Aalto University
School of Science

Master's Programme in
Security and Mobile Computing (NordSecMob)

Daniele Romanini

Privacy and Anonymization of Neighborhoods in Multiplex Networks

Master's Thesis
Espoo, November 29, 2018

Supervisors: Prof. Mikko Kivelä
Prof. Sune Lehmann Jørgensen
Advisor: Prof. Mikko Kivelä

Aalto University
School of Science

Master's Programme in
Security and Mobile Computing (NordSecMob)

ABSTRACT OF
MASTER'S THESIS

Author: Daniele Romanini

Title: Privacy and Anonymization of Neighborhoods in Multiplex Networks

Date: November 29, 2018 **Pages:** 133 + 9

Major: Security and Mobile Computing **Code:** T3011

Supervisors: Prof. Mikko Kivelä
Prof. Sune Lehmann Jørgensen

Advisor: Prof. Mikko Kivelä

Since the beginning of the digital age, the amount of available data on human behaviour has dramatically increased, along with the risk for the privacy of the represented subjects. Since the analysis of those data can bring advances to science, it is important to share them while preserving the subjects' anonymity. A significant portion of the available information can be modelled as networks, introducing an additional privacy risk related to the structure of the data themselves. For instance, in a social network, people can be uniquely identifiable because of the structure of their neighborhood, formed by the amount of their friends and the connections between them. The neighborhood's structure is the target of an identity disclosure attack on released social network data, called neighborhood attack. To mitigate this threat, algorithms to anonymize networks have been proposed. However, this problem has not been deeply studied on multiplex networks, which combine different social network data into a single representation. The multiplex network representation makes the neighborhood attack setting more complicated, and adds information that an attacker can use to re-identify subjects.

This thesis aims to understand how multiplex networks behave in terms of anonymization difficulty and neighborhood attack. We present two definitions of multiplex neighborhoods, and discuss how the fraction of nodes with unique neighborhoods can be affected.

Through analysis of network models, we study the variation of the uniqueness of neighborhoods in networks with different structure and characteristics. We show that the uniqueness of neighborhoods has a linear trend depending on the network size and average degree. If the network has a more random structure, the uniqueness decreases significantly when the network size increases. On the other hand, if the local structure is more pronounced, the uniqueness is not strongly influenced by the number of nodes. We also conduct a motif analysis to study the recurring patterns that can make social networks' neighborhoods less unique.

Lastly, we propose an algorithm to anonymize a pair of multiplex neighborhoods. This algorithm is the core building block that can be used in a method to prevent neighborhood attacks on multiplex networks.

Keywords: social network analysis, complex networks, multiplex networks, neighborhoods, anonymization, privacy

Language: English

Preface

This Master's Thesis was submitted in fulfillment of the requirements for acquiring a Master's degree, according to the NordSecMob double degree programme, at Aalto University and Technical University of Denmark. This thesis was prepared at the Department of Computer Science at Aalto University, and has been supervised by Prof. Mikko Kivelä and Prof. Sune Lehmann Jørgensen.

I would like to express my deep gratitude to Prof. Mikko Kivelä for giving me the opportunity to work with him in the Complex Systems group at Aalto University. Thanks to his knowledge, guidance, and availability I have learned a lot during this period. I am also very thankful to Prof. Sune Lehmann Jørgensen for always being supportive, helping and guiding me with his valuable advice. Moreover, I would like to thank the members of the Complex Systems group for creating a friendly working atmosphere, and for helping me whenever needed.
I wish to thank all the NordSecMob staff for their availability and for organizing this program. This experience allowed me to spend time both in Finland and Denmark, and to significantly broaden my mind. In these two years, I have met many people who are now good friends. I would like to thank all of them for being part of this journey. Thanks also to all my friends in Italy and around the world, for being part of my life and always being supportive.
I wish to thank also the people that gave me a hand or advice during the thesis process, in particular Enrica, for her useful advice on English writing, and Kristian, for his help in writing the Danish version of the abstract. Additionally, I wish to thank Eda, for always being next to me even in the most stressful days.
Finally, I am extremely grateful to my family, for having always supported me in any moment of my life and being there since the very beginning.

Espoo, Finland, November 29, 2018

Daniele Romanini

Symbols and Abbreviations

Abbreviations

ER	Erdős-Rényi (model/network/graph)
WS	Watts-Strogatz (model/network/graph)
RGG	Random Geometric Graph

Symbols

G	Graph (network)
M	Multiplex network
$\langle k \rangle$	Average degree
n	Number of nodes
m	Number of edges
V	Set of vertices
E	Set of edges
C	Clustering coefficient
p_k	Probability of a node to have degree k
L	Layer
ov_E	Edge Overlap proportion
SP	Significance Profile
μ	Mean
σ	Standard deviation
M_{ind}	Multiplex network containing the non-overlapping edges of a multiplex network M
M_{ov}	Monoplex network containing the overlapping edges of a multiplex network M
M_{agg}	Monoplex network resulting from the aggregation of a multiplex network M
I	Isomorphism classes set
$\|I\|$	Cardinality of I (Number of isomorphism classes in set I)

$\mathcal{I}$	Isomorphism type
$[0]$	Node isomorphism
$[0,1]$	Node-layer isomorphism
O	Occurrence Frequency
$\mathcal{N}_\tau$	Neighborhood of type τ
$\mathcal{N}_\subset$	Non-Inclusive Multiplex Neighborhood
$\mathcal{N}_\subseteq$	Inclusive Multiplex Neighborhood
$\mathcal{N}_a$	Aggregated neighborhood
$U_{\mathcal{N}_\tau}$	Uniqueness of neighborhoods of type τ
$U_{\mathcal{N}}^{(a)}$	Aggregated (or monoplex) uniqueness
$U_{\subset[\mathcal{I}]}$	Multiplex inclusive uniqueness according to isomorphism of type $\mathcal{I}$
$U_{\subseteq[\mathcal{I}]}$	Multiplex non-inclusive uniqueness according to isomorphism of type $\mathcal{I}$
$U_k^{(a)}$	Aggregated (or monoplex) degree uniqueness
$U_{k\,[0]}^{(M)}$	Multiplex node degree uniqueness
$U_{k\,[0,1]}^{(M)}$	Multiplex node-layer degree uniqueness
F	Frequency (of a neighborhood's subgraph in a network)
P	Proportion (of a neighborhood's subgraph in a network)

Contents

Chapter 1

Introduction

Over the last few decades, digitalization has caused an increase in the amount of generated data. Their analysis could lead to a significant improvement in science and a better understanding of human behaviour. However, most of this data cannot be shared because of obvious privacy risks. Data are made of attributes regarding, for example, people, and they could contain sensitive information that cannot be made publicly available, as their ensemble could lead to the re-identification of the subjects, also leaking private attributes. Since we live in an interconnected world, most of those data can be modelled as networks, or graphs, where nodes represent entities, and edges represent the relations between them.

In networked data, an additional privacy risk is represented by the structure of the data themselves. For instance, a certain node could be unique in a dataset given the amount of its connections, or the structure of its neighborhood, such as how its friends are connected. Indeed, if a person has four friends in a social network, and all the other people have a number of friends different than four, then that person is easily re-identifiable. The knowledge of the neighborhood's structure is the target of a known identity disclosure attack, called *neighborhood attack* [ZP08], on released social networks data. A considerable amount of anonymization techniques have been developed against this and other types of attacks [ZP11; TP10; Liu+15; ZCÖ09]. Those methods aim to modify the data before their release, to prevent privacy leaks and, at the same time, keep the utility of the data, to be still useful for analysis and research. Nonetheless, some data can be hard to anonymize, because of the high amount of unique values in them. Uniqueness is indeed an important feature to study for anonymization purposes, since the reason why re-identification of entities occurs is that they are somehow unique. The more information the dataset carries, likely, the higher the uniqueness is and, with it, the identity disclosure risk.

Despite the utility of the diversity of available data, a further danger is caused by the variety of data sources, which can be linked together to uncover the identity of the target individuals. For example, the interactions between individuals in social systems happen in different social contexts, and, as each of those contexts can be seen as a network, their combination can be represented as a multiplex network [Kiv+14]. A multiplex network is composed of various interconnected layers, where each layer is a network itself representing either a social context or a temporal slice. For example, in the former case, one layer can represent Facebook friendships and another one Twitter connections or phone calls, while in the latter, each layer can be the view of the system during a particular time period. Multiplex networks carry with them more information than networks represented as a simple graph, and, since more and more systems can be represented with this tool, anonymization techniques should be extended to this new type of systems. Moreover, given the complex anatomy of those networks, the uniqueness of structures such as neighborhoods should be studied to understand what it depends upon, and, consequently, the difficulty of the anonymization problem applied to such data.

1.1 Motivation and objective

The motivation of this thesis derived from the lack of a clear understanding on how the network structure influences the formation of unique neighborhoods, leading to an easier re-identification of the entities in social networks in a neighborhood attack. Networks are of different shapes and sizes, and the diversity of those features can be linked to the probability of having neighborhoods that are either similar or diverse to each other, and, consequently, unique. Moreover, the use of multiplex networks to represent social graphs can lead to a further threat in some cases, since the attacker could be equipped with additional information that can be crucial to the success of the neighborhood attack. Since multiplex networks can be shared in different ways, for example with or without the layer's label or by aggregating them into a single layer, it is also crucial to determine the best strategy for sharing data to minimize the identity disclosure risk.

To understand and quantify the privacy risk associated to the uniqueness of neighborhoods in both simple and multiplex networks, and consequently the best strategy for sharing multiplex data, we systematically study how the uniqueness value changes by generating network with different network models. Network models are tools to generate networks with a particular

structure, given parameters such as the number of nodes and edges. Changing those parameter and measuring the amount of unique neighborhoods in different settings can give us an idea of how the organization of the nodes in a network influences the uniqueness. While doing this experiment with multiplex networks, another essential feature to take into account is the amount of edges that are shared between the layers, which can represent, for example, distinct social contexts. If two contexts do not have any similarity, then the fraction of unique neighborhoods could be totally different compared to the case in which there are clear similarities in the way nodes are linked.

The aim of neighborhoods anonymization algorithms is to modify group of similar neighborhoods as little as possible, by adding or removing edges or nodes, to make them equal to each other. If two neighborhoods are already similar, then few additions or deletions of edges or nodes are needed. On the other hand, if neighborhoods would be entirely different to each other, reaching the anonymization would imply major data modifications, lowering their actual utility and reliability for studies. We want to study the recurrent patterns that form neighborhoods in social networks, making them to be similar, or equal, to each other. The presence of those patterns lowers of the fraction of unique neighborhoods in social networks, allowing anonymization algorithms to work. This analysis can also lead us to figure out the differences in the neighborhoods between empirical data and network models, which are randomly built, without following rules that can be at the base of real-world social networks.

Since more and more data can be represented as a multiplex network, it is important not only to study the risk given by the uniqueness of neighborhoods in such data, but also to move towards the development of a neighborhoods anonymization algorithm for multiplex networks. For this reason, we also want to discuss problems that need to be taken into account while developing an anonymization algorithm and propose a method to make a pair of multiplex neighborhoods equal to each other. This is the central ingredient for a full network anonymization algorithm, and can be fit in one of the existing anonymization' frameworks working on simple networks.

1.2 Contribution and thesis structure

Aiming to understand the uniqueness and structure of neighborhoods in simple and multiplex networks, we first present, in **Chapter 2**, the necessary background information, describing the tools we use in the whole document. We start the chapter by defining networks and multiplex networks, along with their basic features. We then define the problem of determining whether

two networks, or network's neighborhoods, present the same structure, called graph isomorphism. After that, we illustrate the network models with which we conduct the simulations to determine the uniqueness of neighborhoods in networks with specific structures, and the tool to reveal the structure of neighborhoods. Finally, we present the concept of privacy in social networks and the different definitions that exist nowadays, and the techniques with which data can be anonymized, along with their limitations.

To evaluate the development of an anonymization algorithm on multiplex network to protect against neighborhood attack, in **Chapter 3**, we survey the existing methods working on simple networks. We also summarize the state-of-the-art in the anonymization of multiplex or similar types of networks to protect against other types of attack. This will be helpful when considering the right approach of working with more complicated network structure such as multiplex networks.

The multiplex network representation added a degree of freedom to the networks' framework, and most of the concepts related to those can be ambiguous. Therefore, to avoid confusion, it is important to define basic principles clearly when working with these systems. Neighborhoods in multiplex network can also be defined in multiple ways. The chosen definition depends on the attacker's knowledge in the neighborhood attack scenario, and can influence the fraction of nodes with a unique neighborhood. We present two definitions of multiplex neighborhoods and the hypothesis we make for our study in **Chapter 4**, where we also define the uniqueness of neighborhoods in both simple and multiplex networks. We then study, in **Chapter 5**, the variation of the uniqueness of neighborhoods in some of the network models presented in Chapter 2. We adapt network models to a multiplex setting, to understand the difficulty of the neighborhoods anonymization problem in various settings, and consequently decide the best strategy for sharing multiplex network data, which can either be sharing the data with or without layers' label, or aggregating the data from different layers into a single-layer. We generate networks with different parameters and features, such as network size, average degree or, in the case of multiplex networks, the number of overlapping edges between different layers. We also show that the degree of the nodes, which is the amount of edges connected to it, is not enough to characterize neighborhoods, and the number of nodes with a unique degree is not necessarily the same with unique neighborhoods. For one of the models (the *Erdős-Rényi* model), we present the equations to determine the amount of nodes with a unique degree combination both in a simple and in a multiplex network.

In **Chapter 6** we compute the uniqueness of some real-world datasets with different sizes and features and then compare them with the network

models presented in Chapter 5, to understand if models can be used as a proxy for real world-networks and which is the best one.

We analyze the basic patterns recurring in neighborhoods of two of the presented datasets in **Chapter 7**, where we analyzed networks of calls and text messages both in a normal and a multiplex setting, revealing some of the structural differences between them and random networks.

At the end, in **Chapter 8**, we build upon the anonymization algorithms presented in Chapter 3, to discuss their possible adaptation to multiplex networks, pointing out the main features of multiplex networks that should be taken into account and the reasons why existing algorithms cannot be easily used for that kind of systems. As a first effort to reach this, we propose an algorithm to anonymize a pair of multiplex neighborhoods with two layers. The presented method could be used as the basic step to develop a complete anonymization method for multiplex networks.

We conclude by summarizing the main results of our study and discussing possible future work in **Chapter 9**, to improve both the understanding of uniqueness and structure of neighborhoods in multiplex networks, and the development of an anonymization method.

Chapter 2

Background and Preliminaries

In this chapter, we present the necessary background information used throughout the whole document. We start by defining networks, and the type of networks we mostly use in this thesis, multiplex networks, along with the general framework they belong to, multilayer networks. We then present the graph isomorphism problem, which is the problem of determining whether two networks are isomorphic or not. This is a central concept since neighborhoods can be seen as networks themselves, thus, if they have the same structures, they are said to be isomorphic. We also describe the network models we use for our experiments and, at the end, we give an overview of the concept of privacy in social network data.

2.1 Complex Networks

Networks, or graphs, are mathematical objects that consist of nodes (or vertices) and edges, that link pairs of nodes together [New18]. Networks are used for modelling different kind of systems, from biological ones (where nodes represent, for example, proteins, and edges the interaction among them), to public transportation networks (where the edges connect locations) and social networks (where nodes are people or organizations, and edges represent some relationship among those, such as phone calls, meetings, or social media friendships). There are different types of networks, with different characteristic. For example, a graph can have multiple edges among two nodes (in this case we talk about a multigraph), the edges can have a direction from one node to another (directed graph), the nodes or the edges can be labelled (node-labelled or edge-labelled graph) or weighted (weighted network). In this thesis, we focus on undirected, unlabelled and unweighted social networks, without multiple edges between two nodes, and without

self-loops (such as without edges from a node to itself). In this section, we introduce some basic definition of networks, which we use throughout the document. We do the same in the next Section 2.2 with *multiplex networks*, the tool we use to represent a combination of different social networks.

We define a *graph* (or *network*) as a set of vertices (or nodes) and a set of edges $G = (V, E)$. An edge is a tuple of two nodes (unordered, since we are referring to undirected graphs). All the edges that are linked to a node n are said to be *incident* to n. The *degree* k of a node is the number of edges that are incident to it. The *average degree* $\langle k \rangle$ of a network is the average number of edges that are incident to a node in that network. The average degree is obtained by the formula:

$$\langle k \rangle = \frac{2m}{N}\,, \tag{2.1}$$

where m is the total number of edges and N is the total number of nodes in the network.

To define *neighborhoods*, one of the central concepts of this thesis, we first need to define the concepts of *subgraph* and *induced subgraph*. A *subgraph* $G^* = (V^*, E^*)$ of a graph $G = (V, E)$ is a graph where the set of vertices V^* is a subset of V ($V^* \subseteq V$), and the edges set $E*$ is a subset of the edges E between the nodes in V^* ($E^* \subseteq \{(v_i, v_j) \in E | v_i, v_j \in V^*\}$).

An *induced subgraph* $G[V^*]$ is a subgraph of $G = (V, E)$ that contains all the edges between the vertices in V^* (where $V^* \subseteq V$). Thus, in an induced subgraph, the edge set E^* is $E^* = \{(v_i, v_j) \in E | v_i, v_j \in V^*\}$.

Two nodes connected by a link are called *neighbors*. A *neighborhood* $\mathcal{N}(v)$ of a node v in a graph G is then the subgraph induced by set of v's neighbors. Thus, indicating with V^v the set of neighbors of v, and with E^v the set of edges existing between the vertices in V^v, a neighborhood $\mathcal{N}(v)$ in a graph G is defined as:

$$\mathcal{N}(v) = G[V^v] = (V^v, E^v)\,. \tag{2.2}$$

An example of neighborhood of a node is shown in Figure 2.1.

A network can be represented by a square matrix $n \times n$ called *adjacency matrix* A, where rows and columns represent the nodes, and the elements a_{ij} take values 1 or 0, based on whether an edge between the corresponding pair of nodes exists in the network or not. More formally:

$$a_{ij} = \begin{cases} 1, & if\ (i,j) \in E \\ 0, & if\ (i,j) \notin E \end{cases}. \tag{2.3}$$

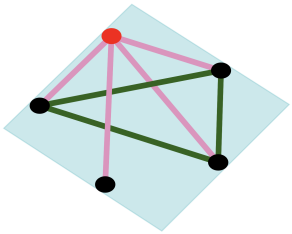

Figure 2.1: A node and its neighborhood. The red node represents the central node and the pink edges the edges connecting it to its neighbors. The neighborhood is composed by the black nodes and the green edges.

The adjacency matrix representation is useful to compare networks during the anonymization process. We present anonymization algorithms that use this representation in Chapter 3, and we also use it in the algorithm we propose in Chapter 8.

The nodes in a network can have different degrees. As we discuss in the Chapters 3 and 5, the degree is a feature that can make nodes re-identifiable, since there can exist only a few nodes with a given degree. The *degree distribution* $P(k)$ is the distribution of the degrees occurring in the network. With the degree distribution, we can estimate the probability of randomly picking a node with degree k, by computing:

$$P(k) = \frac{n_k}{n}\,, \tag{2.4}$$

where n_k is the number of nodes in the network with degree k, and n is the number of nodes. The degree distribution is a central concept in the study of networks. In most of the real-world networks, we can observe a heavy-tailed distribution, such as with tails that are not exponentially bounded. In particular, there has been wide attention to the study of scale-free networks, such as networks whose degree distribution follows (asymptotically) a power law:

$$P(k) \sim k^{-\gamma}\,, \tag{2.5}$$

where γ is the *degree esponent* [Bar+16]. A power law distribution should follow a straight line when plotted on a log-log scale. This kind of degree distribution is typical of networks that have many nodes with small degrees and there are few big hubs (a group of nodes densely connected, such as with high degree). Degree distributions close to power law are often observed in real networks, like the ones we analyze in Chapter 6.

Another important notion (relevant also in our analysis, since it directly characterizes a specific neighborhood) is the *local clustering coefficient* of

a node [WS98], which measures how much a node is "clustered" with its neighbors. The local clustering coefficient of a node n_i is defined as the fraction of the existing edges between the neighbors of n_i out of all the possible edges between its neighbors, and it is computed by:

$$C_i = \frac{m}{\binom{k_i}{2}} = \frac{2m}{k_i(k_i - 1)}\,. \tag{2.6}$$

By averaging the local clustering coefficient of all the nodes in the network, we otbain the *average local clustering coefficient*:

$$C = \langle C \rangle = \frac{1}{n} \sum_i C_i\,. \tag{2.7}$$

In reality, social networks present a low clustering coefficient, meaning that they are sparse (and, as mentioned before, there are just a few nodes that are densely connected, forming hubs, thus with a high value of clustering). We show some examples of real-world networks with different clustering coefficient values in Chapter 6, and we relate it with the difficulty of the anonymization problem.

2.2 Multilayer Networks

As mentioned in the previous section, the classic representation of a network is a set of nodes and edges. However, the systems around us can be very complex, and, with the amount of information growing, we need more powerful tools that can capture more realistic features. For example, systems are dynamical, and they have a complex behaviour that changes over time, or the entities interact in different contexts, or in different ways (for instance, people can communicate with both phone calls or sms). A tool to model these complex systems through the use of networks consists of *multilayer networks* [Kiv+14], which we introduce in this section, along with some of the related definitions and measures. We also present multiplex networks, a particular type of multilayer networks, on which we focus our study.

Multilayer networks are a combination of networks, each one represented in one layer, with possible additional edges that cross different layers. Those edges that link nodes belonging to different layers are *inter-layer edges*, while the edges between nodes in the same layer are *intra-layer edges*. There can be different types of layering, and each one of those is called *aspect*. Aspects can be seen as the dimension of a space. Every node of the network can be

present in one or multiple layers, where a layer is defined as a combination of *elementary layers*, each one corresponding to an aspect. Thus, if a network has n aspects, we would need n elementary layers to identify a specific layer. However, in this thesis, we focus on networks with a single aspect, since most of the real-data can be represented in this way.

A multilayer network can be formally defined with a quadruplet $M = (V_M, E_M, V, \mathbf{L})$, where V is the set of all the vertices in the network; $\mathbf{L} = \{L_a\}_{a=1}^{d}$ is the sequence of the elementary layers in each aspect, and, as mentioned before, we can exploit those to define all the layers in the network as $\hat{L} = L_1 \times \cdots \times L_d$; V_M represents the set of *vertex-layers tuples* (or node-layers), such as the set of vertices that are present in certain layers, and it is defined as $V_M \subseteq V \times L_1 \times \cdots \times L_d$; $E_M \subseteq V_M \times V_M$ is defined as the set of edges between two vertex-layer tuples.

The concept of multilayer networks is comprehensive and general, depending, for instance, on the admitted type of inter-layer edges. As an example, inter-layer edges could link nodes representing different entities in different layers or not. In this thesis, we focus on a particular type of multilayer networks, called *multiplex networks*, where the inter-layer edges link the same node across different layers. Multiplex networks are equivalent to edge-colored multigraphs. A multiplex network M can be defined as $M = \{G_\alpha\}_{\alpha=1}^{b} = \{(V_\alpha, E_\alpha)\}_{\alpha=1}^{b}$, such as a sequence of graphs, where the set of nodes is V_α and the set of edges is $E_\alpha \subseteq V_\alpha \times V_\alpha$. Since a multiplex network is defined as a sequence of graphs, we can call a single graph *monoplex network*.

If the set of nodes is the same in each layer, then the network is said to be *fully interconnected* (or *node-aligned*). However, in this document, we also treat networks where there can be missing nodes in some layers, thus that are not fully interconnected. In multiplex networks, layers can be coupled together in different ways: if they have a specific order (as in the case of the multilayer representation of a temporal network, where each layer represent a specific time-stamp), then the inter-layer edges connect nodes just from one layer to another, creating a network with *ordinal* coupling; conversely, if such an order does not exist (as in the case where each layer represent a different social context), then the network has *categorical* coupling, and the inter-layer edges connect nodes across all the layers.

A multilayer network can be aggregated over different aspects, reducing the amount of them. For example, if we aggregate a multiplex network with one aspect, we obtain a monoplex (or single-layer) network where the set of vertices and the set of edges are the union of, respectively, the set of vertices and edges present in the layers we are aggregating. We then miss the information regarding the original layers of the nodes and edges. We say

that the edges that link the same nodes in different layers are *overlapping*, and we define the *edge overlap* as the proportion (or the amount) of edges that are overlapping in the considered layers. Similarly, the *degree overlap* of a node in a given layer combination is the number of links, incident to a node, shared between those layers.

2.3 Graph isomorphism

To understand if a certain neighborhood is distinguishable from other neighborhoods in the graph or not, we need to define *graph isomorphism* [For96]. Graph isomorphism is the problem to determine whether two graphs (or networks) are equivalent to each other. In our context, two networks (or neighborhoods) are isomorphic if they have the same structure.

Two graphs G and G', with vertices set V and V', are said to be isomorphic ($G \cong G'$), or belonging to the same isomorphism class, if there exists a bijective function γ such that:

$$\gamma : V \to V' \,. \tag{2.8}$$

γ should relabel the vertices of G to the ones of G' (and vice-versa), such that the resulting relabeled graph $G^\gamma = (V^\gamma, E^\gamma)$ is equivalent to G' (or belongs to the same equivalence class):

$$V^\gamma = \{\gamma(v) \mid v \in V\} \,, \tag{2.9}$$

$$E^\gamma = \{(\gamma(v), \gamma(u)) \mid (v, u) \in E\} \,, \tag{2.10}$$

$$G^\gamma = (V^\gamma, E^\gamma) \,. \tag{2.11}$$

γ is an isomorphism between G and G'. Note that the isomorphism relation is edge preserving, as shown in Equation 2.10.
The graph isomorphism can also be defined for different types of graphs, for instance for node-labelled graphs, in two different ways, depending on whether we want to preserve the labels, or we admit the mapping of nodes to other nodes having the same labels (thus preserving the labels' equivalence classes).

An isomorphism can also exist from a graph G to itself. In this case, we talk about *graph automorphism*.

The graph isomorphism and automorphism are not known to be NP-complete or to belong to the class of problems solvable in polynomial time [GJ02].

Given the amount of nodes n, one can compute the number of possible graphs with those nodes. This problem is known as *graph enumeration* [FP73] and can be applied to undirected or directed graphs. The number of simple undirected graphs is given by

$$2^{\binom{n}{2}} = 2^{\frac{n(n-1)}{2}} . \tag{2.12}$$

Note that Equation 2.12 does not give the number of isomorphism classes for unlabelled graphs (however, it gives this number for colored graphs, where each node has a different color), since multiple graphs can belong to the same isomorphism class. The number of isomorphism classes for simple undirected graphs (or, in other words, the number of non-isomorphic simple undirected graphs) is not easy to compute. However, we can say that this number grows rapidly as n grows, and there are tools [McK83] that list all the non-isomorphic graphs with a given amount of vertices.

In the context of graph anonymization, we need to edit a graph g_1 to make it equivalent to another graph g_2, in such a way that, after the editing, the two graphs belong to the same isomorphism class. The edit operations e_i can be vertex or edge insertion, deletion, or substitution. We can limit the number of allowed operations based on the application. The set of edit operations necessary to transform a graph into another is the *edit path* $\mathcal{P}(g_1, g_2)$, and the measure of dissimilarity between the two graphs is the *graph edit distance* (GED) [SF83]. GED between two graphs g_1 and g_2 is defined as

$$GED(g_1, g_2) = \min_{(e_i,\dots,e_k)\in\mathcal{P}(g_1,g_2)} \sum_{i=1}^{k} c(e_i) , \tag{2.13}$$

where $c(e_i) \geq 0$ is the cost of the edit operation e_i.

Graph edit distance has applications also in fields as pattern recognition, in particular in fingerprint or face recognition. Optimal algorithms for solving graph edit distance transform the problem into the one of the shortest path finding. However, many non-optimal methods have also been developed [Gao+10].

