FROM FILE**' in **NETWORK** tab
- Choose entities (= nodes) to load from '**Select entities**' dropdown.
- Multiple selection is possible.
- Type to search for entities.
- If no entities are chosen, the whole network will be loaded.
- Click the '**Load Network**' button to load the network.







Network Manipulation

Selecting Nodes:

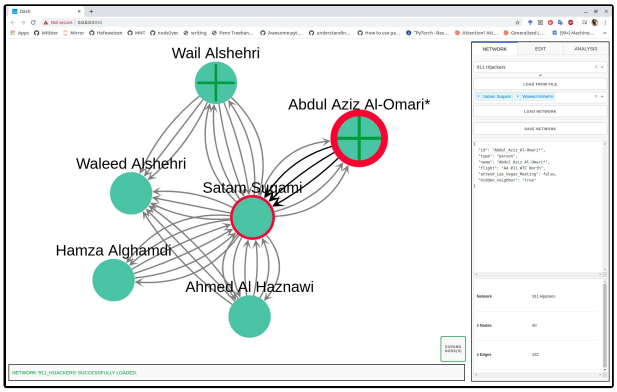
- Click on nodes to select them.
- Selected nodes, their outgoing edges and neighbors are highlighted.
- Click on a selected node to deselect them.

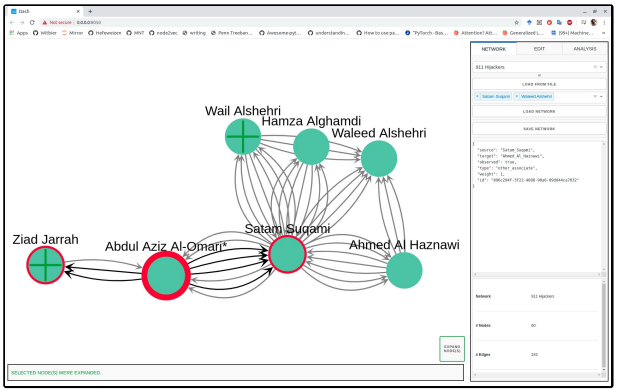
Expand Nodes:

- Nodes with a green plus sign in their body are expandable.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.

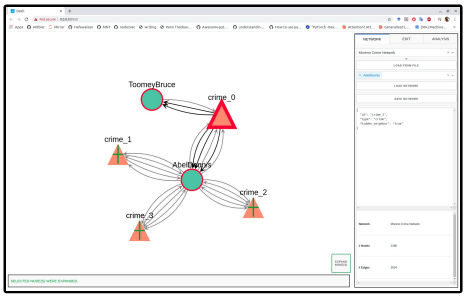
- Click the '**Expand Node(s)**' button to expand selected nodes (if expandable).

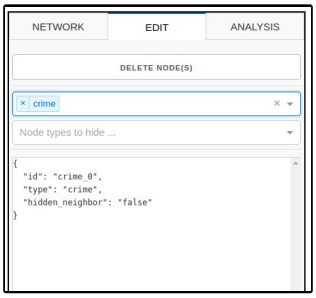


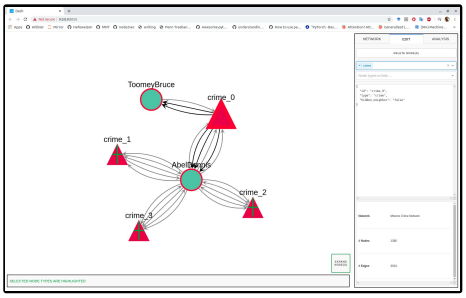


Highlight/Hide Node Types:

- Different types of nodes are indicated by different node shapes.
- Select a type of node from the '**Node types to highlight**' or '**Node types to hide**' dropdowns in Edit tab to hide/highlight specific types of nodes.



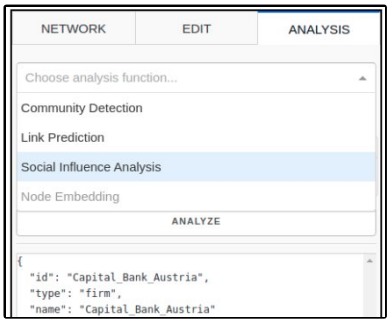


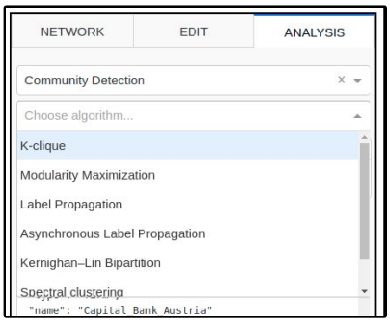


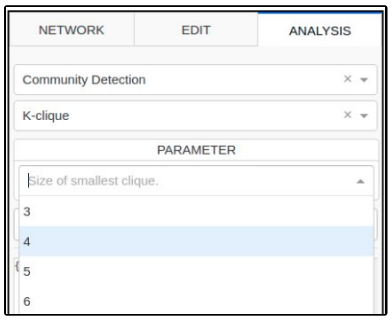
Analysis Functionalities

- Have a network loaded. Go to the **ANALYSIS** tab.
- Choose an analysis function from the '**Choose analysis function**' dropdown.
- Choose an algorithm from the **'Choose algorithm'** dropdown.
- If the algorithm supports parameter: Select parameter from the dropdown in the **'Parameter'** section
- Click the **'Analyze'** button to apply the chosen algorithm to the loaded network..

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833635.