Another problem related to graph edit distance is *graph matching* [BJ00] or network alignment, which is the problem of finding similarity between graphs. Indeed, one can first try to find an alignment between two graphs to compute the minimum edit path between those (this would also be useful when it comes to anonymization, discussed in 3). Some of the developed methods for computing the graph edit distance are based on matching sub-structures of the two graphs, and also neighborhoods. Graph alignment

is a widely studied problem on its own, and numerous algorithms have been developed specifically for it [Kuc+10; SXB08; Lia+09].

2.4 Isomorphism in Multilayer Networks

The problem of isomorphism in multilayer networks is presented in [KP18]. The main points of the paper that are interesting for this thesis are the existence of different types of isomorphism for multilayer networks, and the possibility of reducing the problem to the one of the isomorphism in vertex-colored graphs. Different types of isomorphism, in our context, corresponds to different neighborhood attack's scenario, depending whether only the nodes' labels are shared, or also the layers' label are known or not.

The different kind of isomorphism in multilayer networks depend on the nature of the mapping given by the bijective function that defines the isomorphism. Given two multilayer networks M and M', the types of isomorphism that we can have are:

- *vertex isomorphism*;
- *layer isomorphism*;
- *vertex-layer isomorphism*.

The vertex isomorphism is defined similarly to the one in simple graphs, with the difference that the vertices are identified with vertex-layer tuples. The bijective function γ (or a vertex map) should relabel the vertices of a multilayer graph M, such that $M^\gamma = M'$, keeping unaltered the layers' labels. In particular it is defined as following:

$$V_M^\gamma = \{(\gamma(v), \boldsymbol{\alpha}) \mid (v, \boldsymbol{\alpha}) \in V_M\}\,, \tag{2.14}$$

$$E_M^\gamma = \{((\gamma(v), \boldsymbol{\alpha}), (\gamma(u), \boldsymbol{\beta})) \mid (v, \boldsymbol{\alpha}), (u, \boldsymbol{\beta})) \in E_M\}\,, \tag{2.15}$$

$$M^\gamma = (V_M^\gamma, E_M^\gamma, V^\gamma, \boldsymbol{L})\,. \tag{2.16}$$

In the equations above, V_M is, as before, the set of vertex-layer tuples, and $\boldsymbol{\alpha}$ is the vector of layers in which a vertex v is present.

Indicating with a the elementary layers' indices, and with d the aspects' ones, we have *layer isomorphism* between M and M' if there exists a layer map $\boldsymbol{\delta} : \hat{L} \to \hat{L}'$ that relabels all the existing elementary layers $\boldsymbol{\alpha}$ of M such that $M^\delta = M'$, in the following way:

$$\boldsymbol{L^\delta} = \{L_a^{\delta_a}\}_a^d \text{ and } L_a^{\delta_a} = \{\delta_a(\alpha) \mid \alpha \in L_a\}\,, \tag{2.17}$$

$$V_M^{\boldsymbol{\delta}} = \{(v, \boldsymbol{\delta}(\boldsymbol{\alpha})) \mid (v, \boldsymbol{\alpha}) \in V_M\}\,, \tag{2.18}$$

$$E_M^{\boldsymbol{\delta}} = \{((\gamma(v), \boldsymbol{\delta}(\boldsymbol{\alpha})), (\gamma(u), \boldsymbol{\delta}(\boldsymbol{\beta})) \mid (v, \boldsymbol{\alpha}), (u, \boldsymbol{\beta})) \in E_M\}\,, \tag{2.19}$$

$$M^{\boldsymbol{\delta}} = (V_M^{\boldsymbol{\delta}}, E_M^{\boldsymbol{\delta}}, V, \boldsymbol{L}^{\boldsymbol{\delta}})\,. \tag{2.20}$$

When both a vertex map and a layer map exist, then we can say that there is a vertex-layer map $\zeta = (\gamma, \delta)$ that relabels both the vertices and the layers of M, such that $M^{\zeta} = M'$. In this case, there exists a *vertex-layer isomorphism* between M and M'.

Since the set of vertices V in a mutlilayer network can be seen as the "0th aspect", we can indicate, through a particular notation, that two multilayer networks are isomorphic with respect to certain aspects. This concept allows us to define *partial* isomorphism, in which the labels' permutation is allowed only in certain aspects (for example, if there are two aspects, one can define a vertex-layer isomorphism permuting the layer labels in just the first aspect). In particular, $M \cong_0 M'$ indicates the existence of a vertex-isomorphism, $M \cong_1 M'$ corresponds to a layer isomorphism and $M \cong_{0,1} M'$ to a vertex-layer isomorphism.

For practical computations matters, the authors of [KP18] reduce the multilayer networks' isomorphism problem to the colored graphs isomorphism problems. In this way, it is possible to make use of Bliss [JK07], the backend used by Pymnet [Kiv17] (a multilayer network software library) for isomorphism-related computations. Figure 2.2 shows an example of a reduction of a multilayer network with a single aspect to a vertex-colored graph. We can distinguish two cases for this reduction, based on the isomorphism type and the network type:

- vertex isomorphism: one can assign different colors to the nodes present in different layers (identified with vertex-layer tuples), and connect together the same nodes across different layers with additional "white" nodes;

- vertex-layer isomorphism: conversely to the vertex isomorphism, each vertex-layer tuple is assigned to the same color (since it is not), but still the same nodes across different layers are connected with additional "white" nodes. Moreover, all the nodes present in the same layers are linked to another "black" node, representing the layer itself.

The reduction is also possible for networks with multiple aspects (examples of those cases are presented in [KP18]). As mentioned before in Section 2.2, we focus however on networks with just one aspect.

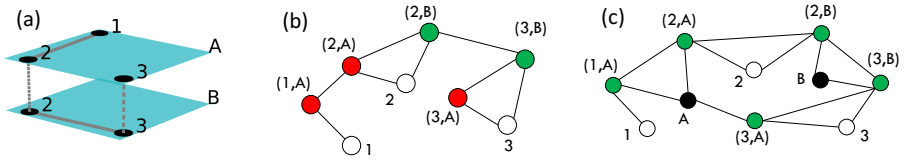

Figure 2.2: Reduction of a multilayer network (on the left) with a single aspect to a vertex-colored network, according to vertex-isomorphism (in the center) and vertex-layer isomorphism (on the right). Figure adapted from [KP18].

One approach to determine if two graphs G and G' are isomorphic is to compute a certain function f, called complete invariant, on each of them. If the output of the function is the same for both G and G', then they are isomorphic (or vice-versa). The complete invariant for vertex-colored graphs is then the same then the one for multilayer networks reduced to them.

The graph enumeration problem in multilayer networks can be defined in different ways, depending on the actual type of graph we want to represent with the framework of multilayer networks. The number of fully interconnected multiplex networks with n nodes and b layers is:

$$2^{b\binom{n}{2}}. \tag{2.21}$$

In the same paper, the number of isomorphism classes for vertex and vertex-layer isomorphism of multiplex networks is computed by listing all the possible graphs up to 5 nodes and 3 layers. The number of classes grows very quickly with both layers and nodes. For example, with 3 layers, the classes of 3 vertices are 36 for vertex-layer isomorphism and 120 for vertex-isomorphism; the classes of 4 vertices are 2381 for vertex-layer isomorphism and 12496 for vertex-isomorphism; the classes of 5 vertices are 1540146 for vertex-layer isomorphism, and 9156288 for vertex isomorphism. The number of classes given by vertex-layer isomorphism is always lower or equal than the ones given by vertex-isomorphism, since there exist more graphs that can be mapped to the same equivalence class according to vertex-layer isomorphism. An example is shown in Figure 2.3.

2.5 Random networks models

In this section, we introduce some models used to generate random networks. Each of this models generates random networks with different structural

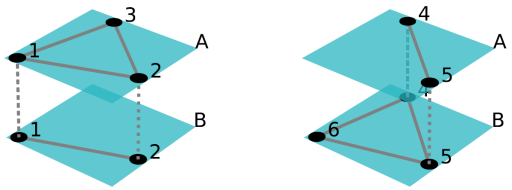

Figure 2.3: Two multiplex networks that are not vertex isomorphic, but are vertex-layer isomorphic

properties, and the study of those can be useful to understand complex phenomena such as disease spreading [New02]. We will instead exploit some of those models to study how the structure of complex networks influences the re-identification of a node and, consequently, changing the difficulty of the anonymization problem. In particular, we will study the uniqueness of neighborhoods in the Erdős - Rényi, Watts-Strogatz and Random Geometric Graph models. We will instead use the Configuration model as a null model to study recurrent neighborhood structures in real-world social networks.

2.5.1 Erdős - Rényi model

The *Erdős - Rényi* (ER) model [ER60] generates random networks with size n. Two variants of the model exist:

- $G(n, p)$: a graph is generated by connecting each pair of nodes with probability p. In this model, the probability of each graph with n nodes and m edges is $p^m(1-p)^{\binom{n}{2}-m}$
- $G(n, m)$: a graph is generated by placing m edges at random in the network of n nodes. This procedure corresponds to uniformly draw a graph from all the possible graphs with n nodes and m edges.

In any random graph, any quantity can be seen as expected value, or ensemble average, of the actual quantity. In the $G(N, p)$ model, the number of edges is, on average:

$$\langle m \rangle = \binom{n}{2} p = p \times n(n-1)/2\,, \tag{2.22}$$

and the average degree as:

$$\langle k \rangle = \frac{2\langle m \rangle}{n} = (n-1)p \;\approx np\,. \tag{2.23}$$

In $G(n,p)$, there are $n-1$ independent trials for each node to generate a link to other nodes. Thus, we can obtain the degree distribution formula:

$$P(k) = Bin((n-1),p) = \binom{n-1}{k} p^k (1-p)^{n-1-k} . \tag{2.24}$$

The degree distribution, for network size that goes to infinity ($n \to \infty$), becomes a Poisson:

$$P(k) \to \frac{\langle k \rangle^k}{k!} e^{-\langle k \rangle} . \tag{2.25}$$

When the average degree is small, there is a relatively few amount of edges, thus we say that we are in a sparse regime, and the network is tree-like. The expected clustering coefficient of an ER graph is p, since the probability of having an edge between any pair of nodes is p. In a sparse regime, $c = p << 1$. On the other hand, when $p = 1$, all the nodes are connected to each other, thus the clustering coefficient is also equal to 1.

2.5.2 Watts-Strogatz model

The *Watts-Strogatz* (WS) model [WS98] is a random graph model that addresses the limitation of the ER model to have a low clustering coefficient. Watts-Strogatz model is a compromise between a regular graph, such as a lattice and an fully random graph. This model can (at least partially) explain the small-world phenomena (from here, the name "small-world" network), and its topology has been used to study also the spreading of infectious diseases.

The algorithm to generate Watts-Strogatz graphs takes as an input the number of nodes n, the mean degree k, that should be an even integer, and a parameter β, that is the probability of rewiring an edge. In the first step, the algorithm generates a regular lattice with n nodes, each one connected to k other nodes, that are the neighbors ($\frac{k}{2}$ per side). The generated graph has $\frac{n\,k}{2}$ edges. In the second step, the algorithm goes through every node n_i clockwise, and rewire every edge connecting n_i to its $\frac{k}{2}$ neighbors with probability β. The other endpoint of each rewired edge is chosen uniformly at random, avoiding self-loops and multi-edges.

Thus, with $\beta = 0$, the generated graph is just a regular lattice, while with $\beta = 1$, the generated graph is a random graph similar to ER, but not the same since every node is anyway connected to at least $\frac{k}{2}$ other nodes. Rewiring has the effect of decreasing the average path length in the network, creating a short-cut between two nodes.
Figure 2.4 shows three different graphs generated with the WS model, with

different values of β.
The clustering coefficient C is, in the ring lattice topology, equal to:

$$C = \frac{3(k-2)}{4(k-1)}\,. \tag{2.26}$$

As k increases, C tends to $\frac{3}{4}$, becoming independent from the network size [BW00]. C has a value closer to the ring lattice one for small values of β. As β increases, the clustering coefficient drops, as in random networks, and, in the limiting case, is inversely proportional to the network size and assumes a value equal to:

$$C = \frac{k}{n-1}\,. \tag{2.27}$$

A drawback of the WS model is the unrealistic degree distribution. Most real-world networks are scale-free, presenting a heavy-tailed distribution in which there exist nodes with various degrees.

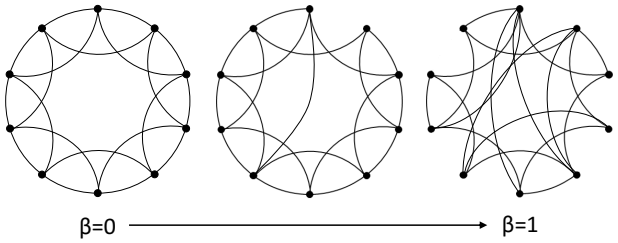

Figure 2.4: Three different cases of the Watts-Strogatz model, depending on different values of the parameter β. As β grows, randomness increases.

2.5.3 Random Geometric Graph model

A Random Geometric Graph (RGG) [Pen+03] is a model for spatial networks, constructed by placing n nodes in a space (according to a given probability distribution), and then connecting a pair of nodes with a link with if the distance between them is within a certain radius. In social networks, Geometric Random graphs can be used to model interactions of people in a space, since two nodes are more prone to interact if they are close to each other. This model leads to the creation of networks with local structures (such as communities). Thus, if the radius is relatively small, the network will be locally dense, but globally sparse.

There exist various types of random geometric graphs, and we focus on *Soft Random Geometric Graphs*. A Soft Random Geometric graph is constructed by uniformly and randomly placing n nodes in the space. If two nodes are within a given radius, they are connected with a specified probability (normally an exponential distribution). An interesting result of the study of Soft Random Geometric Graphs is that, in a high dense regime (such as when the average degree is high, and consequently the network is dense), with an exponential distribution controlling the connections, there is a unique giant component and just isolated nodes. Thus, if we enforce the graph to be without isolated nodes by controlling the parameters of the model, in particular the radius, the network will be fully connected [Pen+16].

The expected average degree of RGG is roughly:

$$\langle k \rangle \approx \pi(n-1)r^2\,, \tag{2.28}$$

where n is the number of nodes and r is the radius.

The network is sparse until the radius is $\Theta(ln(n))$. After this value, the network is dense. With $n \to \infty$, we can have an indication of whether the expected average degree grows quicker or slower than $ln(n)$ by computing the ratio $\frac{nr^2}{ln(n)}$. The reason of multiplying r^2 by n is that, from Equation 2.28, with a fixed average degree, the value of r^2 depends also on n.

2.5.4 Configuration model

The configuration model (originally introduced in [BC78]) is a model that generates random networks with a fixed degree sequence. The configuration model was mainly developed to provide a more realistic variant to the ER model, that produces a non-realistic degree sequence. Indeed, real-world networks do not present a Poisson distribution, but a heavy-tailed one, meaning that most nodes have a tiny degree [New18]. We use the configuration model as a null model in Chapter 7, to analyze the structure of the neighborhoods of real data, through a tool called network motifs, as explained in Section 2.6. The configuration model is a common choice as a null model, mainly to understand if some particular properties of a real network are related to its degree sequence or are actually relevant. This model is commonly used, for example, both for network motifs detection and modularity calculation[1].

[1]Network modularity measures the division of the network into module [New06]. High modularity means that there are densely connected nodes organized in "modules", that are though sparsely connected to each other

To generate a network, the configuration model takes a degree sequence $[k_1, k_2, k_3, ..., k_n]$ as an input, and outputs (after having checked that the degree sequence is realizable through the Erdős-Gallai test [EG60] [2]) a network with the same degree sequence (thus with the same size). In practice, the output is uniformly drawn by the set of all the networks with the given size and degree sequence. Different variants of the configuration model exist [Fos+18]. The generative algorithm of the model used in Chapter 7 consists of a loop that, iteratively, picks a pair of *stubs* (such as nodes with a number of edges incident to them, corresponding to their degree, as shown in Figure 2.5), uniformly drawn from the set of existing stubs (one for each degree of the input degree sequence), and connects them by joining a link from both. The procedure is repeated until no links are left. This method could lead to the creation of multiple edges or self-loops. However, the algorithm we adopt is the one presented in [MW90], which takes care of eliminating this kind of cases by rewiring the edges involved in self-loops or multiple connections.

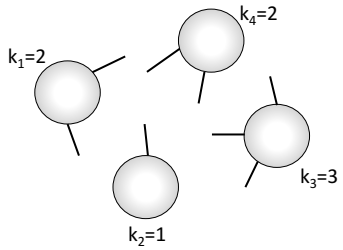

Figure 2.5: Examples of stubs

2.6 Network Motifs

Network motifs are induced subgraphs that are recurrent in networks. We make use of network motifs in Chapter 7 to understand the patterns that compose neighborhoods in social networks. Network motifs have been originally presented in [Mil+02], where are defined as "basic building blocks of

[2]The Erdős-Gallai theorem states that a sequence of integers d_i can be a degree sequence if the sum of the integers d_i is even and, for every k_i such that $1 \leq k \leq n$, the following condition holds: $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} min(d_i, k)$

complex networks". They have been used for studying both biological networks, such as protein interactions [Won+11] and social networks [JKM08], and recently also multilayer networks, such as brain networks [Bat+17] and corporate networks [Tak+18].

To be defined as motif, a subgraph (or, more precisely, an isomorphism class) should occur in a network a number of times that is statistically significant, and to decide that, each subgraph count is compared to the number of occurrences in a null model, such as a random network specifically chosen to compare with the network we are analyzing. Since the computation of large subgraphs can be expensive, normally the motifs analysis is restricted to graphs with a small size, for example with five nodes at most. Depending on the type of network we are analyzing, motifs can be either uncolored, undirected, colored or directed graphs. For each isomorphism class of subgraphs of our interest that occurs in the original network, we should count the realization of that class in the null model. Being the null model a random network, we should generate it many times and count the subgraphs realization every time, and then compute the mean and the variance for any subgraph realizations. At this point we can compute the so-called *Z-Score* for each subgraph G' in the network G, defined as:

$$Z(G') = \frac{F_G(G') - \mu_R(G')}{\sigma_R(G')}\,, \tag{2.29}$$

where $F_G(G')$ is the frequency of the G' in the original network, $\mu_R(G')$ and $\sigma_R(G')$ the mean and the variance of the occurrences of G' in the null model. If the Z-Score is sufficiently high, then G' is over-represented in the network G in comparison to the null model R, and can be called motif. Conversely, if the Z-Score sufficiently low, then the G' is under-represented in the G compared to the null model R, and it is called anti-motif. Practically, a certain threshold is normally chosen to classify a subgraph as a motif or anti-motif. If μ_R is equal to F_G, the subgraph occurs, on average, the same amount of times in the null model and in the network; if σ_R is zero, then either the number of realization of the null model is not enough, or the subgraphs count has every time the same value (for example, this can happen for isolated nodes in the network, when the null model has the same degree sequence of the original network). The number of subgraphs in a graph is assumed to be normally distributed, and, for this reason, each of them has a mean and a variance. More realization of the null model we do, better estimation of the real mean and variance we obtain.

Since the value of the Z-Score can be affected by the network size, if we want to compare various networks, we can normalize the length of the

Z-Score, computing the *significance profile* SP for each motif i, as:

$$SP_i = \frac{Z_i}{\sqrt{\sum_i Z_i^2}}\,. \tag{2.30}$$

An alternative measure to the Z-Score for motifs detection is the p-value, interpreted as the probability that a subgraph is higher in the original network than in the null model. If the p-value is higher than a certain threshold, then the corresponding subgraph is considered as a motif. However, we just use the Z-Score in our study.

The choice of the null model influences motif detection [SZ15]. The null model can be built specifically for a certain study, or can be one of the well-known random graph models, such as Erdős - Rényi or the configuration model.

The most simple methods to discover motifs is an exhaustive search of subgraphs in the network. However, this is infeasible in big networks. For this reason, many more efficient algorithm based on different sampling techniques have been developed [Kas+04; Wer06].

2.7 Privacy definitions

With the increasing amount of personal data and data analysis techniques, the risk for users' privacy has dramatically raised and, consequently, the study and development of privacy-preserving techniques for data analysis has seen great attention among scientists. The existing privacy-preserving and anonymization methods rely on different definitions of privacy, that have been mainly developed for data like vectors. Only later, those definitions have been extended to networks.

Some of the most popular existing definitions of privacy are:

- *naïve anonymization*: replacing the entities/users (or, in graphs, the nodes) IDs with random numbers. We can also consider a network naïvely anonymized when all attributes associated to nodes are dropped (thus the graph is transformed in an unlabelled graph);
- *random perturbation/noise injection* [Kar+03]: randomly modifying or distorting the data; an alternative is to replace the data with some drawn from the same probability distribution to which some noise has been injected with either an additive or multiplicative approach;
- *k-anonymity* [Swe02]: each entry in the released dataset it is indistinguishable from at least other $k-1$ entries (or, in other words, each equivalence class contains at least k values);

- *ℓ-diversity* [Mac+06]: it is a group based anonymization for labelled data, where the labels or attributes can either be sensitive or not; an equivalence class (which non-sensitive attributes have been anonymized, or generalized to a certain value) present in a dataset satisfies ℓ-diversity if it contains at least ℓ different values for the sensitive attributes fields. A released dataset satisfies ℓ-diversity if all its equivalence classes satisfy ℓ-diversity;

- *t-closeness* [LLV07]: an equivalence class present in a dataset satisfies t-closeness if the distance between the distribution of a sensitive attribute present in this class and the distribution of the same attribute in the whole dataset is no more than a threshold t. A dataset satisfies t-closeness if all its equivalence classes satisfy t-closeness (the distance is meant to be the distance between two probability distributions);

- *differential privacy* [Dwo06; Dwo08; D+14]: is a technique that is applied to queries on datasets, ensuring the privacy of the response. Given two adjacent datasets x and x' (such as that they differ for one element), an algorithm $\mathcal{A}$, which takes a dataset as an input, is $\epsilon - \delta$ differentially private if, for all subsets S of the image of $\mathcal{A}$, $S \subseteq im(\mathcal{A})$, the following holds: $\Pr(\mathcal{A}(x) \in S) \leq e^{\epsilon} \Pr(\mathcal{A}(x') \in S) + \delta$. This formula means that the probability of observing S in two adjacent datasets x and x' (for example, a perturbed one and its original copy) differ by an additive and multiplicative factor. ϵ is called the privacy budget and δ is optional, depending on the definition we want to use. Algorithms that respect the differential privacy definition ensure a randomized response, protecting, in this way, the privacy of the individuals represented in the dataset the query is run on.

Those techniques can be applied to data attributes to modify them (or, in the case of differential privacy, to queries to ensure a private, randomized response) and to suppress the uniqueness of the attributes, and make the entities less identifiable. In a dataset, information related to entries are called *quasi-identifiers*. Quasi-identifiers are attributes that do not allow to identify an entry themselves uniquely, but that create a *unique identifier* combined with other quasi-identifiers. In the literature, some studies showed this property of quasi-identifiers with real data. For example, [De +13; PMS18; D+15] have uniquely identified individuals based on, respectively, locations visited, metadata associated to social media usage, and credit card transactions.

k-anonymity is one of the first formal definition of privacy, with a formal probabilistic interpretation of re-identification. l-diversity and t-closeness

were developed to strengthen the definition of privacy given by k-anonymity, addressing some attacks on it.

Differential privacy is instead a modern concept that provides stronger privacy guarantees with a more flexible definition. It has seen increasing attention among scientists, and it has started to be applied also to networks [TC12; Che+14]. Differential privacy was initially developed just to address some particular queries, while, recently, the study of methods for data sharing has also increased [Zha+17]. Some methods have been developed to perform specific network analysis tasks, such as mining frequent subgraphs [SY13], estimating the degree distribution [Hay+09], performing exponential random graphs estimation [LM14]. Most of those methods, however, aim to protect just some information, such as the ones related to nodes (node differential privacy [Kas+13]) or edges (edge differential privacy).

In the next chapter, we see how some of these concepts of privacy are applied to networks and, in particular, what is the role of network's neighborhoods in privacy and anonymization.

2.8 Tools used

The code developed for producing the results of this thesis has been written in *Python 2.7* [Ros95]. The library used for constructing and analyzing multilayer networks is *Pymnet* [Kiv17], and the software for conducting graph isomorphism tests is *Bliss* [JK07] (and its Python wrapper *PyBliss*).

Chapter 3

Literature Review on Neighborhood Attack

In this chapter, we present the main literature on neighborhood anonymization and privacy in social networks (Section 3.1). We also survey the methods that have been developed to prevent neighborhood attack (Section 3.3). Moreover, we illustrate the existing methods for anonymizing multilayer networks and edge-labelled graphs (Section 3.4). The focus of this thesis is neighborhood attack on multiplex networks (a particular type of multilayer networks) and, despite no methods currently exist to address this problem, an overview of both neighborhood attacks on classical single-layer social networks and other types of attacks on multilayer networks (or edge-labelled graphs, that are, for certain aspects, similar to multiplex networks) can be useful to understand the current state in the understanding of this problem.

3.1 Neighborhoods and privacy

More and more data can be modelled as networks nowadays, for instance, social media connections or phone calls. Since networks can be interesting objects also just for their structure, without attributes associated to nodes, one can think that a naïve anonymization approach could be enough for sharing them. However, some structural features can make a node unique and thus re-identifiable in a network, such as the amount of its connections (degree) or the structure of its neighborhood. In this sense, privacy definitions have been adapted to networks, to modify the data to reach, for example, k-degree anonymity [LT08] and k-neighborhood anonymity [ZP08]. The k-neighborhood anonymity tries to prevent the *neighborhood attack*, such as the re-identification of a node based on its neighborhood.

Another approach to anonymize networks, based on modifying neighborhoods, is *neighborhood randomization* [FW15]. This approach consists in changing the endpoint of an edge within the local neighborhood of a node. Neighborhood randomization provides link privacy, and it is one type of random perturbation method. In particular, it is a link perturbation method. Alternative link perturbation methods, not aiming to protect neighborhoods, add a certain amount of edges randomly to the network [YW09] while deleting the same amount, or perform random edge switching.

An alternative to the classical neighborhood attack is the *neighborhood-pair attack* [NA13], which consists in the re-identification a node based on the structure of neighborhoods of two connected vertices. Since the attacker's knowledge is broader than a normal neighborhood attack scenario (where only the structure of a target's node neighborhood is known), this attack has higher re-identification risk than the classical neighborhood attack.

[Hay+07] and [Hay+08] study and formalize the risks of some structural attacks on social networks, in particular the ones in which the attacker's knowledge consists in the degree and the neighborhood subgraph of a node at various levels (or *hops*). For example, the attacker can know only the degree of a node, or, additionally, the degree of its neighbors, etc.. Equally, the neighborhood graphs can be 1-hop or of higher order. Specifically, the first paper [Hay+07] studies the re-identification risk for both degree attacks and subgraphs (or neighborhood) attacks at various levels on some single-layer real world dataset. The second paper [Hay+08] studies the degree attacks at various levels in both real-world data, synthetic data (such as power law, tree, or grid topology graphs) and ER random graphs. In particular, for the latter, the authors distinguish three cases, corresponding to different regimes and edge probabilities: sparse, dense, super dense. They conclude that n the dense and super-dense regime, a node is easily identifiable. In the dense regime, a node is identifiable when the attacker's knowledge includes at least the degree of the neighbors, besides the degree of the node itself. For networks in the sparse regime, as most real-data are, the re-identification probability depends on the network size. We also study how the uniqueness changes in ER graphs and other graph models in Chapter 5. However, differently from those previous works, we also take into account the full 1-hop neighborhood subgraph of a node. [NS09] presents a de-anonymization method. This work also illustrates that the re-identification is easier if the attacker has, besides the knowledge of certain subgraphs, even partly additional information coming from another social graph, showing how different percentage of node and edge overlap ov_E affect the attack's success probability. In Chapter 5 we also study the effect of edge overlap in multilayer networks, but more rigorously, and focusing only on neighborhoods. We

properly define the concept of multiplex neighborhoods (in Chapter 4) and systematically analyze how different values of edge overlap, average degree and networks size affect the fraction of nodes that are easily re-identifiable in different network models.

In general, networks are harder to anonymize than other types of data (such as vectors of values), because, besides the possible attributes of nodes, there are also links, that can reveal information about relations between nodes. An even more difficult task can be the anonymization of multilayer networks, since there are multiple types of links. Few methods are addressing the anonymization of networks with multiple types of links, and some of those are discussed in Section 3.4.

3.2 Neighborhood attack: an overview

In this section, we present an overview of the existing methods to prevent neighborhood attack in social networks, that we discuss in more details in the next sections. In a neighborhood attack, the attacker's goal is to identify one or more nodes t_i in a released network dataset based on their neighborhood structure. The attacker's initial knowledge consists of one or more graphs P_i corresponding to t_i's neighborhoods. To perform the attack, the attacker has to extract the neighborhoods of all the nodes, and then perform isomorphism tests of the resulting graphs against the initially known graphs P_i. All the extracted neighborhoods isomorphic to P_i are candidates for the target t_i. If there is just one candidate, then the victim t_i has been identified, and the attack has been successful.

There exist various types of neighborhood attack based on the attacker knowledge, that can be, for example:

- *1 or more hop-neighborhood structure*: the attacker knows just the neighborhood structure. If the attacker knows, for example, also the 2-hop neighborhood structure, the nodes can be more easily identifiable (however, most of the literature focus on 1-hop neighborhood attack);
- *labelled neighborhood*: in this case, the attacker knows also the labels of the neighbors of the target node;
- *partial knowledge of the neighborhood*: the attacker does not know all the neighborhood structure, but just a part of it.

Obviously, the higher the knowledge of the attacker, the higher the risk of re-identification is. For instance, in some cases, it could be enough to know

the structural information about the neighborhood to identify a node in a network, but, in some other cases, the attributes associated to the nodes (either the central node or the neighbors) can be crucial to identify the target uniquely.

Neighborhood attacks can be also classified based on the type of networks they are applied to, that can be labelled, unlabelled, weighted or unweighted.

The aim of neighborhood attacks countermeasures is to anonymize the network (mainly extending the concept of $k-anonymity$ [Swe02] to the one of $k-neighborhood\ anonymity$ [ZP08]), in order to suppress the uniqueness of certain neighborhoods by adding some noise (e.g. adding or removing edges or vertices from some neighborhoods) to the original data. In this way, for each existing neighborhoods, there will be at least other $k-1$ that are indistinguishable (i.e. that are isomorphic), given the assumed attacker knowledge. The successful attack probability for each neighborhood will then decrease from 1 (case of presence of unique neighborhoods' structure) to $\frac{1}{k}$.

Almost all the presented neighborhood anonymization algorithms in this chapter consist of two main steps: grouping (or clustering) the most similar nodes with at least k members each; transforming the graphs in each group to make them isomorphic to each other, through the application of edit operations. The main edit operation used in the literature is edge addition. By only adding edges or nodes, the resulting anonymized graphs is a subgraph of the original. Finding an optimal solution to the $k-neighborhoodanonymization$ [ZP08] problem means minimizing the number of edit operations needed to anonymize the network, and it is an NP-Hard problem, which becomes NP-Complete for edge-labelled graphs with $k \geq 3$ [ZP11].

The anonymization algorithms proposed in the literature are tested both on empirical datasets and network models (mainly Erdős - Rényi networks). To measure the quality of the anonymization, some network measures before and after anonymization, such as: average degree, average local clustering coefficient, average shortest path length between pairs of randomly picked nodes, amount of edges/nodes.

3.3 Neighborhood attack on social networks

In this section, we present the literature about the neighborhood attack in social networks. Table 3.1 shows a summary of the papers about this topic, the specific considered problem and the main steps of the adopted method to anonymize the network.

We start by illustrating the first paper considering this problem (in Section 3.3.1), along with its minor improvements and modification. Section 3.3.2 presents methods addressing label-neighborhood attack and networks dynamic publishing, Section 3.3.3 illustrates neighborhood attacks on weighted networks, while Section 3.3.4 refers to more general methods treating other structural attacks.

3.3.1 Seminal work on neighborhood attack and its improvement/modification

The authors of the seminal work [ZP08] on neighborhood attack have defined a k-anonymity approach (later extended to a *ℓ-diversity* one with another work, [ZP11]). They consider node-labelled graphs, with the labels forming a hierarchy (for example, there are general labels like "doctor", which are at a higher level of the hierarchy, while others, more specific, like "dentist", that are at a lower level). There is also a meta-symbol * which is the most general label (i.e. the root of the hierarchy). To perform neighborhood anonymization, they allow edge addition and label generalization.

Finding an optimal solution (e.g. adding the minimum amount of edges) to reach the anonymity is a NP-Hard problem[1]. To conduct isomorphism test, the neighborhoods are represented through a *neighborhood component code* (an alternative approach to the computation of a complete invariant). The neighborhood component code (NCC) is the ensemble of the minimum-DFS code [YH02][2] of all the neighborhood components, which are the *maximal connected subgraphs* in the neighborhood of the target node ($Neighboorhood(u)$, if u is the target node). Two neighborhoods of two different nodes u and v are then isomorphic if and only if $NCC(u) = NCC(v)$. Figure 3.1 shows the neighborhood of a node and its components.

The neighborhoods' connected components are used during the anonymization process to align two networks, in order to do less modifications as possible. The heuristic that [ZP08] adopts to anonymize two neighborhoods, consists in matching neighborhood's components pairs (starting with the

[1]The proof consists in reducing the $k - neighborhood$ anonymity problem to the $k - dimensional\ perfect\ matching$ [HSS03] (such as the problem of finding a maximal matching in a k-partite k-uniform balanced hyper-graph). The extended version of the work [ZP11] presents the whole proof.

[2]*Minimum DFS (Depth First Search) Code*: canonical representation of a graph, representing the edges through vertex pair IDs, sorted by the order in which the DFS algorithm visits them. A graph can have multiple DFS code, but the minimum DFS code is the one that respects a linear order on the label set of the vertices, if there is any.

Problem	*Graph type*	*Dynamic publishing*	*Anonymization method*	*Paper*	*Section*
(1-)neighborhood anonymization	Labelled nodes	No	Neighborrhood isomorphism checked through DFS code of the neighborhood Edge addition Label generalization according to label hierarchy	[ZP08]	3.3.1
			Neighborrhood isomorphism checked through DFS code of the neighborhood Edge addition Label generalization through l-diversity	[ZP11]	3.3.1
			Neighborrhood isomorphism checked through DFS code of the neighborhood Edge addition; Node addition (if the edges to be added is to a node above a certain distance, to minimize changes in nodes' distance) Label generalization through l-diversity	[OWK14]	3.3.1
	Unlabelled nodes	No	Neighborhood isomorphism checked through adjacency matrix Edge addition	[TP10]	3.3.1
	Unlabelled nodes; weighted network	No	Neighborhood isomorphism checked through adjacency matrix and neighborhood matrix (that includes edges' weights) Edge addition; Weight modification	[Liu+15]	3.3.3
Label neighborhood anonymization	Labelled nodes	No	Neighborhood isomorphism checked through neighborhood label sequence similarity Edge addition; Node addition Label generalization through super-label creation	[Wan+14]	3.3.2
General structural attacks	Unlabelled nodes	Yes	Partition the network in blocks, then align those to make them automorphic Edge addition; Node addition Nodes ID generalization for dynamic publishing	[ZCÖ09]	3.3.4
	Labelled nodes	Yes	Creation of k-pairwise isomorphic subgraphs Edge addition; Edge deletion Nodes ID generalization for dynamic publishing	[CFL10]	3.3.4

Table 3.1: Overview of the papers regarding neighborhood attack on social networks presented in this document.

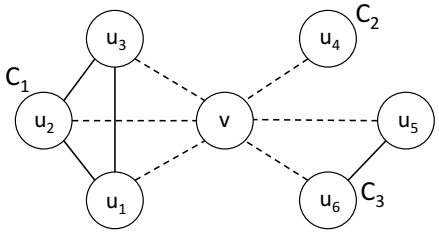

Figure 3.1: A neighborhood of a generic node u and its neighborhood components C_i, which are the connected components in the neighborhood (Figure adapted from [ZP08]).

ones with the highest amount of vertices) and making them isomorphic if they are not already.

To assess the quality of the anonymization, the authors compute an anonymization cost. The anonymization cost takes into account three factors (which different impact can be tuned by three parameters): the nodes label's changes (with the so-called *normalized central penalty*[3]); information loss due to adding edges; number of vertices introduced to a neighborhood to make it anonymous. The choice of neighborhoods pair from the two networks that need to be anonymized is done by matching the degree of the central node. If there are no matching pairs are found, then the pair with the minimum anonymization cost is chosen.

A later extended version of the seminal paper [ZP11] contains an $\ell - diversity$ method for anonymizing a social network where nodes are labelled. In fact, the previous approach can lead to information leak when, in the presence of labelled graphs, the attacker does not know the labels of the neighborhood of the target node. Indeed, with $k - neighborhood anonymity$, with labelled graphs, two neighborhoods with the same structure and the same labels associated are identified by a neighborhood attack, the attacker gets to know the information carried by the labels in any case (even though the exact neighborhood is not identified). The considered types of networks in [ZP11] have nodes with both sensitive and non-sensitive labels (that represent, for example, information that people want to share or not). A social network is considered to satisfy $\ell - diversity$, if, after having partitioned the network' nodes in groups (called *equivalence groups*) and anonymized those, in every equivalence group of nodes, at most $\frac{1}{\ell}$ of the

[3] *Normalized central penalty*: supposing that a node with a label l_1 needs to be generalized to l_2, the normalized central penalty is $\frac{size(l_2)}{size(*)}$, where $size(l_2)$ is the number of l_2 descendent that are leafs in the hierarchy, and $size(*)$ is the number of leafs in the label hierarchy.

vertices are associated with the most frequent sensitive label. Thus, the attacker that knows the 1-neighborhood structure of a victim can infer its sensitive label with a probability not larger than $\frac{1}{\ell}$ (thus, the larger ℓ, the better). This approach gives stronger privacy than $k - anonymity$ and, in any case, the $k - neighborhood\, anonymity$ is still respected.

The paper [TP10] proposed an alternative to the seminal work [ZP08] in terms of matching the neighborhood components, representing them with an adjacency matrix instead of the neighborhood component code. In the adjacency matrices, the vertices are listed in decreasing order of degree and labels (according to the hierarchy). The isomorphism test is then done by checking both the adjacency matrix and nodes' labels. In the case of neighborhoods of different size, the comparison is done on the first sub-matrices of the adjacency matrices of the components with the largest amount of nodes. Instead, in case of no match, edges or vertices need to be added for making two neighborhoods isomorphic. In this way, it is possible to compute the difference of two node-aligned adjacency matrices with the same power, obtaining the position of the edges that need to be added to make them isomorphic. This approach with adjacency matrix can also be easily extended to more than 1-hop neighborhood.

One problem in the neighborhood anonymization methods presented until now in this section consists in consequence of edge addition. Indeed, fake edges could significantly change the distance between nodes. [OWK14] proposes an algorithm to minimize those changes. This method differs from the seminal one in the anonymization process, in particular when an edge needs to be added to a neighborhood, consequently including an additional node to the neighborhood itself. In this work, the "new" chosen node is the one that has the smallest distance (computed by Dijkstra's shortest path algorithm) from the central node of the neighborhoods. If all the possible distances are higher than a certain threshold (chosen by the user), then a fake node is added.

3.3.2 Label neighborhood attack

In this section, we present the label neighborhood attack problem, introduced in [Wan+14]. In the label neighborhood attack, the attacker has a background knowledge not just of the neighborhood structure, but also about the labels of the nodes in it (conversely to the attacks presented in subsection 3.3.1, where the nodes are labelled, but the attacker knows just the structure of the target's neighborhood). The labels can be both sensitive and non-sensitive. During the anonymization process, the nodes and their neighborhoods are grouped, considering not only the neighborhood

structure, but also the additional constraint of having in each group nodes with a certain amount of different sensitive labels (such as attributes which privacy needs to be protected). To make neighborhoods isomorphic, the adopted method allows edge addition (and, if needed, fake nodes addition). In the end, the network should respect *ℓ-graphic-diversity*, defined as:

($\ell - Graphic - Diversity$): For each node u ∈ V(G) that attaches with a sensitive label, there must be at least $l - 1$ other nodes with the same labelled neighborhood graph, with different sensitive labels.

Also in this work, the nodes with higher degrees are considered first for anonymization, mainly treating the nodes with the lowest degree to add in neighborhoods when needed. A difference between this work and the seminal one [ZP08] is that labels do not form a hierarchy, but the generalization is done by creating a super-label, which contains multiple single labels. The information loss due to the application of this technique is measured by

$$LGC(l_u, l'_u) = 1 - \frac{|l_u \cap l'_u|}{|l_u \cup l'_u|}, \tag{3.1}$$

where l_u is set of node u's labels in the original network, while l'_u is the one in the anonymized network. The rest of the information loss is measured as in the seminal work. The full algorithm performance is evaluated by comparing the degree distribution of the network before and after the anonymization, the average local clustering coefficient, and the average shortest path length between nodes with different labels.

3.3.3 Neighborhood attack on weighted social networks

The paper [Liu+15] is about the $k-neighborhood$ anonymization of weighted unlabelled networks. Here, the allowed edit operations to reach $k - neighborhood\ ismorphism$ are edge addition and weight modification. To represent neighborhoods, the authors use two matrices: the usual adjacency matrix (without the central node) and the neighborhood matrix, consisting of two rows, one representing the neighbors and the other the weight assigned to the edges between the central node and its neighbors (there could be also a weight of 0 in the case a weight is not assigned). The grouping of similar neighborhoods is based on those two matrices, and the various groups (or clusters) are sorted based on the number of neighbors and number of neighbors' mutual edges. The sorting process is repeated every time a modification of neighborhoods occurs. Once two neighborhoods are

made structurally isomorphic by edge addition (with the constraint that just one new neighbors can be added to each neighborhood in the process), the weights in the neighborhood matrix can be permuted or, if this is not enough, they can be modified. To measure the method's performance, the authors compute the edge addition rate and the weight change rate.

3.3.4 Protecting the privacy in social networks against general structural attacks

The two papers presented in this section are about *k-automorphism* [ZCÖ09] and *k-isomorphism* [CFL10], two techniques that addresses the graph anonymization problem to prevent structural attacks, including, but not limited to, neighborhood attacks. Both methods are based on first partitioning the nodes in groups that do not share any node, called *non-overlapping groups*. For the anonymization purpose, k-automorphism allows edge and vertex addition, while k-isomorphism allows edge addition and deletion. k-automorphism is applied to unlabelled graphs, while k-isomorphism to labelled ones. k-isomorphism implies k-automorphism, and, in general, gives stronger privacy guarantees. Both methods address dynamic releases, generalizing the vertices ID to protect the privacy, even if in a slightly different way. Indeed, the one in the k-isomorphism paper is more flexible because the time snapshots are treated independently from each other.

The k-automorphism method [ZCÖ09] is based, as most of the methods presented above, on partitioning the graph in k blocks, and then aligning two graphs by edge addition, or, more specifically, edge copy, in order to obtain a network in which, for each vertex, there is another symmetric vertex. The authors defines the anonymization cost as follows:
Anonymization Cost*: Given an original network G and its anonymized version G^*, the anonymization cost in G^* is defined as:*

$$Cost(G, G^*) = (E(G) \cup E(G^*)) - (E(G) \cap E(G^*)), \tag{3.2}$$

where $E(G)$ is the set of edges in G, and $E(G^*)$ is the set of edges in G^*.

The method also handles dynamic network release, which is the process of continuously sharing networks that changes over time. Each release is done at different time-stamps and consists of a network itself. Each vertex has associated a vertex ID in order to keep track of it in each release. This IDs are then generalized (such as grouped together, for example, two vertices with IDs 1 and 2 become both {1,2}), respecting the definition of k-automorphic network above. Thus, if the output of k-1 automorphic functions, that take the same vertex as input, at different times is not the same, the vertices' IDs

are generalized. In the case that vertices are also added or deleted during the dynamic release of the network, the anonymization becomes more difficult, because the attacker could know about the presence or absence of a vertex at different timestamps. In those situations, the generalization aims to have the same generalized IDs present in all releases.

The method is evaluated by comparing the average shortest path length between some pairs of vertices, the local average clustering coefficient, and degree distribution before and after the anonymization.

K-isomorphism [CFL10] has been developed to address some issues of k-automorphism, which protects nodes privacy but can also not protect links privacy. For example, in a k-clique, where all the nodes are connected to all the other nodes by an edge, the attacker cannot precisely indicate two nodes that are target of its attack. Indeed, when looking only at the structure of the network, all the nodes are indistinguishable to each other. However, the attacker can still know that there is at least an edge that connects the two nodes. K-isomorphism aims to protect information related to both nodes and links. In the anonymization procedure, the network is partitioned into groups of disjoint graphs, with the same number of vertices. Then the graphs within a group are anonymized with edge addition or deletion.

The dynamic release of networks is handled by defining a vertex mapping table in which each vertex ID is mapped to other vertices through some isomorphic functions. The list of ID's isomorphic to each vertex (called "compound") replaces the actual vertex ID of every vertex in the subgraph. Regarding vertex deletion in successive releases, the single vertex ID be present in the compound anymore, and in the vertex addition case, the new ID will be present in a compound as the other vertices. In each release, k-isomorphism should be maintained, by performing the anonymization algorithm again.

Finally, the amount of modification introduced by the algorithm are evaluated by comparing the degree distribution and shortest path length of some randomly selected nodes. k-isomorphism performs better than k-automorphism in the comparison of dynamic network release, since it measures the proportion of generalized vertex IDs relative to all vertex IDs in each release.

3.4 Anonymization of multilayer and edge-labelled graphs

In this section, we present the main literature about the anonymization of multilayer networks and edge-labeled graphs. We are interested in multiplex networks, which are, as defined in Section 2.2, both a type of multilayer networks and, in particular, edge-labelled multigraphs. This section can give us an overview of the state of the art of anonymization on those kinds of systems, that are very close to the ones of our interest.

Table 3.2 gives an overview of the papers we present, along with the considered problem and a summary of the used techniques. We first present, in Section 3.4.1 the summary of the results of two papers regarding the complexity of edge-labelled graph anonymization. Then, in Section 3.4.2, we present other two works about label-bag anonymization, which is the problem of making a node indistinguishable from the others based on the labels of the edges incident to it. Finally, in Section 3.4.3, we summarize a regarding k-degree anonymization in multilayer networks.

3.4.1 Complexity of neighborhood attack in edge-labelled graphs

[KSV11] and [Che+13] present and prove the hardness of different edge-labelled social network anonymization techniques, reducing the problems to a table k-anonymization problem, where the rows of the table represent the entries, and the columns the various attributes associated to them. The table anonymization problem is NP-Hard.

The paper considers the anonymization of the *label sequence* of a node, such as the list of the edges' labels that are incident to each vertex. This

Problem	***Graph type***	***Dynamic publishing***	***Anonymization method***	***Paper***	***Section***
Label-Bag Anonymization: k-anonymization of a node based on its label bag (set of the labels of the edges that are incident to it)	Edge-labelled graphs; unlabelled nodes	No	Edge addition	[Li+14]	3.4.2
k-degree anonymization of multi-layer (time varying) networks	Multilayer (time-varying); unlabelled nodes	Yes	Temporal degree vector modification	[RMT15]	3.4.3

Table 3.2: Overview of the papers regarding multi-layer networks and edge-labelled graphs anonymization presented in this document.

problem is called Label Sequence-Based Subset Anonymization Problem (LS-SAP), and it is defined as follows:

*Definition (**Label Sequence-Based Subset Anonymization Problem (LS-SAP)**): Given an edge-labeled graph $G = (V, E, \Sigma)$, $X \in V$, and an integer k, find an edge-labeled graph $G' = (V, E \cup E', \Sigma \cup \Sigma')$ such that X is k-anonymous in G' and the number edges added, $|E'|$, is minimized.*

LS-SAP is NP-Hard, and NP-Complete for $k \geq 3$. This complexity is the same as the k-neighborhood anonymization problem. In fact, k-neighborhood anonymization is reduced to k-label sequence anonymization, since, in both the problems, edges needs to be added to make two graphs isomorphic.

3.4.2 Label-Bag anonymization

The label-bag anonynimization problem [Li+14] is the problem of anonymizing edge-labelled graphs, equivalent to LS-SAP introduced in the previous section. A *label-bag* (LB) is indeed the ensemble of labels of edges incident to a vertex. The anonymization method proposed still has a grouping step, where the aim is to group nodes with similar label-bag. Different grouping strategy, such as hierarchical clustering or feature-based grouping (based on the similarity of the label bags, like Jaccard similarity), are discussed and compared. In every group, the union of the label bags of the nodes within the group is called Target Label Bag (TLB). Instead, the union of the label bags of the nodes within the same group, not counting the node itself, is called Residual Label Bag (RLB). TLB and RLB are used for the edge addition. The edges are added between two nodes that have at least a common label in their RLB and that are not directly connected already. The process is then iterated until the graph is LB k-anonymous (the RLBs should be empty at the end).
An improved version of the method is presented in [Li+], where the edge addition is done taking into account also the maximization of the utility, trying to minimize the distance between the measures of the network before and after anonymization, such as degree distribution, average local clustering coefficient, average shortest path length between randomly chosen nodes.

3.4.3 k-degree anonymization of multilayer networks

[RMT15] studies the k-degree anonymization of multilayer (undirected and unlabelled) social networks, focusing on temporal networks. In the considered temporal networks, each layer corresponds to a particular temporal slice. The difference between this work and the others addressing dynamic network publishing, it is that here one can keep track of a single node during different timestamps, while in previous works (for example in the papers presented in 3.3.2 and 3.3.4) the vertices ID were generalized, thus there was not a real one-to-one correspondence between nodes in different temporal snapshots. The nodes grouping and consequent anonymization is based on their temporal degree vector (the vector containing the degree of the node in each layer). After having anonymized the nodes, it might be necessary to enforce realizability of the various degree sequences, to respect the Erdős-Gallai theorem. This step is done by solving a linear programming problem, projecting an unrealizable degree sequence to the nearest realizable one.
The cost of the anonymization is computed by:

$$cost\, C(D, D^*) = \sum_{i=1}^{n} \sum_{i=1}^{T} \frac{d_i - d_i^*}{Tn(n-1)}\,, \tag{3.3}$$

where D and D^* are the matrices formed by the temporal degree vectors, n is the number of nodes, T the temporal slices, d and d^* the single degree vectors.
The grouping step is done with a modified version of the *k-means* clustering algorithm, run multiple times, selecting the realization with the lowest cost. However, since this approach is not really suitable for large graphs, a greedy algorithm is also presented as an alternative.
The authors found that the more correlated the temporal slices are, the easier defining anonymity groups is, and the higher the temporal resolution is (for example from one month to one week), the higher the anonymization cost is.

Chapter 4

Multiplex Neighborhoods

The aim of this chapter is to define the neighborhoods and the settings of the neighborhood attacks in multiplex networks. We will need these definitions in the next chapters, where we want to understand the difficulty of the anonymization problem to prevent neighborhood attacks on networks. The major obstacle for the existing anonymization algorithms presented in the previous Chapter 3 is the presence of unique neighborhoods (at least if we consider *k-anonymity*, with $k = 2$), because they are easily identifiable by an attacker that knows the target nodes' neighborhood structure and performs an exhaustive subgraph search on the complete network. Understanding the difficulty of the problem given simple network features (such as the number of nodes, average degree, or, in the multiplex case, number of layers and proportion of overlapping edges between different layers) is important to assess the privacy risk when it comes to data sharing.

In order to conduct an analysis on how the uniqueness varies in both network models (Chapter 5) and empirical data (Chapter 6), we need to first define multiplex neighborhoods, and understand how an attacker can extract those from the network in order to look for the target node. We show that multiplex neighborhoods can be defined in multiple ways, and this also affects the isomorphism classes to which each neighborhood belongs and, consequently, the uniqueness value of the network (which is, as we define in subsection 4.1.4, the number of unique neighborhoods in the network). We also define the different types of uniqueness we discuss (according to the network type, monoplex or multiplex, or to the isomorphism type we consider) and, finally, illustrate the hypothesis under which we work (e.g., the types of dataset we are taking into account) and the attacker model for the neighborhood attack in multiplex networks (Section 4.2).

4.1 Neighborhoods in Multiplex Networks

In this Section, we present two possible definitions of multiplex neighborhoods (subsection 4.1.1) and discuss how they can lead to different uniqueness values (subsection 4.1.2). We then discuss how an attacker can practically extract those neighborhoods from a network, to perform the neighborhood attack (subsection 4.1.3). We also present the different definitions of uniqueness, which is a central concept in this thesis (subsection 4.1.4), and discuss about the count of isomorphism classes for multiplex neighborhoods and, in general, for multiplex network (subsection 4.1.5).

4.1.1 Multiplex neighborhoods definition

A 1-hop multiplex neighborhood is the multiplex version of the 1-hop neighborhood in a normal graph (the definition can be naturally extended to multiple hops, but we focus just on nodes' immediate neighbors), as defined in Section 2.1. We defined the neighborhood of a node v in a simple network G as the subgraph induced by v's neighbors (Equation 2.2). Similarly, to define multiplex networks' neighborhoods, we need to first define *multiplex subgraph*:

A ***multiplex subgraph*** M^* of a multiplex network $M^{=}\{G_i\}_{i=1}^{d}$ with layers $\hat{L}$ is defined as:

$$M^* = \{G_i^*\}_{i=1}^{d}\,, \tag{4.1}$$

where G_i^* is a subgraph of G_i, such as the graph in layer L_i of M. M^* has the same layer set $\hat{L}$ as M.

Given a multiplex network M, with layers $\hat{L}$, and all the monoplex networks G_i in each layer L_i (according to the definition in Section 2.2), we identify two different ways of defining a multiplex neighborhood of a node v in a multiplex network M:

- ***Non-Inclusive Multiplex Neighborhood*** ($\mathcal{N}_{\subset}(v)$) is a multiplex network with the same layers L_i as in M. Each single-layer network of $\mathcal{N}_{\subset}(v)$ in layer L_i is the induced subgraph $G_i[V_i^{\subset(v)}]$ of G_i, where $V_i^{\subset(v)} \subset V_{G_i}$ is the set of neighbors of v in layer L_i:

  $$\mathcal{N}_{\subset}(v) = \{G_i[V_i^{\subset(v)}]\}_{i=1}^{d} = \{V_i^{\subset(v)}, E_i^{\subset(v)})\}_{i=1}^{d} \tag{4.2}$$

  $\mathcal{N}_{\subset}(v)$ is generally not node-aligned, because some nodes that are neighbors of v in L_1, can also not be neighbors in L_2, and thus are not included in the network;

- ***Inclusive Multiplex Neighborhood*** ($\mathcal{N}_{\subseteq}(v)$) is a more inclusive version of $\mathcal{N}_{\subset}(v)$. Each single-layer network of $\mathcal{N}_{\subseteq}(v)$ in layer L_i is the induced subgraph $G_i[V_i^{\subseteq(v)}]$ of G_i, where $V_i^{\subseteq(v)} \subset V_{G_i}$ is the vertex set composed by the nodes in V_{G_i} that are neighbors of v in at least one of the layers $\hat{L}$ (and not just in that specific layer L_i). More formally:

$$\mathcal{N}_{\subseteq}(v) = \{G_i[V_i^{\subseteq(v)}]\}_{i=1}^{d}\,, \tag{4.3}$$

where:

$$V_i^{\subseteq(v)} = (\bigcup_{h=1}^{d} V_h^{\subset(v)}) \cap V_{G_i}\,. \tag{4.4}$$

$\mathcal{N}_{\subseteq}(v)$ is a node-aligned network if the neighbors of v are present in all the layers $\hat{L}$. The neighborhood computed according to this definition is called *inclusive* since it can include nodes that are not immediate neighbors in some of the layers of the network (conversely to the previous definition). We can simplify the notation of the vertex set of $\mathcal{N}_{\subseteq}(v)$ in each layer L_i. We can say that $V_i^{\subseteq(v)}$ is composed by $V_i^{\subset(v)}$ and $V_i^{\cap(v)}$, which are other nodes that are neighbors of v in a layer different from L_i.

$\mathcal{N}_{\subset}(v)$ and $\mathcal{N}_{\subseteq}(v)$ do not include v (the *central node*) as a node. Both $\mathcal{N}_{\subset}(v)$ and $\mathcal{N}_{\subseteq}(v)$ are multiplex induced subgraph of their original network M. Moreover, $\mathcal{N}_{\subset}(v)$ can be defined as a *multiplex subgraph* of $\mathcal{N}_{\subseteq}(v)$. Indeed, the vertex set of $\mathcal{N}_{\subseteq}(v)$ in each layer includes both the immediate neighbors of v, that are always in the vertex set of the respective layer of $\mathcal{N}_{\subset}(v)$, plus other nodes (neighbors of v in other layers). An example of Non-Inclusive and Inclusive multiplex neighborhoods is shown in Figures 4.1 *a* and *b*.

A multiplex network M can be aggregated into a single-layer (or monoplex) network M_{agg}, that combine into a single-layer all the edges and nodes of M. Thus, we can define an ***aggregated neighborhood*** $\mathcal{N}_a(v)$ as the neighborhood of v in the network M_{agg}, resulted from the aggregation of the multiplex network M. Being $\mathcal{N}_a(v)$ a neighborhood in a monoplex network, definition 2.2 is still valid for it.

4.1.2 Multiplex neighborhoods isomorphism

In this subsection, we compare the two neighborhood definition we presented in the previous section in terms of isomorphism classes count. This is relevant to our study because the number of possible isomorphism classes

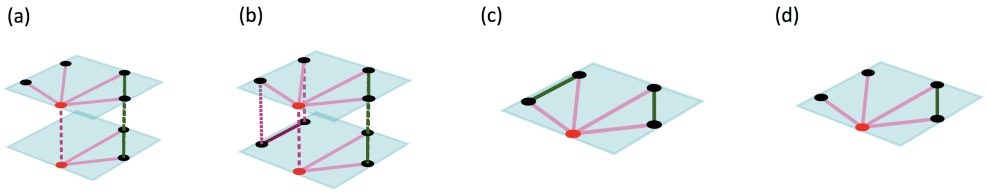

Figure 4.1: An example of Non-Inclusive and Inclusive multiplex neighborhood of the same node. The red node is the central node and the pink edges connect it to its neighbors. Figure *a* represent the Non-Inclusive multiplex neighborhood, and *b* the Inclusive multiplex neighborhood; *c* shows the aggregated neighborhood computed on the full aggregated network (corresponding to the aggregation of the Inclusive multiplex neighborhood in *b*); *d* shows the aggregation of the Non-Inclusive neighborhood represented in *a*.

that can be obtained with one or another definition affects the number of unique neighborhoods in a network.

As we have seen in the previous section, the vertex set $V_i^{\subseteq(v)}$ of each graph in the layers of $\mathcal{N}_{\subseteq}(v)$ includes both the nodes in the vertex set $V_i^{\subset(v)}$ of $\mathcal{N}_{\subset}(v)$, plus possible other node-layer tuples ($V_i^{\cap(v)}$) and the edges between them. For this reason, computing the neighborhood of the same node v according to both definitions can lead to different results. However, since definition $\mathcal{N}_{\subset}$ is less inclusive, there is a higher chance that more neighborhoods would be mapped to the same isomorphism class. For instance, in a network, there can be two neighborhoods X and Y, where X is equal to the Inclusive neighborhood in Figure 4.1 *b*, while Y has the same nodes as X, but without the edge between the two nodes that are not neighbors of the central node. In this case, X and Y are not isomorphic with neighborhood definition $\mathcal{N}_{\subseteq}$, but are isomorphic with neighborhood definition $\mathcal{N}_{\subset}$. The respective $\mathcal{N}_{\subset}$ would look like the one in Figure 4.1 *a*.

As discussed in Section 2.4, two multiplex networks M and M' are isomorphic if there exists a vertex map γ such that $V_M^{\gamma} = V_{M'}$. This applies also to multiplex neighborhoods, since they are multiplex networks themselves. However, we can also say that, if two multiplex networks are isomorphic, according to at least node-isomorphism, an existing vertex map relabels the nodes of one network to the one of another for each layer of the networks. If two Inclusive multiplex neighborhoods $\mathcal{N}_{\subseteq}(v)$ and $\mathcal{N}_{\subseteq}(v')$ of two nodes v and v' are isomorphic, there exists a vertex map γ such that,

for each layer L_i:

$$V_{i\,\mathcal{N}_{\subseteq}(v)}^{\subseteq(v)\,\gamma} = V_{i\,\mathcal{N}_{\subseteq}(v')}^{\subseteq(v)}\,. \tag{4.5}$$

Since in Inclusive Multiplex Neighborhoods, for each layer L_i, there can be two different type of nodes, $V_i^{\subset(v)}$ and $V_i^{\cap(v)}$, we can say that:

$$\gamma(v) = \xi(v) \;\; if \;\; v \in V_{i\,\mathcal{N}_{\subseteq}(v)}^{\subset(v)}\,, \tag{4.6}$$

$$\gamma(v) = \eta(v) \;\; if \;\; v \in V_{i\,\mathcal{N}_{\subseteq}(v)}^{\cap(v)}\,, \tag{4.7}$$

and consequently:

$$V_{i\,\mathcal{N}_{\subseteq}(v)}^{\subset(v)\,\xi} = V_{i\,\mathcal{N}_{\subseteq}(v')}^{\subset(v)}\,, \tag{4.8}$$

$$V_{i\,\mathcal{N}_{\subseteq}(v)}^{\cap(v)\,\eta} = V_{i\,\mathcal{N}_{\subseteq}(v')}^{\cap(v)}\,. \tag{4.9}$$

If only the vertex map ξ exists, then the two considered neighborhoods are isomorphic only if we consider their Non-Inclusive version $\mathcal{N}_{\subset}(v)$ and $\mathcal{N}_{\subset}(v')$. If two neighborhoods are isomorphic according to the Inclusive definition, they are also isomorphic according to the Non-Inclusive one. In fact, if a vertex map γ exists, then also ξ exists. Moreover, for each layer L_i:

$$E_i(\mathcal{N}_{\subseteq}(v))^{\gamma} = E_i(\mathcal{N}_{\subseteq}(v')) \implies E_i(\mathcal{N}_{\subset}(v))^{\xi} = E_i(\mathcal{N}_{\subset}(v')\,. \tag{4.10}$$

Since:

$$E_i(\mathcal{N}_{\subset}(v))^{\xi} = \{(x^{\xi}, y^{\xi}) | (x,y) \in E_i(\mathcal{N}_{\subset}(v))\}\,, \tag{4.11}$$

and since any pair of vertices x, y in one layer L_i of $\mathcal{N}_{\subset}(v)$ belongs also to $\mathcal{N}_{\subseteq}(v)$, then all the edges between them would belong also to $\mathcal{N}_{\subset}(v)$. This is because $\mathcal{N}_{\subset}(v)$ is a multiplex subgraph of $\mathcal{N}_{\subseteq}(v)$, and $V_i^{\subset(v)} \subset V_i^{\subseteq(v)}$. In one layer L_i, there might be an edge between a node $z \in V_i^{\subset(v)}$ that is an immediate neighbor of v and one $h \in V_i^{\cap(v)}$ that is not. In this case, this edge (z,h) would be included in $E_i(\mathcal{N}_{\subseteq}(v))$, but not in $E_i(\mathcal{N}_{\subset}(v))$, since $h \notin V_i^{\subset(v)}$. In fact, for each layer L_i of $\mathcal{N}_{\subseteq}(v)$, the sets $V_i^{\subset(v)}$ and $V_i^{\cap(v)}$ are disjoint. What has been just said can be also extended to node-layer isomorphism. Indeed, two neighborhoods that are node-layer isomorphic considering the Inclusive definition, then are also node-layer isomorphic considering the Non-Inclusive definition.

The number of possible isomorphism classes (both for vertex-isomorphism and vertex-layer isomorphism) that can be obtained by computing the neighborhoods of a network using definition $\mathcal{N}_{\subseteq}$ is always higher or equal than the one obtained using definition $\mathcal{N}_{\subset}$. Indeed, the number of possible isomorphism classes grows with the number of nodes, in simple and in

multiplex network (as it is shown in [KP18]), and $\mathcal{N}_{\subseteq}$ has in general more nodes than $\mathcal{N}_{\subset}$.

We can also compare $\mathcal{N}_{\subset}(v)$ and $\mathcal{N}_{\subseteq}(v)$ with the aggregated neighborhood $\mathcal{N}_a(v)$. $\mathcal{N}_a(v)$ contains all the edges that exist in every layer between the nodes that are neighbor of v in at least one layer. Indeed, by aggregating $\mathcal{N}_{\subseteq}(v)$ we can always obtain $\mathcal{N}_a(v)$. Conversely, when aggregating $\mathcal{N}_{\subset}(v)$, some of the edges observable with either $\mathcal{N}_a(v)$ or $\mathcal{N}_{\subseteq}(v)$ may not be included. An example of this case is shown in Figure 4.1: in Figure 4.1 *b* we can observe an additional edge in $\mathcal{N}_{\subseteq}(v)$ compared to $\mathcal{N}_{\subset}(v)$; the same edge can be observed in the aggregated neighborhood in Figure 4.1 *c*; aggregating $\mathcal{N}_{\subset}(v)$ would not allow to observe the additional edge present in $\mathcal{N}_{\subseteq}(v)$, as can be seen from Figure 4.1 *d*.

$\mathcal{N}_{\subseteq}(v)$ always contains at least the same amount of information of $\mathcal{N}_a$, since the number of nodes and edges is the same in both neighborhoods, and, additionally, $\mathcal{N}_{\subseteq}(v)$ contains the information on which layers the nodes and links are located. Thus, we can state that the number of possible isomorphism classes according to both node and node-layer isomorphism that can be obtained using definition $\mathcal{N}_{\subseteq}(v)$ is higher or equal than the ones obtained with $\mathcal{N}_a(v)$[1]. On the other hand, it is hard to compare the number of isomorphism classes that can be obtained using $\mathcal{N}_{\subset}$ and $\mathcal{N}_a$, since $\mathcal{N}_{\subset}$ contains information about the location of some of the nodes (the immediate neighbors of the central node), but may not contain information about the location of the nodes that are not immediate neighbors and the edges between them. We discuss how this result is related to the uniqueness of neighborhoods in Section 4.1.4.

4.1.3 Multiplex neighborhoods extraction

In the light of neighborhood attack, the choice of using definition $\mathcal{N}_{\subset}$ or $\mathcal{N}_{\subseteq}$ depends on the attacker's background knowledge: we can use $\mathcal{N}_{\subset}$ if the attacker knows just about the single neighborhoods of the target node in different social networks, while $\mathcal{N}_{\subseteq}$ implies that the attacker also knows about the existence of the neighbors and the links between those in all the considered social networks, even if they are not neighbors of the target in one of the networks. We can give a practical example with online social networks such as Facebook and Twitter, assuming v as the target node of a

[1]This statement is based on the fact that the number of multiplex isomorphism classes is higher than the ones of simple graphs, as shown in [KP18]

neighborhood attack:

Example: $\mathcal{N}_{\subset}(v)$ is built by extracting the network of v's friends on Facebook and of v's followers on Twitter separately, and then combine them in a multiplex network. If f_1 and f_2 are two friends of v on Facebook, and they also have a Twitter account, but they do not a follow v on Twitter, then f_1 and f_2 are not included in the Twitter layer of the neighborhood. Conversely, $\mathcal{N}_{\subseteq}(v)$ would include both f_1 and f_2 in the Twitter layer, as well as an edge between those in the case they follow each other.

In this thesis we mostly focus on definition $\mathcal{N}_{\subset}$, and we conduct all the experiments in the next Chapters 5, 6 and 7 according to it. This is probably a more realistic approach, since it is easier for the attacker to access the neighborhoods of a target node, while, in $\mathcal{N}_{\subseteq}$, the attacker also needs to know the connections of nodes that are not the primary target. However, the analysis in this document can be extended to $\mathcal{N}_{\subseteq}$ following the same methodology.

To extract neighborhoods $\mathcal{N}_{\subset}$ from a network, we first extract the single-layer neighborhoods in all the layers and then build the corresponding multiplex neighborhood. Conversely, to extract $\mathcal{N}_{\subseteq}$, we extract the induced subgraph formed by all the nodes that are neighbors of the central node in at least one layer. However, since we do not include the *central node* in the neighborhood's graph, we need to distinguish the two types of possible node-layer tuples: the ones that are in the neighborhood of the central node, and the ones that are not. Indeed, to compute the fraction of unique neighborhoods in a network, it is necessary to map each neighborhood to an isomorphism class, for example through the computation of the complete invariant (a unique string corresponding to one and only one isomorphism class). If the two different node-layer types are not distinguished, then all the node-layer tuples would be interpreted as actual neighbors, potentially leading to the mapping of two neighborhoods that are different to the same isomorphism classes.

4.1.4 Multiplex neighborhoods uniqueness

We want to study the uniqueness of neighborhoods in a network. A neighborhood in a network M is unique if the number of times that the corresponding isomorphism class (which in this case we call *neighborhood isomorphism class*) occurs in M is exactly 1. Before introducing the definition for the uniqueness of neighborhoods, we need first to define the *occurrence frequency*

$O_{\mathcal{N}_\tau(v)}$ of a neighborhood $\mathcal{N}_\tau(v)$ (τ depends on the type of the neighborhood, which can be multiplex inclusive, non-inclusive or aggregated) in a network M (monoplex or multiplex), with vertex set V. We define $O_{\mathcal{N}_\tau(v)}$ as the number of neighborhoods in M that are isomorphic to $\mathcal{N}_\tau(v)$:

$$O_{\mathcal{N}_\tau(v)} = \sum_{v' \in V} \delta(\mathcal{N}_\tau(v) \cong \mathcal{N}_\tau(v'))\,, \tag{4.12}$$

where

$$\delta(\mathcal{N}_\tau(v) \cong \mathcal{N}_\tau(v')) = \begin{cases} 1, & if\ \mathcal{N}_\tau(v) \cong \mathcal{N}_\tau(v') \\ 0, & otherwise \end{cases}\,. \tag{4.13}$$

We can then define the *uniqueness of neighborhoods* $U_{\mathcal{N}_\tau}$ (or, simply, *uniqueness*) of a network M with n nodes as following:

$$U_{\mathcal{N}_\tau} = \sum_{v \in V} \frac{\delta(O_{\mathcal{N}_\tau(v)} = 1)}{n}\,, \tag{4.14}$$

where:

$$\delta(O_{\mathcal{N}_\tau(v)} = 1) = \begin{cases} 1, & if\ O_{\mathcal{N}_\tau(v)} = 1 \\ 0, & otherwise \end{cases}\,. \tag{4.15}$$

Thus, the uniqueness of neighborhoods corresponds to the fraction of neighborhood structures that occur one and only one time in the network. If the value of uniqueness is equal to one (maximum uniqueness), it means that there are only unique neighborhoods in the network M, thus no neighborhood is isomorphic to any other. Conversely, if $U_{\mathcal{N}_\tau} = 0$ (minimum uniqueness), every neighborhood occurs at least two times in M, and if $U_{\mathcal{N}_\tau} = 0.5$, half of the neighborhoods occur just one time in M.

To simplify the notation, depending on the neighborhood type τ (multiplex inclusive, non-inclusive, or aggregated), the graph type $\mathcal{G}$, the isomorphism type $\mathcal{I}$, and the aggregation level $\mathcal{A}$, we can consider the uniqueness as a function of a quadruplet:

$$U_{\mathcal{N}_\tau} = U(\mathcal{G}, \mathcal{N}_\tau, \mathcal{I}, \mathcal{A})\,, \tag{4.16}$$

where $\mathcal{G}$ is either a multiplex network M or a monoplex network G; $\mathcal{I}$ can be either node (indicated with [0]) or node-layer isomorphism (indicated with $[0, 1]$); $\mathcal{A}$ can indicate an aggregation, assuming value 1, or not, assuming value 0. We can define simplified notation for every case of uniqueness we consider:

- Aggregated (or monoplex) uniqueness ($U^{(a)}_{\mathcal{N}}$):

$$U^{(a)}_{\mathcal{N}} = U(G, \mathcal{N}_a, [0], 1)\,, \tag{4.17}$$

([0] is indicated as the isomorphism type since in a monoplex network there are no layers, thus only nodes can be considered for isomorphism). The aggregated (or monoplex) uniqueness is the uniqueness of neighborhoods in the aggregated network, or, in general, in a monoplex network. Indeed, an aggregated network, in our case, is always a monoplex network:

- Multiplex (node or node-layer) inclusive uniqueness ($U_{\subseteq[\mathcal{I}]}$):

$$U_{\subseteq[\mathcal{I}]} = U(M, \mathcal{N}_{\subseteq}, \mathcal{I}, 0)\,, \tag{4.18}$$

  where $\mathcal{I}$ is either $[0]$ or $[0, 1]$ depending on the isomorphism type used. the multiplex (node or node-layer) inclusive uniqueness is the uniqueness of neighborhoods in multiplex networks, where the neighborhoods are defined as inclusive multiplex neighborhoods $\mathcal{N}_{\subset}$, according to the definition in Section 4.1 (node or node-layer depend on the isomorphism type used).

- Multiplex (node or node-layer) non-inclusive uniqueness ($U_{\subset[\mathcal{I}]}$):

$$U_{\subset[\mathcal{I}]} = U(M, \mathcal{N}_{\subset}, \mathcal{I}, 0)\,, \tag{4.19}$$

  where $\mathcal{I}$ is either $[0]$ or $[0, 1]$ depending on the isomorphism type used. The multiplex (node or node-layer) non-inclusive uniqueness is the uniqueness of neighborhoods in multiplex networks, where the neighborhoods are defined as non-inclusive multiplex neighborhoods $\mathcal{N}_{\subset}$, according to definition in the previous Section 4.1 (node or node-layer depend on the isomorphism type used).

Since the number of possible isomorphism classes with a given number of nodes is strongly related with the uniqueness of neighborhoods (in fact, higher the number of possible isomorphism classes, higher the number of possible neighborhoods, higher the uniqueness), based on the conclusion from Section 4.1, we can say that with definition $\mathcal{N}_{\subseteq}$, the uniqueness is always higher (or equal) than in the aggregated network, while this is not always true if we use definition $\mathcal{N}_{\subset}$. Hence, the uniqueness $U_{\subseteq[\mathcal{I}]}$ of a multiplex neighborhood (or, in general, of a multiplex network) computed according to definition $\mathcal{N}_{\subseteq}$ respects the following inequality:

$$U_{\mathcal{N}}^{(a)} \leq U_{\subseteq[\mathcal{I}]}\,, \tag{4.20}$$

where $U_{\mathcal{N}}^{(a)}$ is the uniqueness in the aggregated network M_{agg}, as defined above.

We also define the *degree uniqueness*, as the fraction of nodes with a unique degree in a network. We call those nodes *unique degree nodes*. In multiplex networks, unique degree nodes are the ones where the combination given by the degrees in all the layers is unique, where the degree in a layer is the degree of the monoplex networks in that layer (thus by counting just the intra-layer edges). However, this definition is valid just for node-isomorphism, while in node-layer isomorphism the layers are interchangeable, thus also the degrees in those so are. In Chapter 5, we discuss and study the degree uniqueness, especially for Erdős-Rényi networks. Studying the degree uniqueness is important since there are methods that anonymize a network with respect to the nodes' degree (even in multiplex network, for instance [RMT15]), as we have seen in Chapter 3. The uniqueness of degree (or, in multiplex network, of degrees' combination) is relevant to our study since we can show that the neighborhood anonymization is, in general, a more difficult problem than the degree anonymization and we can analyze the cases in which degrees are able or not to represent the full neighborhoods in terms of uniqueness.

Based on the uniqueness definition as a quadruplet (Equation 4.16), indicating the degree (or degree combination) as k, and considering the degree as a special case of a neighborhood, we define the various types of degree uniqueness as following:

- aggregated (or monoplex) degree uniqueness ($U_k^{(a)}$): fraction of unique degree nodes in monoplex networks (or in aggregated multiplex networks). Defined as:
$$U_k^{(a)} = U(G, k, [0], 1)\,; \tag{4.21}$$

- multiplex node degree uniqueness ($U_{k\,[0]}^{(M)}$): fraction of unique degree nodes in multiplex networks, with respect to node isomorphism. Defined as:
$$U_{k\,[0]}^{(M)} = U(M, k, [0], 0)\,; \tag{4.22}$$

- multiplex node-layer degree uniqueness ($U_{k\,[0,1]}^{(M)}$): fraction of unique degree nodes in multiplex networks, with respect to node-layer isomorphism. Defined as:
$$U_{k\,[0,1]}^{(M)} = U(M, k, [0, 1], 0)\,; \tag{4.23}$$

In Chapter 5 we study the uniqueness of Non-Inclusive neighborhoods $U_{\subset[\mathcal{I}]}$ in three network models, by analyzing how the network structure influences it with different model parameters. If the uniqueness is higher in a

network with certain parameters value compared to others, it means that the number of possible neighborhoods (or neighborhood isomorphism classes) under those specific conditions is greater. Given the difficulty of computing the number of possible neighborhood isomorphism classes $|I|$ (especially in multiplex networks), the simulations can also help us to understand the variation of $|I|$ in different settings. Another interesting insight that can be obtained from this study is the understanding of the conditions under which the uniqueness of neighborhoods of $\mathcal{N}_{\subset}$ is higher or lower than the uniqueness of aggregated neighborhoods $\mathcal{N}_a$.

Since we are going to present simulations in which networks are generated from models, there exists more than one possible graph with the same features (network size, average degree, edge overlap in the case of multiplex networks), and all the possible graphs define a probability distribution. Based on that, we can define the expected uniqueness value of a network $\mathcal{H}$ (monoplex or multiplex) as:

$$\langle U_{\mathcal{N}_\tau}^{(\mathcal{H})} \rangle = \sum_{G_i \in \mathcal{H}} \langle U_{\mathcal{N}_\tau}^{(G_i)} \rangle \cdot P(G_i)\,, \tag{4.24}$$

where the sum is over all the possible graphs G_i, $P(G_i)$ is the probability of the graph G_i to occur, and $U_{\mathcal{N}_\tau}^{(G_i)}$ is the uniqueness value in that particular graph.

4.1.5 Multiplex networks' isomorphism classes count

Multiplex networks are systems that can be decomposed in different subsystems to be studied, mainly separating the edges that are overlapping across the layers (thus shared between the layers), and the ones that are not. This decomposition analysis can give us an interesting perspective on the count of possible isomorphism classes in a multiplex network, which is normally hard to compute).

For our analysis, we consider networks with just two layers, thus the overlapping edges are shared among all layers. In general, we can think of a multiplex network (or, equivalently, a multiplex neighborhood) as a system composed of two different systems:

- M_{ind}: a multiplex network which contains all the edges that are not overlapping across different layers (thus each layer is independent of any other);
- M_{ov}: a single-layer network which contains just the overlapping edges.

With this in mind, we can say that the number of isomorphism classes $|I_M|$, hence of possible neighborhoods, of a generic multiplex network M with n nodes, is given by three factors: the number of isomorphism classes of M_{ind} ($|I_{M_{ind}}|$, which is the number of possible multiplex isomorphism classes with the number of nodes of M_{ind}); the number of isomorphism classes of M_{ov} ($|I_{M_{ov}}|$); the effect due to the interaction between these two systems. If the fraction of overlapping edges $ov_E$is zero or one, such as there are no overlapping edges or all the edges are overlapping, we can write:

$$|I_M| = \begin{cases} |I_{M_{ind}}|, & if\ ov_E = 0\,, \\ |I_{M_{ov}}|, & if\ ov_E = 1\,. \end{cases} \tag{4.25}$$

From the above equation, if there is complete overlap ($ov_E = 1$), the number of isomorphism classes does not grow with the number of layers, and the problem of counting multiplex isomorphism classes is reduced to the count of single-layer ones. When there is no overlap ($ov_E = 0$), the computation is simple just in the case the layers do not share any node, and it would be:

$$|I_{M_{ind}}| = \prod_{i=1}^{d} |I_{M_{L_i}}|\,, \tag{4.26}$$

where i indicates the index of the layers, and d is the higher layer's index. The reason for the presence of the product is that one needs to compute all the possible combination of graphs across all the layers. In the case of a multiplex network with only two layers, the above Equation 4.26 becomes:

$$|I_{M_{ind}}| = |I_{M_{L_1}}| \times |I_{M_{L_2}}|\,, \tag{4.27}$$

where $|I_{M_{L_1}}|$ and $|I_{M_{L_2}}|$ are the number of isomorphism classes of the graphs in layer L_1 and L_2, respectively. If the layers of the network share some nodes, then all the isomorphism classes cannot be computed in this way, since, in the count, there would be included also the ones with overlapping edges.

4.2 System and attacker model

We consider the problem of identity disclosure in sharing social networks data, such as a list of nodes, representing entities, and edges, representing relationships between a pair of nodes. In particular, we take into account neighborhood attacks on undirected and unlabelled multiplex networks with

one aspect. We call this kind of attack *multiplex neighborhood attack*. We focus on the re-identification of a node in data that are claimed to be "anonymized" by dropping the nodes' attributes (this is what is usually called naïve anonymization).

The attacker's background knowledge consists of the neighborhood structure of one or more particular nodes, in various social networks. We model the ensemble of these social networks as a unique system, such as a multiplex network (with a single aspect), where each layer represents one social network. In our case, the attacker knows the complete 1-hop neighborhood of a node (consisting in the neighboring nodes, plus the edges among them), across all the layers of the network. We focus on multiplex neighborhoods of type $\mathcal{N}_{\subset}$, presented in the previous Section 4.1. Thus, the attacker knows the neighborhood of the target in every single social network and, after that, it combines this knowledge to a multiplex network by connecting the same nodes with intra-layer edges. The layers of the multiplex network can either represent networks with different types of relationships (for example, Facebook friendships, phone calls or text messages) or the same kind of relationships at different time-stamps (in this case, we talk about a temporal network, and the layers have a particular order given by the time, that cannot be modified).

Some neighborhood structures can be unique (in the case there is no other isomorphic neighborhood structure) already in a single-layer network (composed by just one social network, but not as a result of the aggregation of multiple layers), while others can belong to the same isomorphism class. Anonymizing the network with methods similar to the ones presented in Chapter 3 can be harder if the attacker is equipped with the knowledge of various single-layer networks. The attacker can indeed combine the information coming from different layers to make the re-identification easier. Intuitively, more layers are shared, easier the attack would be (given that the attacker knows the node's neighborhood structure in all of them), and harder the anonymization is (both because of the presence of more unique structure, and from a computational point of view). However, as explained in the previous in subsection 4.1.4, there are cases in which sharing of the aggregated network (a monoplex network with all the links existing between nodes in all the single-layer networks considered) could be worse than sharing the multiplex network, since we can lose information about where the edges are actually located (or, in other words, which social network they belong to).

A variant of the multiplex neighborhood attack is the one where the attacker knows the whole 1-hop neighborhood in some of the layers, while he/she has a partial knowledge of the neighborhood in others. As an

example, with a three-layer multiplex network, the attacker could possess the knowledge of a neighborhood across two layers, while knowing just the degree of the target nodes in a third layer. This case is the same as knowing the entire neighborhood in all the three layers, if, in the third one, there are no edges among the neighbors of the target nodes. However, in our analysis, we do not take this option into account.

We assume that the attacker has access to the system composed by different social networks represented already as a multiplex network. In reality, the adversary could have access to different datasets separately, and could either perform the neighborhood attack separately on each monoplex network, and then combine the findings to identify its target (the neighborhood that matches its knowledge in all the graphs are the possible targets), or construct a multiplex network from the data and perform then the multiplex neighborhood attack.

Technically, to perform the multiplex neighborhood attack, the attacker needs to match the known neighborhood graph with one (or more) of the neighborhoods in the original system. Thus, it is needed to perform an isomorphism test against all the 1-hop neighborhoods of the dataset, until (s)he finds all the matching subgraphs. If the target node's neighborhood is unique, then the attack is successful; otherwise, the more isomorphic subgraphs are present in the network, the lower the success probabilities are. In fact, in this case, the adversary either has to guess which neighborhood is the right one or needs to exploit additional information to identify the victim.

The attacker can also possess or not the information regarding the layer's labels (for example, because the data are shared with or without them). In this case, the re-identification problem becomes more difficult: being the layer interchangeable, less unique neighborhoods would be present. This scenario is not applicable to temporal networks, where the layers' order (such as "timestamp 1", "timestamp 2", "timestamp 3"...) is crucial. However, if the temporal network has, for instance, more than two timestamps, and the attacker knows the neighborhood computed from just two consecutive timestamps, but not exactly which ones, then the attacker would need to conduct multiple isomorphism tests with all the possible consecutive layers combinations.

The types of isomorphism that the attacker should take into account are the following:

- *vertex isomorphism* in the case of known layer's labels (and with temporal networks);
- *vertex-layer isomorphism* in the case of unknown layer's labels.

These types of isomorphism have been introduced in Section 2.4. If the networks are shared by aggregating the layers in a single one, then also the attacker should have (or should build) an aggregated single-layer network from his/her background knowledge and, in this case, the type of isomorphism used in the isomorphism tests is the classical graph isomorphism, defined in Section 2.3. However, in this case, the attack may fail since the aggregation of a multiplex neighborhood of type $\mathcal{N}_{\subset}$ can lead to a different aggregated neighborhood than the one actually present in the aggregated graph.

Chapter 5

Uniqueness of neighborhoods in random networks

In this chapter, we study the uniqueness of neighborhoods in three random network models, to understand how different network structures influence the formation of unique neighborhoods, and, consequently, the difficulty of the anonymization problem. Indeed, it can be more difficult to anonymize a network if the number of unique neighborhoods is higher. Our aim is to gain a deeper understanding of the problem that anonymization methods try to solve. A better overview of the situation can guide the creation of an apposite algorithm for anonymization, especially in the case of multiplex networks, where the complexity of the system is higher than in simple networks.

As mentioned in Section 3.1, previous studies [Hay+07; ZP11] assessed the uniqueness of neighborhoods in empirical data, without performing an analysis of the factor which this feature depends on. Another paper [Hay+08] studied the uniqueness of nodes, based on their neighbors' degree, at various hops, only in Erdős-Rényi networks.

We analyze and discuss that the proportion of unique neighborhoods' structures increases with the average degree and decreases with the network size. In fact, the bigger the network is, the more chances of having isomorphic classes are present.

With the definition and hypothesis specified in Chapter 4, we conduct simulations with three network models, Erdős-Rényi, Watts-Strogatz, and Random Geometric Graphs and their multiplex versions, varying the network size, average degree, and, in the multiplex case, the edge overlap proportion between different layers. Our aim is to understand how different structures behave in terms of anonymization difficulty. In particular, we consider the data sharing of a multiplex network in three different settings: the layers'

labels are shared; the layers' labels are not shared; the layers are aggregated in a single-layer. These three settings correspond to different attackers' knowledge, and, consequently, they lead to different attacks, mainly because of distinct isomorphism tests that the attacker has to conduct while searching for the target node.

We also present some formulas to determine the proportion of unique degree nodes and triangles in Erdős-Rényi networks. Those equations are useful in our study since unique degree nodes can be enough to explain the uniqueness of neighborhoods when the average degree is relatively small, and, consequently, there are no edges between the neighbors of a node.

5.1 Simulation method

As stated above, the aim of this chapter is to get an idea on how network features affect the uniqueness in three network models, Erdős-Rényi (ER), Watts-Strogatz (WS) and Random Geometric Graph (RGG), both in their monoplex and multiplex versions.

In the next sections, we explain how we build the multiplex version of the three considered network models, Erdős-Rényi, Watts-Strogatz, and Random Geometric Graph, and illustrate the results related to those. We first present in detail the process of building each model with the desired features, then show how the uniqueness in those models varies with size, average degree and proportion of edge overlap, which is the proportion of edges that are shared, i.e. overlapped, between two different layers. We have generated networks from different models from sizes from 100 to 10000, and for a range of average degrees from 0.1 until 90, computing the amount of of nodes with unique neighborhoods and, for the Erdős-Rényi model, nodes with unique degree (regarding which we also present some equations). Nodes with unique degree correspond to the ones with unique neighborhoods if there are no triangles in neighborhoods (i.e. if the local clustering coefficient is equal to zero). For each combination of parameters we have generated 5 networks, computed the uniqueness in each of those and then taken the mean and variance of the obtained uniqueness values. The plots in this section are based on the computed mean. The variance of each of the results is very low and, when the standard deviation is plotted as an error bar, it is almost not visible). Therefore, it is not taken into account since negligible.

We are going to show how the uniqueness varies with network size and average degree, to understand how long it takes for a system to pass from uniqueness 0 to uniqueness 1, assuming a growth process either in the size or in the amount of edges (and consequently in the average degree). Moreover,

we want to understand what are the edge overlap values that lead to greater or lower values of uniqueness. After presenting some results of complete simulations for each of the model, we compare them better in Section 5.6. In that section, we run simulations to specifically find a certain uniqueness value, in particular 0.5, such as when half of the nodes in the network have unique neighborhoods, with a modification of the binary search algorithm.

For simplicity, we consider multiplex network models with just two layers, fully interconnected, and with the same amount of edges and average degree in each layer. Each model has three main parameters: network size n, expected average degree $\langle k \rangle$ and edge overlap ov_E. The considered value of average degree is the average degree of the aggregated network, that is the average degree that the network would have if aggregated into a single-layer. Since we conduct simulations with network models, we always refer to the uniqueness values as expected uniqueness (e.g. $\langle U_{\mathcal{N}_\tau} \rangle$).

5.2 Multiplex Network models

To study the uniqueness in various settings, we want to build multiplex networks that have different factors of correlations between the layers. Indeed, in real-world data, two different networks with the same nodes could be strongly correlated to each other, sharing a high amount of edges, and this correlation would be reflected by a high value of edge overlap. Contrarily, two network could also be almost completely uncorrelated, with a low value of edge overlap). For example, if two layers represent different contexts and the nodes represent people, a high value of edge overlap would mean that people interact in the same way and with the same other entities across all the contexts. Conversely, a low value of edge overlap would reflect an almost independent behaviour of the represented entities across the considered contexts.

Exploiting the possibility of seeing a multiplex network as a system composed by one system with two layers with independent (i.e. not overlapping) edges, and a second system with only one layer that contains the overlapping edges, called the *overlapping layer*), we can construct a multiplex network with a desired edge overlap value starting from monoplex networks. In fact, we can always build a multiplex network with two layers starting from the independent generation of each of the two layers, which, for large-enough networks, will not have any overlapping edge. We can then generate an additional layer with the desired amount of overlapping edges, which will then be copied to both of the other two layers.

The average degree $\langle k_a \rangle$ of the aggregated network is given by the sum of the average degree of the independent layers (two times, one for each layer) plus the average degree of the overlapping layer. Every multiplex network with different edge overlap values but with the same average degree would then result in the same single-layer network if aggregated.

The expected number of edges $\langle m_i \rangle$ in each layer is computed by:

$$\langle m_i \rangle = \frac{\langle k_a \rangle \, n}{2} \times \frac{1}{2} \, . \tag{5.1}$$

This is obtained by reversing Equation 2.1 of the computation of the average degree divided by two, since in each layer the average degree is half of the average degree in the aggregated network.

We can also compute the average degree and the number of edges in each of the system that form a multiplex network M: M_{ind}, the system with two independent layers, and M_{ov}, the system with a single layer containing the overlapping edges.

In the overlapping layer, the average degree $\langle k_{ov} \rangle$ and the number of edges m_{ov} are:

$$\langle k_{ov} \rangle \; = \; \langle k_a \rangle \; \times \; ov_E \, , \tag{5.2}$$

$$m_{ov} = \frac{\langle k_{ov} \rangle \, n}{2} \, . \tag{5.3}$$

While in each of the independent layers, the average degree $\langle k_{ind} \rangle$ and the number of edges m_{ind} are:

$$\langle k_{ind} \rangle \; = \; \frac{\langle k_a \rangle \; - \; \langle k_{ov} \rangle}{2} = \frac{\langle k_a \rangle \; - \; (\langle k_a \rangle \times ov_E)}{2} \, , \tag{5.4}$$

$$m_{ind} = \frac{\langle k_{ind} \rangle \, n}{2} \, , \tag{5.5}$$

Thus the expected number of edges in each of the resulting layers is

$$\langle m_i \rangle = m_{ind} + m_{ov} \, . \tag{5.6}$$

Consequently, also the average degree in each layer is: Thus the expected number of edges in each of the resulting layers is :

$$\langle k_i \rangle = k_{ind} + k_{ov} \, . \tag{5.7}$$

5.3 Erdős-Rényi model

In this section, we present the simulation results we obtained with the multiplex version of the Erdős-Rényi model. Moreover, we present formulas to determine the fraction of unique degree nodes in monoplex and multiplex Erdős-Rényi networks. The nodes with unique degree are uniquely identifiable in a network. For this reason, it is important to understand whether the degree is enough to describe the uniqueness (and, consequently, the re-identifiability) of the nodes, or more complete structure such as neighborhoods are needed. We also present the formula to determine the fraction of neighborhoods with at least a triangle in a Erdős-Rényi network. In this way, we can understand the gap between the fraction of nodes with unique degree and the ones with unique neighborhoods. Indeed, the presence of triangles causes the diversity of neighborhoods even with the same degree.

5.3.1 Multiplex Erdős-Rényi model

The Erdős-Rényi network is by nature a random network and, because of the way a network is generated from this model, two different large sparse ER networks with the same amount of nodes would share almost no edges. Thus, if we build a multiplex ER network in which the monoplex networks in each layer are generated according to the ER model, we would obtain no overlapping edges if the network size is large enough and the average degree small enough to have a sparse network. However, we want to obtain multiplex networks that have some correlation across the layers, represented by a certain amount of overlapping edges. Moreover, we would like to control the amount of correlation between the layers, hence the amount of overlapping edges. For this reason, we generate a multiplex network with a given value of edge overlap that, aggregated, would give an ER network with the desired number of nodes and average degree.

The multiplex version of the Erdős-Rényi model is built in various steps. First, given the number of nodes, the average degree and the proportion of overlapping edges in each pair of layers, we compute the number of edges needed in each layer, according to Equation 5.1. Once we have the number of wanted edges in each layer, we randomly add edges to the two layers separately, without overlapping them. Finally, we generate another monoplex ER graph with a number of edges (that are not already present in the other layers) given by the overlapping proportion, and we copy those edges to each of the layers. The amount of necessary edges in all the layers, and in the overlapping layer, can be obtained from the equations presented

in Section 5.2.

5.3.2 Unique degree nodes in monoplex Erdős-Rényi networks

In this section, we study and provide a formula for the fraction of unique degree nodes in a monoplex Erdős-Rényi networks ($U_k^{(a)}$). In Section 5.3.4, we extend the same calculations for multiplex ER networks. Unique degree nodes are important since if in a neighborhood of a node there are no edges between the neighbors, that neighborhood is entirely described by the degree of the central node, and the local clustering coefficient of the central node is equal to zero. If all the nodes in the network have local clustering coefficient equal to zero, then the nodes with unique degrees would also be the ones with unique neighborhoods. In other words, the uniqueness of neighborhoods is explained completely by nodes with unique degrees when the average local clustering coefficient of a network is equal to zero. When the average degree starts increasing, at some point, depending on the network size, the neighborhoods would also includes triangles. Triangles are formed by the edges between the neighbors, since there are always edges between the central node to its neighbors. When triangles are present, unique degree nodes would not be enough to describe neighborhoods' uniqueness. We provide formulas to compute the number of neighborhoods with at least one triangle in Section 5.3.3.

When triangles appear, the local clustering coefficient value increases, and, while unique degree nodes still have unique neighborhoods, they are not the only uniquely identifiable nodes. Indeed, nodes with the same degree can have different neighborhoods because of a different amount or disposition of triangles in the neighborhood (e.g. one node of degree 3 has two edges between its neighbors, while another with the same degree has just one edge between its neighbors). This behaviour of Erdős-Rényi networks can be seen in Figure 5.1, for network with 100 and 2500 nodes. In this figure we can see that triangles appear quite soon when the average degree starts increasing, and the unique degree's nodes fraction remains relatively low. Moreover, the actual uniqueness of neighborhoods corresponds to the degree uniqueness when the average degree is small ($\langle U_k^{(a)} \rangle = \langle U_{\mathcal{N}}^{(a)} \rangle$).

$\langle U_{\mathcal{N}}^{(a)} \rangle$ increases after triangles appear in neighborhoods, since, as mentioned before, the degree of nodes is not enough to describe the uniqueness of neighborhoods anymore. The point where the curves representing $\langle U_{\mathcal{N}}^{(a)} \rangle$ and $\langle U_k^{(a)} \rangle$ diverge corresponds to an average degree value which is higher as the

network is bigger (as can be noticed also from Figure 5.1). Indeed, in a big network, compared to a small one, there are more possibilities to have two nodes with the same average degree and also with the same neighborhood, and that is the reason why, generally, in bigger networks, there would be a smaller fraction of unique degree nodes and also of unique neighborhoods (in Figure 5.1, we can see that the values of $\langle U_k^{(a)} \rangle$ are higher in the network with 100 nodes compared to the one of 2500). Thus, the uniqueness of neighborhoods of relatively large networks in their sparse region can be totally explained by the degree uniqueness, and triangles become less and less determinant to identify unique neighborhoods as the network size grows (this can be noticed by the gap from the neighborhood with triangles and unique neighborhoods curves in Figure 5.1). However, we need triangles to identify unique neighborhoods when networks start becoming more dense.

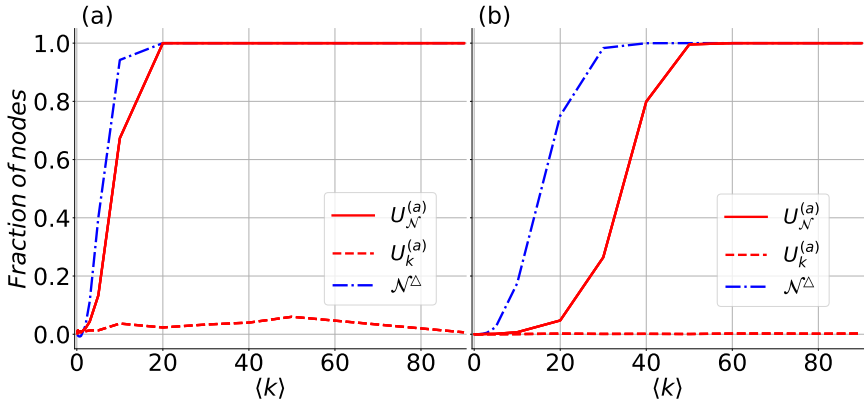

Figure 5.1: Expected uniqueness of neighborhoods (red line), degree uniqueness (red dashed line) and proportion of neighborhoods with at least a triangle (blue line) in ER networks of size 100 (a) and 2500 (b).

A complete overview of the trend of unique degree nodes' in a monoplex network until the dense region is shown in Figure 5.2. It is clear that when the network is dense, the unique degree nodes start to decrease, until the network becomes complete ($\langle k_a \rangle = n - 1$), and there are no unique degree nodes anymore since all the nodes are connected to each other. Obviously, when the network is complete, also the uniqueness of neighborhoods decreases since all the neighborhoods are the same (we discuss more about the uniqueness of neighborhoods in ER in Section 5.3.5). In the dense regime, the behaviour of multiplex network can be a bit different from the monoplex ones. Indeed, since we study multiplex network with an average degree of the corresponding aggregated network as a parameter,

the networks in each single-layer are not themselves complete when the $\langle k_a \rangle = n - 1$, implying the existence of different neighborhood isomorphism classes in the multiplex system. However, in this thesis, we mainly focus on multiplex networks in the sparse regime, since dense networks are rare in reality (and also require higher computational resources to be generated).

The expected proportion of unique degree nodes $\langle U_k^{(a)} \rangle$ in simple graphs can be determined by the following formula:

$$\langle U_k^{(a)} \rangle = \sum_{k=0}^{\infty} p_k (1 - p_k)^{n-1} \,, \tag{5.8}$$

where p_k is the probability of a node to have degree k, according to the degree distribution, according to the degree distribution formula (Equation 2.24):

$$\begin{aligned} p_k &= \binom{n-1}{k} p^k (1-p)^{n-1-k} \\ &= \frac{(n-1)!}{k!(n-1-k!)} \left(\frac{\langle k \rangle}{n-1} \right)^k \left(1 - \frac{\langle k \rangle}{n-1} \right)^{n-1-k} . \end{aligned} \tag{5.9}$$

The range of degrees to sum over in Equation 5.8 includes all the possible values, from zero to $n - 1$ (when the network is complete). Indeed, the probability to have nodes with degree equal or greater than the number of nodes is 0 in graphs without self-loops and multi-edges. Figure 5.2 shows, for a small network of size 50, that the curve obtained with Equation 5.8 corresponds to the one given by the simulation. From Figure 5.2 we can see that the curve has a convex shape, and the maximum fraction of nodes with unique degrees is reached when the average degree is half of the maximum one ($\frac{n-1}{2}$). This means that, if we take the first derivative of Equation 5.8, and we evaluate it at $k = \frac{n-1}{2}$, it would result being equal to zero:

$$\left. \frac{d \langle U_k^{(a)} \rangle}{dk} \right|_{\langle k \rangle = \frac{n-1}{2}} = 0 \,. \tag{5.10}$$

From the same Figure 5.2, it is clear that the maximum fraction of unique degree nodes in the network is 7%. The nodes with unique degrees would never be able to explain the uniqueness of neighborhoods $U_{\mathcal{N}}^{(a)}$ of an ER network if $U_{\mathcal{N}}^{(a)} = 1$. As an example, in a very small ER network with 4 nodes, with $\langle k \rangle = \frac{4-1}{2} = 1.5$, $\langle U_k^{(a)} \rangle$ is approximately equal to 0.25. The value of $\langle U_k^{(a)} \rangle$ will then decrease as the network size increases. The fraction of unique degree nodes will be able to explain the whole uniqueness of

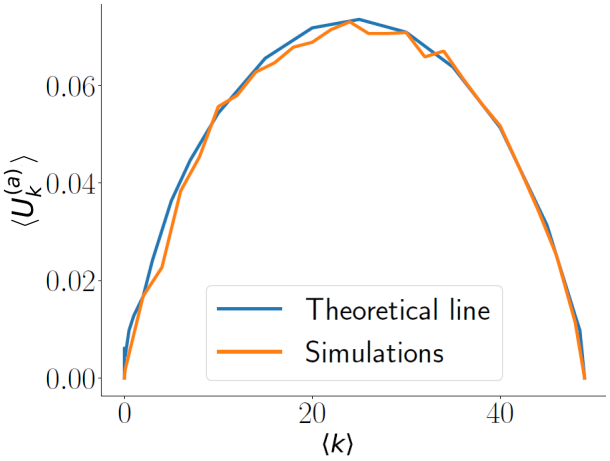

Figure 5.2: Fraction of unique degree nodes in ER monoplex network, with network size equal to 50, with different average degree values (horizontal axis). Both the theoretical line (in blue) according to Equation 5.8 and the curve derived by the simulations (in orange) are shown. The values from the simulation are computed as the mean of 100 network generation (the error bar corresponding to the standard error of the mean is not shown since the values are negligible).

neighborhoods just in sparse networks, when the local clustering coefficient is low, as shown in Figure 5.1.

5.3.3 Triangles in Erdős-Rényi networks

As mentioned above, when triangles appear in neighborhoods, the formula for unique degree nodes is not sufficient to explain neighborhoods' uniqueness. In this section, we show how to compute the expected fraction of neighborhoods with at least one triangle.

The probability to have at least one triangle in a neighborhood of a nodes with degree k is:

$$p_{\mathcal{N}^k_{a(\triangle)}} = 1 - (1-p)^{\binom{k}{2}} = 1 - (1-p)^{\frac{k(k-1)}{2}}, \tag{5.11}$$

where p is the parameter of the Erdős-Rényi model (the probability to add an edge between a pair of nodes).

From Equation 5.11 we can obtain the expected fraction of neighborhoods with at least one triangle $\mathcal{N}_{a(\triangle)}$ in an ER graph, by summing over all the

possible degrees and multiplying $p_{\mathcal{N}_k\triangle}$ by the degree distribution:

$$\langle \frac{\#\mathcal{N}_{a(\triangle)}}{n} \rangle = \sum_{k=0}^{\infty} (p_{\mathcal{N}_k\triangle} \times p_k)\,. \tag{5.12}$$

The trend of $\langle \frac{\#\mathcal{N}_{a(\triangle)}}{n} \rangle$ can be seen in Figure 5.1.

5.3.4 Unique degree nodes in multiplex Erdős-Rényi networks

In this Section, we extend the formulas to compute the fraction of unique degree nodes (provided in Section 5.3.2) to multiplex networks. We show how to compute multiplex degree uniqueness with respect to both node and node-layer isomorphism.

In multiplex ER network, unique degree nodes are the ones where the combination of the degree in all the layers is unique, as defined in Section 4.1.4. To obtain a valid formula for unique degree nodes in multiplex networks, in Equation 5.8, we need to substitute p_k with $p_{k_1k_2}$ (the probability of a node to have degree k_1 in layer L_1 and degree k_2 in layer L_2), if there are two layers, or $p_{k_1}p_{k_2}p_{k_3}$ if there are three layers, etc. For simplicity, here we present formulas for multiplex networks that are fully interconnected and with only two layers.

In addition to the notation presented in Section 4.1.4, we need to define the following notation, which will be useful to present the formulas for unique degree nodes in multiplex networks:

- $p_{k_1k_2[0]}$: probability of a node to have degree k_1 in layer L_1 and degree k_2 in layer L_2, with respect to node isomorphism;
- $p_{k_1k_2[0,1]}$: probability of a node to have degree k_1 in layer L_1 and degree k_2 in layer L_2, with respect to node-layer isomorphism;
- $p_{k_{ov}}$: probability of a node to have degree k in the overlapping layer.

We should also remember the decomposition of a multiplex network M into two systems M_{ind} (the multiplex network with two independent layers), and M_{ov} (the monoplex network containing the overlapping edges), defined in Section 4.1.4. This notation will be useful to study the degree of nodes, since the degree of a node in each of the layers is not always independent, because of the overlapping edges, which contribute to the actual degree of the nodes.

In the case of a network with two layers and node isomorphism, Equation 5.8 becomes:

$$\begin{aligned}\langle U^{(M)}_{k\,[0]}\rangle &= \sum_{k_1,k_2}^{\infty} p_{k_1k_2[0]}(1-p_{k_1k_2[0]})^{n-1} \\ &= \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} p_{k_1k_2[0]}(1-p_{k_1k_2[0]})^{n-1}\,,\end{aligned} \tag{5.13}$$

where the first sum is over all the possible combinations of k_1 and k_2. To compute $p_{k_1k_2[0]}$, we need to take into account the independent degrees of the two layers in the network M_{ind}, and the ones in the overlapping layer M_{ov}:

$$p_{k_1k_2[0]} = \sum_{k_{ov}=0}^{\infty} p_{k_{ov}}(p_{k_1-k_{ov}})(p_{k_2-k_{ov}})\,. \tag{5.14}$$

When the overlapping proportion is equal to zero, p_{k_1} and p_{k_2} are independent. Conversely, when the overlapping proportion is equal to 1, then the degrees in both layers are the same, and the probability of unique degree nodes is equal to the one in a single layer network:

$$p_{k_1k_2[0]} = \begin{cases} p_{k_1}p_{k_2}, & if\ ov = 0 \\ p_{k_{ov}}, & if\ ov = 1\,. \end{cases} \tag{5.15}$$

In node-layer isomorphism, the probability $p_{k_1k_2[0,1]}$ to have nodes with a given degree combination is the same as in node isomorphism just when $k_1 = k_2$. Otherwise, $p_{k_1k_2[0,1]}$ is equal to having either one of the possible combinations of degrees k_1 and k_2 in the two layers:

$$p_{k_1k_2[0,1]} = \begin{cases} p_{k_1k_2[0]}, & if\ k_1 = k_2 \\ p_{k_1k_2[0]} + p_{k_2k_1[0]}, & otherwise\,. \end{cases} \tag{5.16}$$

Moreover, in node-layer isomorphism, to obtain the proportion of nodes with unique degrees, it is not needed to sum over all the combination of k_1 and k_2 as in Equation 5.13, but just on the unique combinations not taking into account the possible permutations of degrees (for example, if we have already summed with $k_1 = 1$ and $k_2 = 2$, we do not need to sum again with $k_1 = 2$ and $k_2 = 1$):

$$\langle U^{(M)}_{k\,[0,1]}\rangle = \sum_{k_1\neq k_2}^{\infty} p_{k_1k_2[0,1]}(1-p_{k_1k_2[0,1]})^{n-1}\,. \tag{5.17}$$

The equations presented in this section are valid for networks that are large and sparse. The sparsity condition is needed since, when the network is dense, we may not be able to properly control the overlap since there might be edges overlapping by chance. In the following, we show that the presented equations are valid for large and sparse networks.

The number of edges m_{ov} in the overlapping layer M_{ov} is given by the total number of edges m_a, in the aggregated network, multiplied by the overlapping proportion ov_E, which can be seen as the parameter p of the ER network M_{ov}:

$$m_{ov} = p_{ov} \times m_a \,. \tag{5.18}$$

Since $m = \frac{\langle k \rangle \, n}{2}$ (obtained by solving the formula for the average degree in Equation 2.1 for m), and $p = \frac{\langle k \rangle}{n-1}$ (from Equation 2.23), we obtain:

$$m_{ov} = p_{ov} \times m_a = \frac{\langle k_a \rangle \times ov_E}{n-1} \times \frac{\langle k_a \rangle \, n}{2} \,. \tag{5.19}$$

With $n \to \infty$, we can cancel out n and $n-1$, and Equation 5.19 becomes:

$$m_{ov} = p_{ov} \times m_a \to \frac{\langle k_a \rangle^2}{2} \times ov_E \,, \tag{5.20}$$

which does not depend anymore on the network size. In summary, in large networks, the number of edges depends just on the expected average degree and not anymore on the network size. Thus, if the average degree has a value for which the network is sparse, the presented equations will work, since we are able to properly control the edge overlap value. When the average degree is high, instead, the number of edges is also high, and there could be edges that are overlapping by chance.

Figure 5.3 shows that the theoretical lines computed with Equations 5.13 and 5.17 are overlapping with the ones of networks generated with the ER multiplex model, with edge overlap values of 0, 0.5 and 1 and 100 nodes. With edge overlap equal to one, the unique degree nodes' proportion is the same with respect to both node and node-layer isomorphism, as well as in monoplex networks. The fraction of unique degree nodes with respect to node isomorphism is higher than the one with respect to node-layer isomorphism since, in this last case, the isomorphism is less strict (being the layers interchangeable), and two classes that are not node isomorphic can be node-layer isomorphic. The degree uniqueness in the multiplex case is significantly higher than in the monoplex one, and the maximum values are reached when the edge overlap $ov_E = 0.5$. This means that, at $ov_E = 0.5$, there is more variety of degree combinations across the layers. Moreover,

the degree uniqueness in multiplex networks is higher than in monoplex ones (or, equally, in multiplex networks with $ov_E = 1$), thus, in the multiplex case, there is a greater number of nodes uniquely identifiable knowing just the combination of their degree in all the layers.

Figure 5.5 shows the trend of the degree uniqueness and neighborhoods' uniqueness in multiplex networks with the same edge overlap values as Figure 5.3, for networks with size 100 and 5000. The behaviour of the neighborhoods uniqueness with respect to the degree uniqueness is similar as in the monoplex case. The only difference is that, with certain values of overlap (e.g 0.5, in Figure 5.5 *d*), the neighborhoods uniqueness curve diverges significantly from the degree uniqueness curve before than in the monoplex case. Also, when there is no overlap (e.g. Figure 5.5 *c*), the multiplex curves diverge before the the monoplex ones, but not so in advance as with $ov_E = 0.5$. In big networks, the fraction of nodes with unique degrees combination is less than in small ones, thus nodes with unique degree combination would be significantly less identifiable in big networks than in small ones, as can be seen from the gap between the curves of degree uniqueness and neighborhoods uniqueness in Figure 5.5, which is noticeably different in a network with 100 nodes (Figures 5.5 *a* and *b*), where the degree uniqueness reaches values between 0.6 and 0.8, right after the neighborhoods uniqueness is 1, and 5000 nodes (Figures 5.5 *c* and *d*), where the degree uniqueness is less than 0.1 (the degree uniqueness would reach higher values when the average degree is very high, and the neighborhoods uniqueness has reached the value one for an average degree of less than 90, as can be seen in Figure 5.5).

5.3.5 Unique neighborhoods in Erdős-Rényi graphs

In the previous sections, we discussed the unique degree uniqueness in ER networks, mentioning its relation with the neighborhoods' uniqueness. In this section, we discuss more in depth the trend of the latter, both in the monoplex and multiplex case. We later conduct the same analysis with networks generated with WS (Section 5.4.2) and RGG (Section 5.5.2) graphs, to compare the neighborhoods' uniqueness in graphs with different structure.

Figure 5.4 shows the uniqueness trend for small monoplex networks with 100, 200 and 300 nodes with the all the possible values of average degree, from zero to $n-1$. Despite the different network size, the uniqueness of neighborhoods goes to the maximum value (one) almost immediately. Differently from the degree uniqueness, the neighborhoods' uniqueness does not reach its maximum value when the average degree is half of the possible maximum one ($\langle k \rangle = \frac{n-1}{2}$), but before it. However, when the network is complete, all the nodes are connected to each other, thus all the

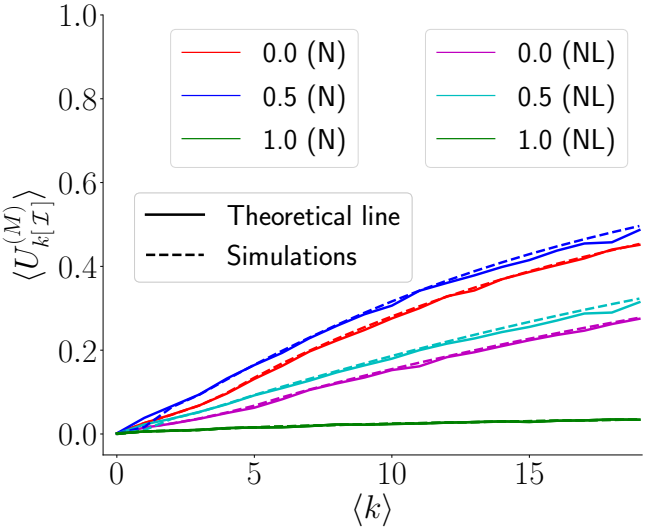

Figure 5.3: Fraction of unique degree nodes in ER multiplex networks of size 100 with different average degree (on the horizontal axis), with respect to both node (indicated in the legend with N) and node-layer isomorphism (indicated in the legend with NL), with edge overlap values of 0, 0.5, and 1.0. The continuos lines are given by the values computed with the simulations, while the dashed lines are given by the values computed with the theoretical formula.

neighborhoods are the same, and there is no uniqueness. The neighborhoods' uniqueness starts decreasing only when the network is almost complete. Indeed, the number of unique neighborhoods is high even when the network is dense, but not complete. As for unique degrees' nodes, this behaviour in the almost complete region may not be observable in multiplex networks, since, in our experiments, the maximum average degree $(n-1)$ is the one in the aggregated network, thus, even if when $\langle k \rangle = n-1$, not all the networks in each layer may be complete, depending on the value of edge overlap ov_E.

We have now seen that the uniqueness reaches its maximum value quite soon when the average degree increases, despite the network size. To have a better overview of the uniqueness variation with respect to both average degree and number of nodes, Figure 5.6 *a* shows the uniqueness' trend in monoplex ER networks varying those two parameters. We can see that the uniqueness goes to zero faster, smaller the average degree is. On the other hand, the uniqueness increases to one almost immediately with all the network sizes. For instance, with average degree of two, it has already reached

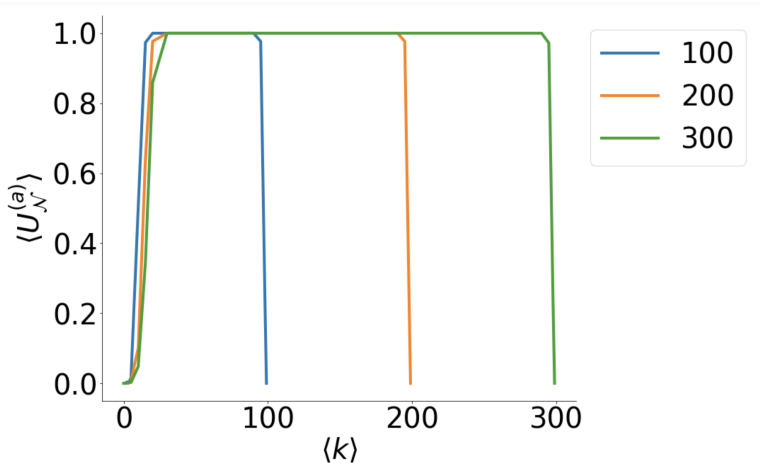

Figure 5.4: Uniqueness of neighborhoods variation in monoplex ER networks of size 100, 200 and 300, for the complete range of average degree of each network.

one with relatively small network size (this has been also seen in Figure 5.4). This trend means that there would be more unique neighborhoods as the average degree increases, at least in the sparse region. Indeed, the higher the average degree is, the higher the possibility of neighborhood formation is, since, being more nodes present in a neighborhood, for example, more combination of triangles would be possible. The uniqueness in multiplex networks would have a similar trend than Figure 5.6.

In multiplex networks, the edge overlap value also influences the uniqueness values (as it has been seen for degree uniqueness in Figure 5.3). Figure 5.5 shows the proportion of unique neighborhoods in networks of 100 and 5000 nodes, for multiplex networks with edge overlap of 0.0, 0.5 and 1.0. With respect to the uniqueness, the edge overlap equal to 1.0 have the the same behaviour in both type of isomorphism and is the same as in monoplex networks (indeed, with $ov_E = 1$, the two layers are exactly the same). Independently from value of edge overlap, the uniqueness with respect to node-layer isomorphism is always lower than node isomorphism, as expected (and as it is also valid for degree uniqueness). From those figures, we can also see that with edge overlap of 0.5 (Figures 5.5 *b* and *d*), the multiplex uniqueness is higher than in the monoplex one. This difference becomes more evident as the network size increases (see the difference between the network with 100 and 5000 nodes). However, with $ov_E = 0$ (Figures 5.5 *a* and *c*), the uniqueness in the monoplex network (which is the same as in the aggregated one) is not always lower than in the multiplex ones. As can

be seen from Figure 5.5 *c*, the monoplex/aggregated uniqueness becomes higher than the multiplex one when the average degree increases. This is justified by the way neighborhoods are extracted (as explained before in Section 4.1).

The influence of different edge overlap values on the uniqueness is shown in Figure 5.7 *a*, where there is also a comparison between the different type of isomorphism and the aggregated networks in terms of uniqueness (for networks with 5000 nodes). With edge overlap value equal to 1, the uniqueness points of multiplex and monoplex networks again correspond. The areas where the aggregated uniqueness is higher than the multiplex uniqueness are the ones with low overlapping proportion and seem wider as the average degree increases. In fact, with low overlapping values, there are less shared edges between the layers, thus there are more possibilities that two nodes x and y could be neighbors of another node n in one layer but not in another one. Moreover, the chances that those two nodes x and y are also connected to each other are greater when the average degree is higher (since there are more edges in the network in general).

From Figure 5.7 *a*, it also emerges that an overlapping proportion of 0.5 (or, however, near 0.5) leads to the maximum amount of uniqueness. This means that, with overlapping proportion of 0.5, there are more possibilities in neighborhood formation, thus higher chances of having different isomorphism classes. This result is also related to the isomorphism classes count (discussed in Section 4.1.5) or graph enumeration problem. Indeed, as said before, the graph enumeration is a difficult problem, especially in multiplex networks, and the pick of uniqueness at overlap of 0.5 suggests that there are generally more multiplex networks with this edge overlap value.

We can get an idea on how fast the uniqueness' transition from the minimum to the maximum value is, by looking at Figures 5.8 *a* and *b*, that show the uniqueness transition from value 0 to 1 with different network size and average degree in multiplex ER networks with respect to node isomorphism. We can see that the transition from $U_{\mathcal{N}_\tau} = 0$ to $U_{\mathcal{N}_\tau} = 1$ is quite fast relatively to the width of the two areas with extreme values. Also from this Figure, we can see that $U_{\mathcal{N}_\tau}$ is generally higher in bigger and more dense networks with $ov_E = 0.5$ compared to $ov_E = 0$. Despite the differences in the values between the shown multiplex with respect to node isomorphism and node-layer isomorphism and aggregated networks, the width of the area in between $U_{\mathcal{N}_\tau} = 0$ and $U_{\mathcal{N}_\tau} = 1$ is almost the same in those other cases.

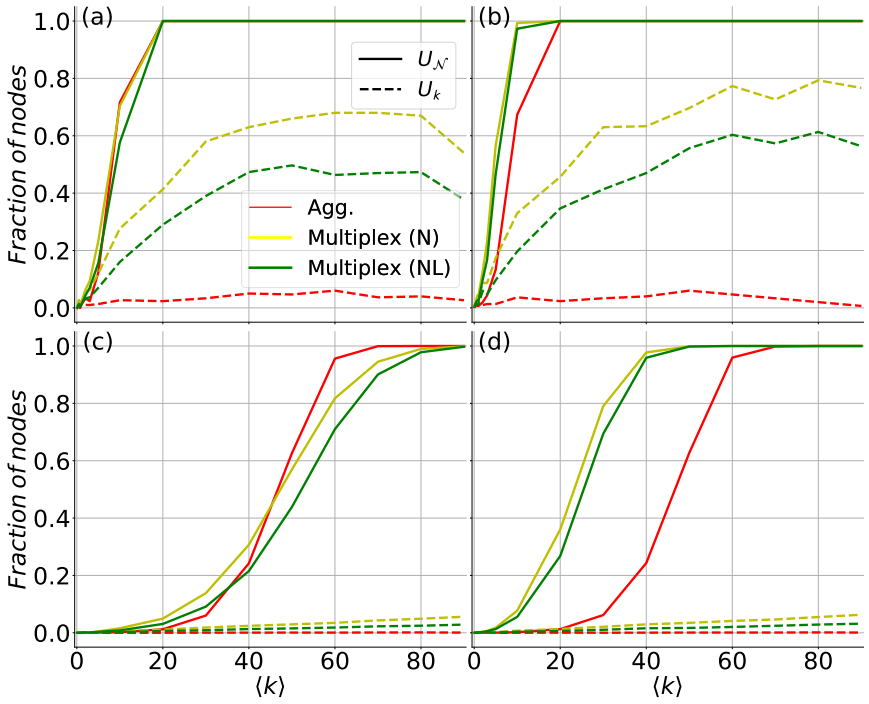

Figure 5.5: Expected uniqueness of neighborhoods (continuos lines) and fraction of unique degree nodes (dashed lines) in ER multiplex and monoplex networks of size 100 and 5000 with different edge overlap values, with respect to node and node-layer isomorphism. *a*: 100 n, 0.00 $ov_E = 0$; *b*: 100 n, 0.5 $ov_E = 0.5$; *c*: 5000 n, 0.00 $ov_E = 0$ *d*: 5000 n, $ov_E = 0.5$. The legend for all the figures is placed in figure *a*. *Agg.* stands for aggregated network; N stands for node isomorphism; NL stands for node-layer isomorphism.

5.4 Watts-Strogatz model

Similarly to what we did for the ER model, we conduct experiments to determine the uniqueness in Watts-Strogatz networks. The graphs generated with the WS model have a different structure compared to the ER ones. We want to study the networks' anonymization difficulty in different settings, thus we use different models to represent various graph structures and understand their behaviour in terms of uniqueness. We first present the multiplex version of the Watts-Strogatz model we use, and then the

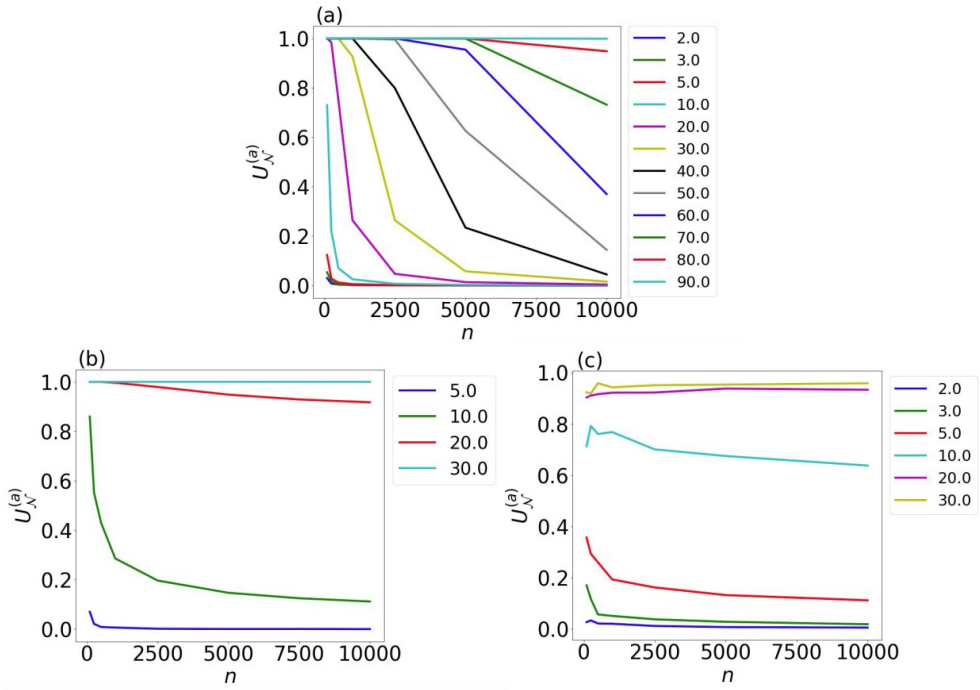

Figure 5.6: Expected uniqueness of neighborhoods (vertical axis) in monoplex network models, with different network size (horizonatal axis) and different average degree (in different colors). The values of average degree are reported in the labels. *a*: ER (the lower values refers to the lower lines. The average degree increases as the lines move towards higher values of uniqueness); *b*: WS (with $\beta = 0.5$); *c*: RGG.

simulation results.

5.4.1 Multiplex Watts-Strogatz model

In an ER network, any pair of nodes have the same independent random probability to be connected to each other, leading to a low clustering coefficient. However, real-world networks have a higher value of local clustering coefficient, since the nodes tend to be organized in a certain way, depending on the modelled context. As explained in Chapter 2, Section 2.5.2, Watts-Strogatz model addresses this limitation of ER. To generate a multiplex version of a Watts-Strogatz networks, we exploit the ring-lattice structure to control the correlation between the different layers. We keep the edges that have not been rewired in all the layers, and split the rewired edges

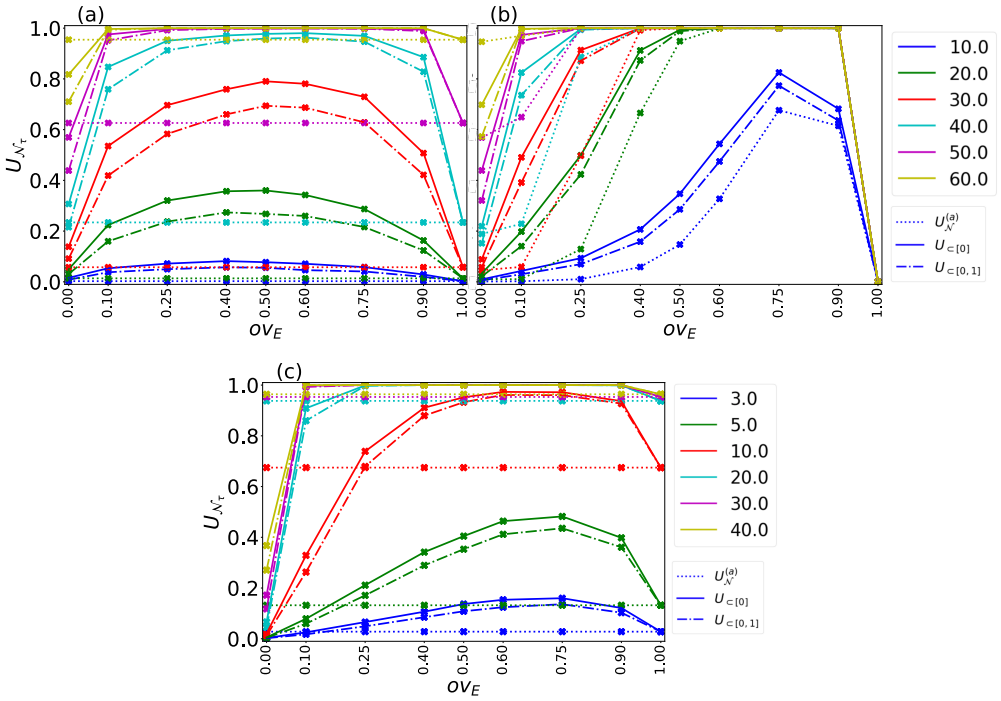

Figure 5.7: Expected uniqueness of neighborhoods (vertical axis) in network models in multiplex networks with respect to node and node-layer isomorphism, and in monoplex networks, with different overlapping proportion (horizontal axis) and different average degree (in different colors). The values of average degree are reported in the labels. The smaller average degree corresponds to the bottom lines, and the higher ones are the top lines. *a*: ER; *b*: WS (for monoplex networks, instead of the edge overlap proportion, the probability of rewiring β is reported); *c*: RGG. The legend of *a* and *b* is the same and is placed in the upper right corner.

between the layers. The meaning of rewiring edges is to shorten the average path length, going towards a randomized structure as the probability of rewiring increases. With this method of generating a multiplex WS network, we keep the "more organized" part of the system (represented by the ring lattice) unchanged in all the layers: this can represents, for example, entities that, in different contexts (represented by the layers), behave partially in a known way, but also have a certain amount of unpredictable behaviour.

Practically, the multiplex Watts-Strogatz graph is built by first generating a monoplex Watts-Strogatz network (with the algorithm described in Section 2.5.2) and then copying the non-rewired edges in both layers, and copying

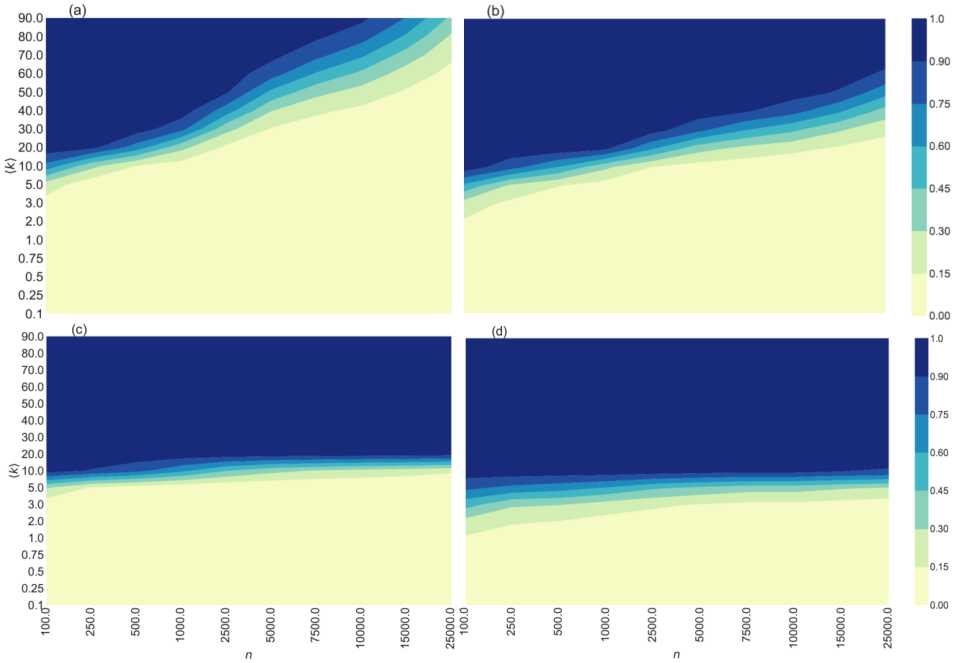

Figure 5.8: Heatmaps representing the variation of the uniqueness of neighborhoods value (in color: the blue area corresponds to a uniqueness of 1, while the yellow area correspond to a uniqueness of 0) in multiplex networks with respect to node isomorphism. The horizontal axis represents the network size, while the vertical axis represents the average degree. Figures a and b are about ER networks with edge overlap of 0.0 and 0.5, respectively. Figure c reports the values for the WS model, and d for RGG.

half of the remaining edges in one layer and the other half in the other layer. Thus the edges that stayed in their original place in the ring lattice would be the overlapping ones, while the others would not overlap. The probability of rewiring β controls the overlap, and can be computed given the desired ov_E (at least in large and sparse networks, since in small and dense networks there can be overlap by random chance, since, in this case, a randomly rewired edge could lead to the creation of an edge that already exists in the initial ring lattice configuration):

$$\beta = 1 - ov_E \,. \tag{5.21}$$

In fact, if $ov_E = 1$, then $\beta = 0$, and all the existing edges would stay in the original place as part of the ring-lattice. Conversely, if $ov_E = 0$, then $\beta = 1$, there would be no edges in the ring lattice, and, consequently, no overlap.

5.4.2 Unique neighborhoods in Watts-Strogatz graphs

The Watts-Strogatz (WS) model has a different parameter compared to ER, the probability of rewiring β, with which we control the overlap in the multiplex networks (we have explained in subsection 5.4.1 how we construct a multiplex WS graph) which leads to different configurations even in the monoplex networks. With β equal to zero, the network remains in its initial configuration, such the ring lattice one, with uniqueness equal to zero (in both the monoplex and multiplex versions), since all the nodes are linked to k neighbors. The average degree needs to be at least 2 (and, in general, an even integer), since the model requires each node be connected to $k/2$ neighbors on each side (as explained in Section 2.5.2). For this reason, differently from the ER network, there are different values of uniqueness in a WS monoplex network with different β.

The influence of the parameter β even in the monoplex version is visible in Figure 5.7 b, where the uniqueness with different values of β are compared for a network with 10000 nodes with respect to all the considered types of isomorphism. We can see that the uniqueness' trend is similar both in the multiplex and monoplex cases (due to the way the network is built, always starting from a monoplex network with a given β). In general, the uniqueness with respect to node isomorphism is higher than the one with respect to node-layer isomorphism (as expected), and the uniqueness in the aggregated network is generally higher than in the multiplex one with high values of β (which correspond to low value of overlapping proportion, as in Equation 5.21).

With $ov_E = 0$, the values of uniqueness are similar to the ones in ER networks (Figure 5.7 *a*), since, with maximum probability of rewiring, the network reaches a configuration close to an ER graph (thus the uniqueness would behave in the same way). From Figure 5.7 *b*, we can also notice that the maximum value of uniqueness is reached at 0.75 of overlapping proportion ($\beta = 0.25$), instead of 0.5 in ER. This means that there is a higher amount of neighborhoods' isomorphism classes when the network is closer to the ring lattice and less than half of the edges have been rewired. When the overlapping proportion increases, at the same value of ov_E, the uniqueness in WS networks is generally higher than in ER networks (see for example Figures 5.7 *a* and *b*, with average degree 10, 20 and 30), also in the monoplex case. This can also be seen from Figure 5.8 *c*, showing the area with $U_{\subset[0]}$ equal to one and zero in multiplex WS networks with $ov_E = 0.5$. The area with $U_{\subset[0]} = 1$ is wider than in the corresponding ER graph with the same edge overlap value (Figure 5.8 *b*), and, according to what we have discussed in relation to Figure 5.7 *b*, it would be bigger with $ov_E = 0.75$,

and, on the other hand, for ER, it would approach to the values in Figure 5.7 *a*.

With a fixed network size, assuming a growth in the values of the average degree, WS graphs take a different amount of time to pass from $U_{\mathcal{N}_\tau} = 0$ to $U_{\mathcal{N}_\tau} = 1$, depending on the probability of rewiring. This can be seen from the already presented Figure 5.7 *b*, but also from Figure 5.6 *b*, which shows the uniqueness in monoplex WS networks with $\beta = 0.5$. If β was equal to 1, the figure would have been similar to the one about an ER network (Figure 5.6 *a*), while, if β was lower (e.g. at 0.25, but not close to 0), the uniqueness lines would have appeared even more flat, close either to the bottom or the top of the figure, showing a very fast uniqueness transition from 0 to 1 with increasing average degree. Indeed, in WS network, by increasing the average degree, the uniqueness shifts from 0 to 1 faster than in ER networks (as it can also be seen from Figure 5.7 *b*, where the width of the zone between 0 and 1 is smaller then in ER networks).

Basically, with high edge overlap proportion ov_E (thus with low probability of rewiring β), the network presents some more organized structure compared to a random configuration, and this leads to higher values of uniqueness, since some neighborhoods would also be more dense and the network would not always be locally tree-like. On the other hand, if the overlapping proportion is too high, there would be almost no uniqueness since the network would approach the ring lattice structure which does not present any unique neighborhood itself.

5.5 Random Geometric Graph model

In this section, we present the result of the simulations on a multiplex version of the Random-Geometric Graph. This model can represent an additional network structure, more realistic than the previously considered models. Nodes in the network are indeed organized in groups (delimited by a radius), and this makes the Random Geometric Graph a more realistic model compared to either Erdős-Rényi or Watts-Strogatz.

5.5.1 Multiplex Random Geometric Graph model

As the Watts-Strogatz model addresses the limitation of the Erdős-Rényi model to have low clustering coefficient, it does not take into account the possibility of being locally dense, but globally sparse, as it is typical of real-world networks (which, for instance, have communities). Even though a WS graph presents a certain kind of local structure given by the ring

lattice, it does not have local structures that are densely connected, as a Random Geometric Graph has. To generate the multiplex version of the RGG model, we exploit the local structures (formed by nodes distant to each other within a certain radius) to control the overlap. The overlap can be interpreted as the correlation between groups of nodes (that are located within a certain radius) across different contexts (for example, if those dense regions are communities of people, we can model the correlation of their behaviour across different social networks).

A multiplex Random Geometric Graph consists of multiple layers (two in our case) generated as a Soft Random Geometric Graph, where nodes are placed in the same position across all the layers, and the radius value within which two nodes are connected with a certain probability is also equal for all layers. We place the nodes uniformly at random in a bi-dimensional space. The length of the radius and the probability that two nodes are connected depend on the overlapping proportion. Indeed, we can control the overlap by changing the probability of connecting two nodes within a certain radius: for example, with $ov_E = 1$ (all the edges are equally present in each layer of the network) all the nodes within a certain radius would be connected to each other and, since the nodes are in the same position in all the layers, all the edges would overlap; whereas, if $ov_E = 0.5$, half of the possible edges would be present between nodes within a certain radius and, in a large enough network, this would result in just half of the edges overlapping.

The probability p that there is a link between two nodes within a certain radius is indeed equal to ov_E (with $0 < ov_E \leq 1$). From this consideration and the formula for average degree in RGG (Equation 2.28), we can obtain the radius value needed for a certain overlap (the radius value is necessary to generate the network in each of the layers). We can obtain the average degree $\langle k_i \rangle$ of the network in each layer L_i by incorporating the probability of an edge to be present within a certain radius in Equation 2.28:

$$\langle k_i \rangle \approx p\pi(n-1)r^2\,, \tag{5.22}$$

Since:

$$p = ov_E\,, \tag{5.23}$$

Equation 5.22 becomes:

$$\langle k_i \rangle \approx\ ov_E\,\pi\,(n-1)\,r^2\,. \tag{5.24}$$

The radius value is obtained by solving the above Equation 5.24 for r:

$$r = \sqrt{\frac{\langle k_i \rangle}{\pi\,(n-1)\,ov_E}} = \sqrt{\frac{\frac{2m}{n}}{\pi\,(n-1)\,ov_E}}\,. \tag{5.25}$$

We can also obtain the average degree in each layer in relation to the one in the aggregated network $\langle k_a \rangle$ from the above formula and Equations 5.2, 5.4 and 5.7. Indeed, from Equation 5.7:

$$\begin{aligned} \langle k_i \rangle &= \langle k_{ind} \rangle + \langle k_{ov} \rangle \\ &= \frac{\langle k_a \rangle (1 - ov_E)}{2} + (\langle k_a \rangle \times \langle ov_E \rangle) \\ &= \frac{\langle k_a \rangle (1 + ov_E)}{2}, \end{aligned} \tag{5.26}$$

which, solved by $\langle k_a \rangle$, while plugging the formula of $\langle k_i \rangle$ from Equation 5.24, becomes:

$$\begin{aligned} \langle k_a \rangle &= \frac{2 \langle k_i \rangle}{1 + ov_E} \\ &= [ov_E \, \pi \, (n-1) \, r^2] \times \frac{2}{1 + ov_E}. \end{aligned} \tag{5.27}$$

Since we do not admit multi-edges (i.e. multiple edges between the same pair of nodes), the number of nodes within a certain radius is limited, consequently also the possible amount of edges is. For this reason, when the expected average degree is high, especially for small networks, the actual average degree reached with this computation is lower than the expected. Moreover, the equation of the original (monoplex) model, 5.22, would be an exact equality (and not approximate, as it is) while assuming periodic boundaries. We are not using periodic boundaries, thus some nodes may not be linked by an edge even if they should be. This situation is especially noticeable with small and dense networks.
However, to remedy the problem of obtaining an average degree smaller than expected (especially in dense networks), we can use a heuristic, such as increasing the radius value by an additional factor based on the network size (such that more nodes would be connected with a link). For example, we can modify Equation 5.25 by increasing the expected average degree by an additional value, that, instead of being $\frac{2m}{n}$, would be, for example, $\frac{2.2m}{n}$ (in this case the additional factor is $\frac{0.2m}{n}$). We have empirically seen that the following additional factors would give networks with average degree close to the expected value:

- $\frac{0.2\,m}{n}$, for network size $n \leq 1000$;
- $\frac{0.15\,m}{n}$, for $1000 < n \leq 10000$;
- $\frac{0.05\,m}{n}$, for $10000 < n \leq 100000$.

In Equation 5.25, if $ov_E = 0$, then $r \to \infty$, and all the nodes are within the same distance radius (practically, we need to select a radius large enough to contain all the nodes, for example 100000). The network becomes then similar to an ER graph, where two nodes are connected with a certain probability, regardless of their position. In this case, the probability p is given by the formula in Equation 2.1. Since the average degree $\langle k \rangle$ in Equation 2.1 is the one of the aggregated network, to obtain the probability that two nodes are connected in each layer, we need to divide p by two (since there are two layers), thus:

$$p = \frac{\langle k_a \rangle}{2} \times \frac{1}{n-1}\,. \tag{5.28}$$

5.5.2 Unique neighborhoods in Random Geometric Graphs

In Random Geometric Graphs, the overlap is controlled by the radius value and the average degree. In Section 5.4.2 we have seen that, in WS graphs, a more organized structure leads to higher values of uniqueness compared to networks with more randomness. RGG graphs present even more local structures than WS graphs, since a group organization emerges from the fact that nodes are connected to each other within a certain radius.

Figure 5.6 *c* shows the variation of uniqueness in a monoplex RGG network with different network size and average degree. We can see that, the uniqueness' curves decrease very slowly with the growth of the network size, and certainly more slowly than in ER networks. In node and node-layer isomorphism (not shown here), we have almost the same flat behaviour. We can see this kind of trend also from Figure 5.8, that shows the uniqueness in RGG with respect to node isomorphism with edge overlap equal to 0.5. The flat behaviour can be explained by the fact that the network present a always a "community structure" [D+18], and there would be always group of nodes independently on the size (as it is typical of real world networks). This trend would not be present anymore if the edge overlap value is equal to zero. Indeed, in this case, the radius would go to infinity and the network would not present any local structure anymore, but two nodes would be connected with a certain probability despite their distance, and the network would be exactly like an ER graph (the uniqueness would behave as, for example, in Figure 5.6 *a*).

The networks generated with the RGG model present a similar trend in terms of uniqueness for all the overlapping proportion (excluding 0). This is justified by the non-significant variation in the radius value, despite the difference in the edge overlap. Indeed, for instance, in a multiplex network

of 10000 nodes and average degree $\langle k \rangle = 10$, the radius value with $ov_E = 1$ results to be 0.018 (according to Equation 5.25), and the number of edges in each layer 50000; with the same settings, changing the overlapping value to ov_E to 0.5, leads to a radius equal to 0.022, or to 0.029 when $ov_E = 0.25$; when instead $ov_E = 0$, the radius goes to 2.82, which is a significantly higher value compared to the one corresponding to other overlapping values.

However, besides the uniqueness' flat trend being very similar, the edge overlap value plays a role in computing the actual uniqueness value. Indeed, as can be seen from Figure 5.7 c, as for the WS model, the maximum uniqueness corresponds to an edge overlap of almost 0.75. In this Figure we can see, as in the other models, that with low overlapping values, the aggregated uniqueness is higher than the multiplex one.

5.6 Uniqueness' linear trend and models comparison

In this section, we present the results of a more precise of the uniqueness' trend in random networks. Having now an overview on how the uniqueness behaves with different network models, we estimated better its variation through a modified version of a binary search algorithm. Through this analysis, we have discovered that the uniqueness has a linear trend depending on network size and average degree. We first present the adopted algorithm for our analysis (subsection 5.6.1) and, then, the results (subsection 5.6.2).

5.6.1 Binary Search

To better compare the uniqueness of neighborhoods in different models, we run a binary search algorithm that looks for a certain value of uniqueness in a network generated according to a model with a given number of nodes, in a range of average degree values delimited by two extremes. In particular, we have seen in the previous sections of this chapter that the uniqueness transition from value 0 to value 1 is relatively fast, thus 0.5 as uniqueness value (which corresponds to having half of the neighborhoods in the network unique) can give us a good idea of this trend in function of average degree and network size (as it is similar shown in the heatmaps of Figure 5.8). With a binary search algorithm, we can have a better estimate of the curve corresponding to $U_{\mathcal{N}_\tau} = 0.5$.

Our algorithm is a stochastic and continuos version of the classical binary search algorithm [Wei]. The binary search algorithm searches for a target

value in a certain range, evaluating first the extreme values of the interval and, if those are not the ones we are looking for, evaluating the middle value. If the middle value corresponds to the target value, the algorithm stops, otherwise we continue the search process with a new interval corresponding to either the lower part of the interval (delimited by the original lower extreme and the middle value), or the upper part (delimited by the middle value and the original upper extreme).

In our binary search algorithm, with a given network model and network size, the target value is the value of average degree corresponding to uniqueness of neighborhoods of 0.5. To decide on which side of the interval to move, we exploit the fact that, with a fixed network size, the uniqueness grows with the average degree (at least in the sparse region). Thus, if a certain average degree value we are evaluating gives a uniqueness value higher than the one we want, we move to the left (or lower) part of the interval, which is the one containing lower values; otherwise we move to the right part, by always computing the middle value of the new interval. The binary search is a recursive algorithm, and it continues until we find the value corresponding to 0.5 uniqueness, or the extremes of the interval we are evaluating are too close to each other.

To compute the uniqueness value corresponding to each average degree, we generate five networks with that average degree and the given network size, and we take the mean of the corresponding uniqueness values. Since we want to be sure every time we decide on which new interval to evaluate, and also when to stop, we compute a confidence interval (at 99% confidence level in our case) of the mean of the uniqueness value of the networks generated with certain parameters and we check whether the target uniqueness value is contained in that interval: if it is not, we move either to the right or the left side; if it is, we do new simulations to have a better estimation of the real mean and, if after a maximum number of simulation (we chose 30) the target uniqueness value is still in the interval, then the evaluated average degree is the one we are looking for, otherwise, we continue with the search. We also set a tolerance level to the target uniqueness value (we chose 0.02, thus if we find an average degree corresponding to either 0.52 or 0.48, the search process is considered successfully ended).

Every time we decide on where to move, we do it at 99% confidence level, thus the total confidence level of the whole process is 0.99^D where D is the number of decisions taken. With 30 decisions (the maximum number of simulations we allowed), we obtain a confidence interval of 73%, which is an acceptable confidence level.

5.6.2 Results

We have ran the binary search algorithm illustrated in subsection 5.6.1 for multiplex networks with overlapping proportion of 0.0, 0.5 and 1 for all the three network models previously presented (ER, WS, RGG), for network sizes from 100 to 10000 and with an interval of average degree between 1 to 100.

In Figure 5.9 we can see the results of the algorithm with respect to node and node-layer isomorphism with all the three evaluated network models. We have plotted the results in a log-log scale. The reported curves correspond to a uniqueness value of 0.5 (the middle of the uniqueness transition areas in the heatmaps in Figure 5.8), and, below the curve, there is the area with uniqueness lower than 0.5 (almost all 0, since we have seen that the transition from 0 to 1 is fast), while above the curves, the uniqueness value would be equal to or at least near 1. When the curves' trend is to increase, it means that the general uniqueness is decreasing at the same network size. In other words, with a fixed average degree and higher number of nodes, the uniqueness would be lower. Thus, to obtain the same value of uniqueness we need to increase both network size and average degree.

We can observe that the uniqueness decreases linearly in function of network size and average degree, therefore we fit a straight line to the curves (the fitted lines have equations $log(y) = m \times log(x) + c$). We can see that, apart from an edge overlap value equal to 1, the node isomorphism generally leads to higher uniqueness than the node-layer one. In ER networks, when the overlap is equal to 0, the behaviour is almost the same as in monoplex networks and, in general, the lines corresponding to different overlap values are almost parallel to each other.

The 0.5 uniqueness lines of ER networks are among the ones with greater slope compared to the other lines. When the other networks have edge overlap value equal to 0, the behaviour is similar to the one of ER networks (as we have also seen in the previous sections). This means that, with high randomness in the structure (represented by the ER networks), when the network size and average degree grow, the uniqueness goes significantly down.

When we increase the amount of local structure, such as with WS or RGG models, the uniqueness decreases more slowly. Indeed, when passing from WS to RGG, the slope of the lines decrease. Obviously, at an infinite size, also in Random Geometric Graphs there would be no uniqueness, but the uniqueness value is less dependent on the size. In fact, the lines corresponding to uniqueness equal to 0.5 are almost parallel to the horizontal axis. With edge overlap of 0.5, the uniqueness area in RGG is higher than

with overlap of 1.0. This is expected since, with total overlap, the network has the same neighborhoods of its aggregated version. The edge overlap of 0 for WS is not reported since, with that value, all the edges are overlapping, thus the there is no uniqueness. (this is due to the way we construct the network, explained in Section 5.4.1). However, in WS, there can be different configurations even in the monoplex networks, thus we have reported the uniqueness with probability of rewiring $\beta = 0.5$ in monoplex network. We can see that the corresponding line is parallel to the ones of multiplex networks with overlapping proportion of 0.5. However, as expected, the uniqueness is lower. To observe a higher uniqueness value in the aggregated network than in the multiplex one, we would need to lower the probability of rewiring, as can be seen in Figure 5.7).

To conclude, as we go from a more organized and locally dense structure towards more randomness and less density in the network structure, the uniqueness in networks would depend more and more on the size and average degree value, and, in general, tend to decrease as these two parameters grow. Overall, we have higher values of uniqueness that depends less and less on the size when the network local structure is more pronounced. This is also confirmed by the fact that, in multiplex WS and RGG, the peak of the uniqueness is with an edge overlap value of 0.75 (Figure 5.7), which, in both models, means that there is more cohesion in the structure than randomness. Indeed, we have complete randomness with edge overlap of 0.00 and a minimum amount of it with edge overlap equal to 1. With edge overlap higher than 0.75, thus closer to 1, the uniqueness decreases, since the number of isomorphism classes naturally decreases, as there are fewer possibilities of diverse neighborhoods formation.

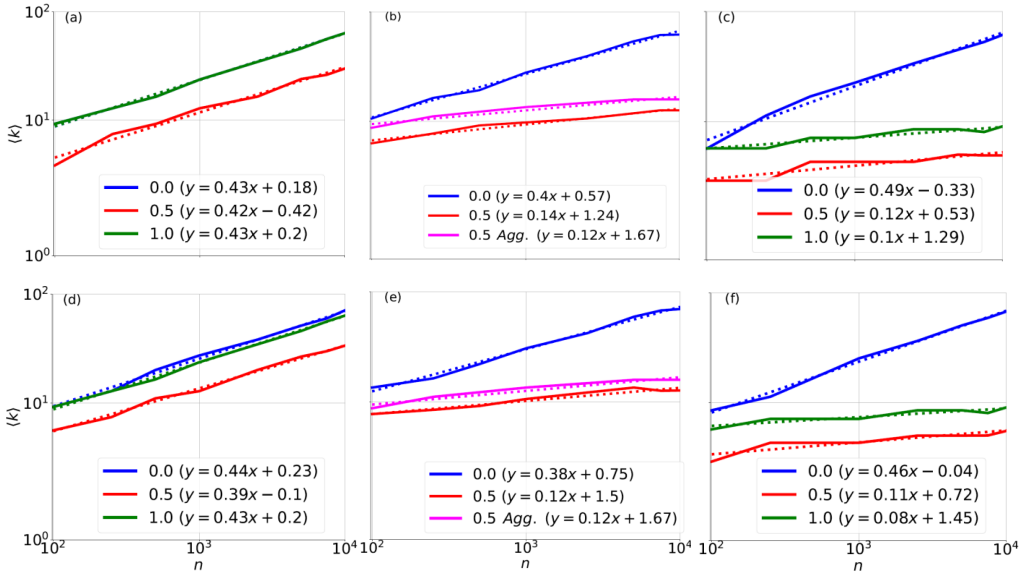

Figure 5.9: Lines representing 0.5 uniqueness in three network models, ER, WS and RGG (from left to right), estimated with a binary search algorithm, in a log-log scale. Figures a, b and c represent the uniqueness with respect to node isomorphism; Figures d, e, f represent the uniqueness with respect to node-layer isomorphism. The horizontal axis is the network size n, while the vertical axis is the average degree $\langle k \rangle$. The area below the line is the one with uniqueness < 0.5, while above the lines the uniqueness is > 0.5. The edge overlap values reported are 0.0 (in blue), 0.5 (in red) and 1 (in green). The edge overlap equal to 1.0 is not reported for the WS model since the uniqueness is always equal to zero. However, in the Figures regarding the WS model (b and e), the monoplex uniqueness is reported with probability of rewiring β equal to 0.5 (in magenta). The continuos lines are the ones obtained with the simulations during the binary search process, while the dashed lines are the corresponding linear fit (that have equations $log(y) = m \times log(x) + c$).

Chapter 6

Uniqueness of neighborhoods in empirical networks

In Chapter 5, we have studied, through the usage of network models, how the uniqueness of neighborhoods changes with the network structure and how the multiplexity can influence it. On top of this, it is also crucial to study how the uniqueness varies in empirical data, and understand if the models can be used as a proxy for real-world networks in terms of anonymization difficulty.

In this chapter, we analyze how the uniqueness varies in some empirical networks with different features, such as size, average degree and clustering coefficient. We first introduce the considered datasets in Section 6.1, along with their basic network features, then, in Section 6.2 we show the distribution of the neighborhood structures' occurrence frequency, compare them and discuss how the uniqueness of each network relates to its network features. Finally, in Section 6.3, we discuss if the presented datasets are comparable to the models analyzed in the previous chapter in terms of uniqueness (still according to the neighborhood definition $\mathcal{N}_{\subset}$, presented in Chapter 4).

6.1 Datasets

We consider three different social network datasets. We analyze the network data in a multiplex setting, considering also each single-layer separately and aggregating all the layers into a single one. This analysis can also help to understand the best strategy to share data to minimize the nodes re-identification risk based on neighborhood attack or, alternatively, to

anonymize data with an appropriate algorithm prior to the actual sharing, without modifying the network radically with, for example, edges addition. The different ways to share multiplex data can either be sharing the layer separately, aggregating them, or sharing them directly as a multiplex network.

The datasets we analyze are communication networks, including mainly calls and text messages. These kinds of datasets are common and interesting for research [Sto+14; Kiv+12], and the identity disclosure risk in sharing them is clear, since communication habits, along with some auxiliary information (such as the context, the timestamp or the group of users present in the dataset, or nodes' metadata), can be a fingerprint uniquely associated to one user. The analyzed datasets are the following:

- **(CopNet)** *Copenhagen Network Study* [Sto+14]: a dataset of various interactions (phone calls, text messages, face-to-face interactions, Facebook friendships) among about 800 students of Technical University of Denmark. We consider 7 months of data (from January to July) of year 2014. The face-to-face interactions are recorded with the Wi-Fi signals of the mobile phones that were distributed to the students taking part of the experiment, thus a single person results to interact with others very often, especially during lectures' time at university. For this reason, we thresholded the face-to-face interactions selecting the ones occurring just in the weekends or every evening after 18, with a distance corresponding to 1.5 metres or less (measured by signal strength). Moreover, to avoid selecting meetings occcurring by chance, we considered just the interactions that happened at least once every month.

- **(CountryCalls)** *Country-scale mobile phone calls dataset* [Kar+11; Kiv+12]: a dataset of phone calls and text messages among the subscribers of a mobile operator in an European country, during a period of 7 months (from January to July) in year 2007.

- **(CompanySms)** *Company text messages* [Wu+10]: a temporal network of text messages (SMS) among the members of a company over a period of one month. We built two different networks with the same data: one dividing the data in two snapshots of time, building a two-layers network (called *CompanySms2*), and one with three time snapshots, building a three-layers network (*CompanySms3*).

Tables 6.1 and 6.2 illustrate basic network measures of the considered single-layer and multi-layer networks. The same tables show also the uniqueness of the respective networks (in the monoplex or aggregated networks, or

according to node or node-layer isomorphism for multiplex networks), expressed in terms of fraction of unique nodes with respect to the total amount of nodes; the edge overlap is expressed in terms of amount and fraction of overlapping edges relative to the total amount of edges (intra-layers) present in the multiplex network. For multiplex networks, the values reported in the table of average degree and clustering coefficient (corresponding to the average local clustering coefficient) refer to the ones in the aggregated networks. The uniqueness values of the multiplex networks of *CompanySms* are expressed just according to node-layer isomorphism since in a temporal network layers have a specific order and it does not make sense to share them without labels.

In Section 6.2, we discuss how the network features of each dataset relate to the uniqueness of neighborhoods.

6.2 Data features and uniqueness of neighborhoods

In this section, we discuss the basic features of each of the datasets introduced in Section 6.1 and relate them to the number of unique neighborhoods in the networks.

As it can be seen from Tables 6.1 and 6.2, *CopNet* is the smallest considered dataset, while *CountryCalls* is the biggest one. The uniqueness score of each network built from *CopNet* is significantly higher than the networks in *CountryCalls*. This seems to confirm the finding of the previous chapter about the decrease of uniqueness with the network size. All the single-layer datasets in *CopNet* present almost the same uniqueness value (≈ 0.2), besides the one of Facebook's friendships (FB), whose uniqueness is definitely higher 0.874). This is because the average degree of this network is also the highest of the dataset (20.837 compared to ≈ 4.5 of the other networks), leading to more chances of neighborhood formation.

Being the average degree of FB (in *CopNet*) that high, it also dominates the other datasets when combined into a multiplex network (or aggregated), as can be noticed from the average degree values in Table 6.2. This is also visible from the degree distribution in Figure 6.1 (plotted as Complementary Cumulative Distribution[1], with logarithmic axis, so that we can zoom on

[1]this is not the classical definition of the Complementary Cumulative Distribution (1-CDF), since it gives the probability $P(X \geq x)$, that a variable X takes value greater or equal than x.

Net. name	n	m	$\langle k \rangle$	C	$U_{\mathcal{N}}^{(a)}$
Call - CopNet	695	1606	4.622	0.233	0.201
Sms - CopNet	707	1607	4.546	0.225	0.198
F2F - CopNet	495	1025	4.141	0.278	0.159
FB - CopNet	810	8439	20.837	0.330	0.874
Call - CountryCalls	5193086	10660902	4.105	0.214	0.024
Sms - CountryCalls	4303611	8857642	4.116	0.122	0.025
CompanySms2 - layer 1	31908	32313	2.025	0.049	0.016
CompanySms2 - layer 2	33417	33998	2.035	0.044	0.015
CompanySms3 - layer 1	26309	24249	1.843	0.040	0.011
CompanySms3 - layer 2	28025	26644	1.901	0.038	0.011
CompanySms3 - layer 3	26883	24764	1.842	0.037	0.012

Table 6.1: Dataset's single-layer networks' basic measures (number of nodes n, number of edges m, average degree $\langle k \rangle$, clustering coefficient C) and uniqueness.

Net. name	n	m *(total)*	ov_E	m *(agg.)*	$\langle k \rangle$ *(agg.)*	C *(agg.)*	$U_{\mathcal{N}}^{(a)}$	$U_{\subset[0]}$	$U_{\subset[0,1]}$
Call-Sms (CopNet)	731	3213	1217 (0.379)	1996	5.461	0.245	0.281	0.511	0.481
Call-Sms-F2F (CopNet)	753	4238	1717 (0.405)	2521	6.696	0.245	0.339	0.695	0.645
Call-Sms-FB (CopNet)	836	11652	2960 (0.254)	8687	20.782	0.324	0.854	0.938	0.931
F2F-FB (CopNet)	828	9464	706 (0.074)	8758	21.155	0.330	0.864	0.909	0.889
Call-Sms (CountryCalls)	5559145	19518544	5626536 (0.288)	13892008	4.997	0.202	0.044	0.135	0.127
CompanySms 2	44090	66311	14089 (0.212)	52222	2.369	0.059	0.022	0.053	-
CompanySms 3	44090	75657	23435 (0.309)	52222	2.369	0.059	0.022	0.087	-

Table 6.2: Dataset's multiplex networks' basic measures (number of nodes n, number of total edges $m\,(total)$, edge overlap ov_E, number of total edges in the aggregated network $m\,(agg.)$ average degree in the aggregated network $\langle k \rangle$, clustering coefficient in the aggregated network C), and uniqueness in the aggregated networks, and in the multiplex networks with respect to node and node-layer isomorphism.

the tails of the distribution). *FB* has, in general, nodes with much higher degree compared to *Call* and *Sms*, and, in a multiplex setting, the networks

F2F-FB and *Call-Sms-FB*, when aggregated, have almost the same degree distribution as the monoplex *FB*. Furthermore, this affects the occurrence frequency of neighborhoods (defined in Equation 4.12.), shown in Figure 6.2 as a Complementary Cumulative Distribution of the Occurrence Frequencies $O_{\mathcal{N}_\tau(v)}$. From the Complementary Cumulative Distribution of $O_{\mathcal{N}_\tau(v)}$, we can see how many neighborhoods occur one time by looking at the difference between the second and the first represented value. From the definition of Complementary Cumulative Distribution, the first value is always one, since all the classes are occurring a number of times greater or equal than one. The bigger the gap between the first and the second value, the higher the uniqueness in the network is. From this representation we can also read the *k-anonymity* of the network as $1 - CDF(O_{\mathcal{N}_\tau(v)} = k)$.

From Figure 6.2 *b* we can see that most of the classes in the network *Call-Sms-Fb* are unique or have a low occurrence frequency, while the multiplex network without the Facebook layer, $Call-Sms$, presents a more widespread distribution. The occurrence frequency of the monoplex networks are even more distributed, and the number of neighborhoods with occurring less times also decreases if the multiplex network is aggregated.

In bigger datasets as *CopNet* and *CountryCalls* (Figures 6.2 *c* and *d*), the number of neighborhoods that are $k-anonymous$ for low values of k is high and does not vary significantly. This means that most of the neighborhoods that are $k-anonymous$ are also $(k+1)-anonymous$ with low values of k. This does not happen in the smaller datasets, since there is less amount of neighborhoods and, consequently, less possibilities that two neighborhoods would belong to the same isomorphism class.

CompanySms presents a significantly lower clustering coefficient compared to the other datasets, around 0.05 compared to ≈ 0.2 in almost all the cases. It also presents a lower average degree of ≈ 2 against 4 or 5 in *CopNet* and *CountryCalls*, as well as a lower average degree, ≈ 2 against 4 or 5 in *CopNet* and *CountryCalls*. With such a low value of average degree and density, it is normal that *CompanySms* is the presented dataset with the lowest uniqueness value. Indeed, triangles contribute towards the formation of unique neighborhoods, that, otherwise, would be described by just the degree.

Taking into account the consideration of Chapter 5, according to which the uniqueness decreases with the network size, and being *CountryCalls* significantly bigger than *CompanySms*, one would expect that *CountryCalls* had a lower uniqueness value. However, as discussed previously in the same Chapter 5, there are other factors to take into account, such as average degree and local clustering coefficient. Indeed, the values of average degree and local clustering coefficient in *CompanySms* are less than a half compared

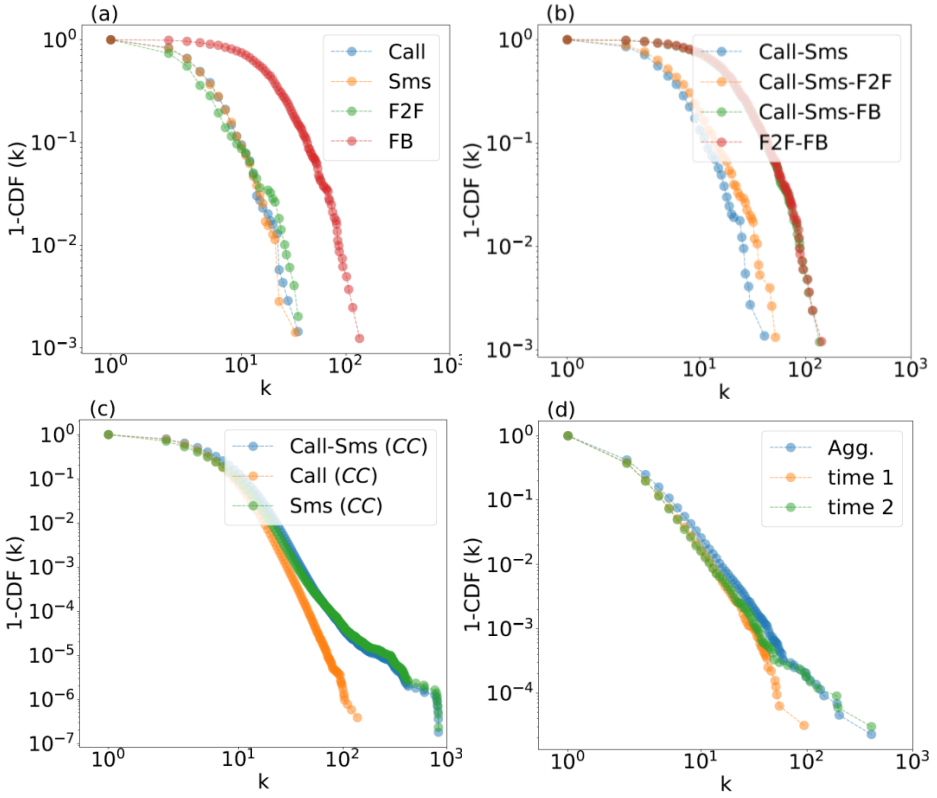

Figure 6.1: Degree distribution of the presented datasets as a Complementary Cumulative distribution, both for the monoplex and aggregated multiplex networks: Monoplex networks of CopNet (a); Multiplex networks of CopNet (b); CountryCalls(CC) (c); CompanySms2 (d).

to the other two datasets (*CopNet* and *CountryCalls*), thus, with less edges per node (indicated by the average degree), there are less possibilities of neighborhoods formation. Additionally, *CountryCalls* presents more nodes with higher degrees than *CompanySms* (as can be noticed from the degree distribution in Figure 6.1 c and d), and, since nodes with many connections are rare, they also contribute to the uniqueness (or at least, to the rarity) of neighborhoods in the network. For instance, if there is only one node with many connections, that node is surely unique because of its degree.

The similarity of single-layer networks in some datasets (such as *Call* and *Sms* of *CopNet* and *CountryCalls*, or the various layers of *CompanySms*) can be seen from the similar average degree and number of nodes, but it is also reflected on the degree distribution (Figure 6.1) and in the occurrence frequencies of the neighborhoods (Figure 6.2).

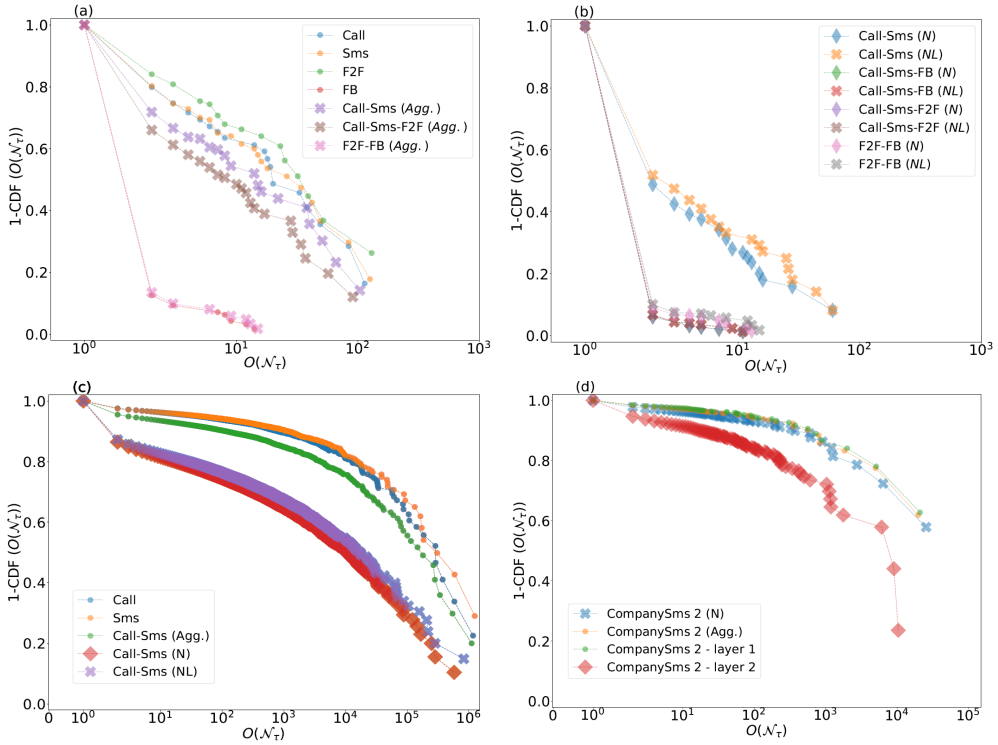

Figure 6.2: Complementary Cumulative distribution (1-CDF) of the Occurrence frequency of neighborhoods in the presented datasets: Monoplex networks of CopNet (*a*); Multiplex networks of CopNet (*b*); Country Calls (*c*); CompanySms2 (*d*). The vertical axis can also be read as the *k-anonymity* of neighborhoods. When specified in the legend, N and NL stand for multiplex network with neighborhood computed, respectively, with respect to node isomorphism and node-layer isomorphism; *Agg.* stands for aggregated network.

For the presented multiplex networks, passing from node to node-layer isomorphism to the aggregated network, the number of classes occurring just one time decreases, especially when the network is aggregated, along with the other classes occurring a low amount of times. This happens because, passing from node to node-layer isomorphism (and also to the aggregated network), the requirements for two neighborhoods for being isomorphic become less strict, thus there would be more neighborhoods in the same isomorphism class. In this case, the aggregation of multiplex networks leads to a lower uniqueness value in comparison with multiplex cases. In Chapter

5 we have seen that, with the neighborhood definition we are using ($\mathcal{N}_{\subset}$), it is not always given that the uniqueness in the aggregated network ($U_{\mathcal{N}}^{(a)}$) would be lower than the one in the multiplex networks ($U_{\subset[\mathcal{I}]}$). However, we have also seen that $U_{\mathcal{N}}^{(a)}$ is often higher than $U_{\subset[\mathcal{I}]}$ in network models when the average degree is not very small. However, the presented datasets have a relatively low average degree (from 2 to 5, or 20 for the Facebook's friendships network in *CopNet*), thus this result is also in line with the findings in the previous chapter.

In terms of anonymization difficulty, knowing how many classes occur a certain amount of time is important, since, if we want to reach *k-anonymity*, we need to anonymize classes occurring less than k times. The fact that there are classes occurring several times in the network (i.e. that are isomorphic to each other), and that other classes are similar to each other, is due to the presence of similar parts in the neighborhoods' graphs. The similarity of neighborhoods' classes allows the anonymization to be possible without radical modification to the network. In Chapter 7, through a motif analysis, we study the recurring patterns in neighborhoods that cause the similarity between them.

6.3 Data and models

In this section, we discuss whether the models analyzed in Chapter 5 are comparable to the presented datasets in terms of uniqueness. We have generated networks with similar features of some the dataset, choosing, for the multiplex networks, the ones with two layers and having similar number of nodes and edges in both layers (since we considered models with equal number of layers and edges in both layers).

The models of the size of the biggest dataset, *CountryCalls*, resulted to have uniqueness equal to zero in the case of Erdős-Rényi (ER) and Watts-Strogatz (WS). From the results of the previous chapter, this was expected, since the uniqueness in ER goes to zero quite fast with the network size, while in WS is usually slower. However, in this case, an edge overlap value of 0.288 (as it is for the *Call-Sms* network in this dataset) means that the probability of rewiring is more than 0.7, which is already close to a random configuration in the WS model. Conversely, the simulation of the Random Geometric Graph shows a uniqueness value greater than zero, meaning that, despite the uniqueness goes to zero towards infinity, probably the size of 5 millions is not yet enough for reaching the minimum uniqueness of zero. The uniqueness for RGG of size similar to *CountryCalls* is shown in

Table 6.3, and it results to be a similar value for the monoplex case. In the multiplex case, instead, the uniqueness value in RGG is similar the one of the aggregated network of *Call-Sms*, but not of its multiplex (which is more than double than in the model). This may be due to the fact that, in a real network, there can be much more diversity in the neighborhood formation than in a model, and this is even more noticeable in the multiplex case, where the number of possible isomorphism classes is much higher than in the monoplex one. However, in the model of size of *CountryCalls*, the aggregated uniqueness $U_{\mathcal{N}}^{(a)}$ is slightly higher that the multiplex uniqueness $U_{\subset[\mathcal{I}]}$. This trend can be explained by the neighborhood definition we adopt, and, as it has been shown in Figure 5.7, $U_{\mathcal{N}}^{(a)}$ can be higher than $U_{\subset[\mathcal{I}]}$ with relatively low value of edge overlap and average degree.

The models with similar features as *CompanySms2*, also resulted in a uniqueness value close to zero for ER and WS in most of the cases. An exception is in the single-layer networks with size ≈ 30000 (in the dataset *CompanySms*), where the uniqueness value in WS is 0.0003, which is still very low, but not exactly zero. The RGG, in this case, presents a lower value of uniqueness than in the model in both the monoplex and multiplex case (however, it is still comparable).

Regarding the smallest dataset, *CopNet*, the networks are still in the area of low uniqueness for an ER network of almost 700 nodes. Precisely, the obtained value is 0.02. For the WS model, for probability of rewiring of 0.7 - 0.8, we have obtained a value of 0.01. These results are justified by the fact that the dataset is more dense than the networks generated with those models, which also do not explain the multiplex cases. Also in this case, RGG better explains the multiplex networks of the datasets compared to other models, with uniqueness values of ≈ 0.70. However, this value is higher than the one in the dataset, including the single-layer/aggregated neighborhood uniqueness which is significantly higher than the one in the dataset. The reason for this can be that the network is still relatively small and, with small networks, the variance could be also higher than in large ones. In general, RGG is the most realistic of the presented models, therefore it is normal that it is the one that best explains the datasets.

Having a model such as RGG that has a similar bahaviour as the data is important to anonymization purposes. Since we have identified a trend in the model, we can compare the uniqueness of the datasets to the one in the model and have an idea on "how far" the data are from being completely anonymous, with $U_{\mathcal{N}_\tau} = 0$. We can then modify the data by changing their size and average degree (i.e. thresholding), without introducing fake nodes or edges, being guided by the models. Alternatively, we can use

anonymization algorithms given that our data do not have uniqueness equal to one and do not have features that collocate them in an area very far from the anonymous one.

Model	n	$\langle k \rangle$	ov_E	$U_N^{(a)}$	$U_{\subset[0]}$	$U_{\subset[0,1]}$	Comparable Nets.				
							$U_N^{(a)}$			$U_{\subset[0]}$	$U_{\subset[0,1]}$
							CopNet Call	*CopNet* Sms	*CopNet* Call-Sms(Agg.)	*CopNet* Call-Sms[0]	*CopNet* Call-Sms[0,1]
ER	700-750	4.5-5	0.38	0.020	0.20	0.16	0.201	0.198	0.281	0.511	0.481
WS				0.016	0.13	0.08					
RGG				0.65	0.78	0.70					
							CompanySms2 layer1	*CompanySms2* layer2	-	-	-
RGG	30000	20.837	0.330	0.009	0.021	0.016	0.016	0.015	-	-	-
							-	-	*CompanySms2*(Agg.)	*CompanySms2*	-
RGG	45000	2	0.2	0.008	0.018	0.014	-	-	0.022	0.053	-
							CountryCalls Call	*CountryCalls* Sms	*CountryCalls* Call-Sms (Agg.)	*CountryCalls* Call-Sms[0]	*CountryCalls* Call-Sms[0,1]
RGG	4500000-5500000	4-5	0.2	0.046	0.044	0.036	0.024	0.025	0.044	0.135	0.127

Table 6.3: Mean uniqueness values of networks generated with models (mean over 5 realizations) comparable to the presented datasets. Since in some cases the values did not significantly change (at least in the first decimal digits) when changing the network size or average degree, they are reported in the same row.

Chapter 7

Neighborhood Motifs Analysis

Having now an idea on how the uniqueness of neighborhoods changes in social networks, we want to understand the basic ingredients and patterns that compose the neighborhoods, allowing some of them to be isomorphic or similar to each other. The similarity of neighborhoods allows anonymization algorithms to work without radically modifying the data. Moreover, in Chapter 5, we focused on uniqueness in random networks, and we have seen that the network structure and density influence the uniqueness value. Real-world networks can have more variety in the local structure than random networks, and, as analyzed in Chapter 6, models cannot always be used as a proxy for the uniqueness of empirical data. This analysis can also reveal more differences between the neighborhood structure of models and real-world networks.

To better understand the structure of neighborhoods in real-world networks, we analyze the network motifs (introduced in Section 2.6) that form the neighborhoods in the datasets *CopNet* and *CountryCalls* (presented in Section 6.1), considering the systems with up to 2 layers composed by the data regarding Calls and Sms from both networks.

Network motifs were defined as "basic building blocks of complex networks" in the original reference [Mil+02]. Similarly, we can define the patterns that we are going to analyze as "basic building blocks of networks' neighborhoods", thus we can call them *neighborhood motifs*. In our case, the considered subgraphs are just the ones in the immediate neighborhood of the nodes (where, in the multiplex case, neighborhoods are defined as Non-Inclusive Multiplex Neighborhoods, $\mathcal{N}_{\subset}$, according to the definition in Section 4.1.1), thus we are excluding possible motifs that go beyond the 1-hop neighborhood of the central node.

Since we compute the patterns in the network formed by the neighbors of each node, the central node will always be present in the resulting motifs,

since it has incident edges to all its neighbors. For these reasons, a classical motif analysis of the network may lead to different results.

The neighborhood motifs analysis can lead us to a deeper knowledge of neighborhood structure and the relation of real-world networks with random ones. Since we are interested in the connections between the neighbors of a node for the motifs analysis, we need to choose a null model having neighborhoods that are comparable to the datasets, for instance with the same number of nodes in each neighborhood. For this reason, we chose the configuration model (introduced in Section 2.5.4) as a null model, since its degree sequence is the same as in the original network. The variant of the configuration model we adopt takes also into account the overlapping degree of each node, which is the number of edges incident to each node that are shared between multiple layers. With this choice, from the analysis, the differences in the structure of neighborhoods, caused by the presence or absence of links between pairs of neighbors, would emerge, instead of differences causes by another degree distribution. We compute both over-represented and under-represented patterns (sometimes called *anti-motifs*) in the social network datasets compared to the random networks. For multiplex networks, we compute the subgraphs according to both node and node-layer isomorphism, where the node-layer isomorphism is applied directly to the subgraph, and not to the neighborhood it is extracted from.

Additionally, neighborhood motifs, in the context of social networks, can help to understand how the social circle of the represented entities is organized. Mainly, we could understand if the network is mostly locally tree-like (the neighbors of a node are not connected to each other) or dense (we can observe transitivity between neighbors). For instance, the presence of densely connected nodes means that people hang out in groups (or at least that the central nodes are part of a bigger company of friends). Moreover, the multiplex representation allows us to distinguish the behaviour of a node in different social contexts or, in the case of temporal networks, at different timestamps.

7.1 Methods

We now present the methods we use for the neighborhood motifs analysis. We start by explaining, in subsection 7.1.1, the two different ways we count subgraphs in the neighborhoods (subsection 7.1.1), to check whether the neighborhood size has an impact in the discovery of significant motifs; we then explain the sampling method we adopt in presence of bigger neigh-

borhoods (subsection 7.1.2), where it is computationally difficult to extract all the possible subgraphs and, eventually, we show how we estimate the sampling error (subsection 7.1.3).

7.1.1 Neighborhood motifs count and proportion

In this section, we explain the methods used to conduct the neighborhood motifs analysis. We count the induced subgraphs that are present in each neighborhood of each social network, and then compare the count to the one in the realization of the null model, by computing the Z-Score (defined Section 2.6). We count the motifs in each neighborhood in two different ways, depending on whether we want to consider the neighborhoods equal to each other or we want to normalize the count based on the neighborhood size. Indeed, bigger neighborhoods can have more subgraphs occurring many times, and they can contribute more towards the count of motifs, while we are interested in the distribution of motifs across all the neighborhoods. The methods we use to count the subgraphs in the neighborhoods are the following:

1. *Frequency*, (F): simple count of the subgraphs. Each subgraph counts the same regardless the size of the neighborhood it is present in. We indicate the frequency of a subgraph G in a neighborhood of a node v as $F_{G(\mathcal{N}_{\subset}(v))}$, and in the whole network as F_G. The total count F_G of a specific subgraph G in the whole network is the sum of the frequency of G in all the neighborhoods:

$$F_G = \sum_{v \in V} F_{G(\mathcal{N}_{\subset}(v))}\,, \tag{7.1}$$

   where V is the vertex set of the network;

2. *Proportion*, (P): fraction of the induced subgraphs of a certain size present in a neighborhood out of the possible induced subgraphs of the same size in that neighborhood. The number of possible induced subgraphs is the binomial coefficient of the neighborhood size over the subgraph (or motif) size (in fact, the number of possible motifs does not grow linearly with the neighborhood size, but much faster, with the binomial coefficient). We indicate the proportion P of a subgraph G in a neighborhood $\mathcal{N}_{\subset}(v)$ of a node v as:

$$P_{G(\mathcal{N}_{\subset}(v))} = \frac{F_{G(\mathcal{N}_{\subset}(v))}}{\binom{|\mathcal{N}_{\subset}(v)|}{|G|}}\,, \tag{7.2}$$

where $|\mathcal{N}_{\subset}(v)|$ is the size of the neighborhood of v (equal to the degree of v) and $|G|$ is the size of the subgraph G. To count each subgraph in the whole network, we simply sum the proportion P_G obtained from each neighborhood:

$$P_G = \sum_{v \in V} P_{G(\mathcal{N}_{\subset}(v))}\,, \tag{7.3}$$

As mentioned at the beginning of this chapter, the null model we chose is the configuration model. Since we are analyzing neighborhoods, we want to evaluate the connection patterns between the neighbors of a node, thus we want to compare those with a randomized version with the same neighborhood sizes. In fact, in the configuration model, the degree sequence is the same as the original network, thus the number of nodes with a given neighborhood size are the same.

In our analysis, we focus on motifs of size (i.e. number of nodes) from 2 to 4 (included). This is because the computational time for extracting and computing the isomorphism classes of bigger graphs increases quickly, especially in the multiplex case. Furthermore, 4 is just a bit less than the average degree of most of the considered datasets (as can be seen from Tables 6.1 and 6.2), thus these small motifs could already give a good idea of how the social circles of nodes is organized and what are the patterns forming neighborhoods.

7.1.2 Counting and sampling subgraphs

We now illustrate how we count the subgraphs in the full network and how, in presence of bigger neighborhoods, we estimate that count by sampling some subgraphs out of all the possible ones.

To conduct the analysis, we first go through all the nodes in the network and extract the non-inclusive neighborhood $\mathcal{N}_{\subset}$ of each of them. If the neighborhood size is less or equal than 10, we go through all the combinations of nodes in the neighborhood (where the number of nodes in the combinations varies from 2 to 4, as the size of the motifs we are interested in), and compute the induced subgraphs given by them. Otherwise, if the neighborhood size is higher than 10, we conduct uniform sampling without replacement from the possible combinations of nodes in the neighborhood. We replace the single nodes, but not the full combination, thus a single node can still be sampled multiple times as part of other induced subgraphs.

As mentioned above in subsection 7.1.1, the number of all the possible subgraph of a certain size is given by the binomial coefficient of the number of nodes in the neighborhood over the subgraph size ($\binom{|\mathcal{N}_{\subset}(v)|}{|G|}$). After we

have sampled the subgraphs, we need to estimate the real count. Indicating with s the number of sample (sample size), and with S_G the sampled count of the graph G, we can compute the estimated frequency $\hat{F_G}$ by multiplying the sample proportion of G for the binomial coefficient, which is the total number of possible combinations, or, in statistical terms, the population size[1]:

$$\hat{F_G} = \frac{S_G}{s} \times \binom{|\mathcal{N}_{\subset}(v)|}{|G|}. \tag{7.4}$$

Once we have the frequency (or estimated frequency) for each subgraph in each neighborhood, we can also compute its proportion, according to Equation 7.2 (when we sample nodes, we need to substitute F_G with $\hat{F_G}$, and we indicate the estimated P_G with $\hat{P_G}$). Finally, we can compute the total Frequency or Proportion of each subgraph in the network, according to Equations 7.1 and 7.3.

To obtain the Z-Score for each subgraph, we then need to do the same calculation mentioned above with all the realization of the random networks, thus compute F_G and P_G (or $\hat{F_G}$ and $\hat{P_G}$) for each subgraph in the network, and the corresponding μ and standard deviation σ.

The number of realization of the null model influences the Z-Score. Indeed, each subgraph has a certain value of mean and variance associated and, with more realizations, we can estimate better the real values. For the *CopNet* dataset, we do 100 realization of the null model for each considered network, while for *CountryCalls*, we do 100 realization. The reason of having less realization in *CountryCalls* compared to *CopNet* is that *CountryCalls* is significantly bigger, and big systems tend to self-averaging (moreover, what counts to obtain significant results is also the number of data points, which in *CountryCalls* is much bigger than in *CopNet*). The amount of possible neighborhoods (and consequently the possible patterns in them) that appear in big networks is higher than in small ones. In small networks, we most likely need to do more realization of the null model to observe the subgraphs that are present in the original network.
When we conduct sampling in multiplex networks, we sample combination of nodes, and not of node-layer tuples. Thus, if a node is present in multiple layers it is still considered a single node during the sampling process. We chose the following sample size s, depending on the size of each subgraphs

[1]This is a classical way to estimate the count when we have a sampled proportion. For example, if we want to estimate how many nodes of different types there are in a population of 1000 nodes, and we sample 40 nodes of one type from a population with a sample size equal to 100 nodes, the sample proportion of nodes of that type is $\frac{40}{100} = 0.4$. To estimate the number of nodes of that type in the real population of 1000 nodes, we compute $\frac{40}{100} \times 1000$.

G_i (in fact, the bigger the subgraph size, the higher the number of possible combination of nodes is):

- $s = 50$, for $|G| = 2$;
- $s = 150$, for $|G| = 3$;
- $s = 300$, for $|G| = 4$.

We chose those values because they are slightly higher than the number of possible combinations with a neighborhood size equal to 10 (size under and with which we do not conduct sampling). In the next subsection 7.1.3 we discuss how we estimate the error due to sampling.

7.1.3 Sampling error estimation

The number of samples affects the count estimation, and we should choose it in order to minimize the error of the estimation. A way to estimate the error is computing the standard error of a percentage, given the number of samples n. This sampling error estimation method assumes:

- random sampling (as we do);
- the distribution where the samples come from is a normal distribution (as it also assumed for the computation of the Z-Score);
- the number of samples is small compared to the population size with large neighborhoods. This is our case, since the number of nodes' combinations grows quickly with the neighborhood size.

We now show how we estimate the sampling error of each subgraph for each neighborhood and, at the end, for the entire network by computing the standard error.

In our case, the percentage p_G is the sampled percentage of a subgraph G with a specific size t, relative to the total amount of subgraphs of the same size t sampled from a neighborhood. We indicate the number of samples from a neighborhood as s. The standard error SE_G of a subgraph G is then estimated as:

$$SE_G = \sqrt{\frac{p_G(1 - p_G)}{s}}\,. \tag{7.5}$$

The number of samples s is at the denominator in Equation 7.5, thus increasing it would lead to a lower error, thus to a better estimation of the actual Frequency (and consequently, the Proportion) of a particular subgraph in

the network's neighborhoods.

With Equation 7.5, we compute the standard error for the percentage of a specific graph G in a single neighborhood. However, we conduct the sampling process on all the neighborhoods that are bigger than a certain size (in our case, 10) in the whole network. Thus, we want to compute the standard error S_G for the percentage of G in the network. Since we conduct sampling just on neighborhoods bigger than 10, we are actually estimating the percentage of G for those neighborhoods, and we indicate it as $\hat{p'_G}$. The neighborhoods we are not sampling from do not contribute towards the sampling error.

To estimate the sampling error in the whole network we also need the total number of samples, which is the number of samples for each neighborhood, s, multiplied by the number of neighborhoods we sample from. The standard error $S\hat{E}_G$ for a graph G in the full network becomes:

$$S\hat{E}_G = \sqrt{\frac{\hat{p'_G}(1-\hat{p'_G})}{s\ N_2}}\,. \tag{7.6}$$

The concept of standard error is connected with the one of margin of error. The margin of error is commonly used for estimating the error of random sampling in surveys' results. The margin of error can be computed as amount of standard errors, depending on the size of the confidence interval we want to obtain. A margin of one standard error corresponds to a confidence interval of 68%, according to the *z-table* of the normal distribution ($z - value = 1$). Similarly, 95% confidence interval corresponds to 1.96 standard error, while 99% confidence interval corresponds to 2.58 standard error.

In our experiment, we obtained Standard Error values that are at maximum around 1%, even computing the margin of error for a confidence level of 99%. We consider those values negligible, thus we do not discuss them further.

There exists alternative ways to estimate the error. For instance, we could simply repeat the sampling on the same graphs multiple times for the same number of sample and then compute some measures like Coefficient of Variation c_v, which is the ratio of the standard deviation σ of a population over the mean μ ($c_v = \frac{\sigma}{\mu}$), or the error of the mean. However, this would require additional sampling, and we could use the same computational power to sample multiple times from the same network, which would lead to a better estimation of the actual occurrence of a subgraph in the network.

7.2 Results

In this Section, we illustrate the results of the neighborhood motif analysis obtained with the methodology explained in Section 7.1. We start by comparing the distribution of the occurrence frequency of the subgraphs' classes in both the analyzed datasets and the corresponding null model, similarly to what we have done for the neighborhood's occurrence frequency in Chapter 6.2 (Figure 6.2). The number of times each class occurs in the network is related to the uniqueness, since, the more times the class occurs, the more likely the neighborhoods would be similar to each other, allowing an easier anonymization process. On the other hand, if there would be subgraphs occurring few times in the network, a high amount of neighborhoods would belong to different isomorphism classes, increasing the uniqueness value and making the anonymization problem harder. We then show the Z-Scores of the subgraphs extracted form the analyzed datasets, both in the monoplex and multiplex case, with the two counting methods introduced before in subsection 7.1.1.

7.2.1 Class distribution

As we have seen in Section 6.2, in real-world data (and, in particular in the analyzed datasets, such as *CopNet* and *CountryCalls*) most of the neighborhoods' classes occur just one time, while the amount of classes appearing more times becomes lower with the increasing occurrence frequency. This characteristic is also reflected in the degree distribution of real-world network, whose trend is similar to the one of a power law (introduced Section 2.1). This means that the neighborhood classes, like the degree values, are not all occurring the same amount of times, thus their distribution is not completely random. Indeed, if each class had the same probability to occur, the distribution would be uniform, and the plot would look like a flat line. We can observe a similar behaviour in the distribution of the occurrence frequency of subgraphs' classes in neighborhoods, reported in Figure 7.1 for the datasets *CopNet* and *CountryCalls*, and the mean of the respective null models. We report all the possible monoplex networks (Calls, Sms and aggregation of both Calls and Sms) and multiplex networks (where one layer is Calls and the other Sms), with respect to both node and node-layer isomorphism. Figures 7.1 *a* and *c* show the occurrence frequency of subgraphs' classes with respect to the Frequency of each subgraph (F), while Figures 7.1 *b* and *d* are with respect to the Proportion (P) (defined

before in subsection 7.1.1). Since the proportion P is the frequency scaled also in function of the neighborhood size, some of the subgraph can have occurrence value less than one. The occurrence frequency for a subgraph could be less than one also if, for instance, a particular subgraph did not occur in all the realization of the null model.

The number of multiplex subgraphs is much higher than the monoplex ones, which, for sizes from 2 to 4, are just 17, as shown in Figure 7.2. For this reason, the curves in Figures 7.1 c and d, representing the multiplex classes, are much more dense than the curves in Figures 7.1 a and b, which represents the monoplex classes.

Since the considered datasets have a significant difference in size, it is normal that the occurrences of all the subgraph classes in them are higher in *CountryCalls* (which has around five millions of nodes) compared to *CopNet* (which has around eight-hundred nodes). In the monoplex case, the higher amount of occurrences is in the aggregated networks. This is especially true for *CountryCalls*, while for *CopNet* the classes in Calls and Sms also have almost the same occurrences in the aggregated network. In fact, the Calls and Sms layers in *CopNet* are similar to each other, as can also be noticed from Table 6.1, or from the degree distribution in Figure 6.1. Moreover, the amount of overlapping edges between the Calls and Sms networks in *CopNet* is higher than in *CountryCalls*. Since less edges are shared, the aggregation would lead to the formation of diverse classes when aggregated.

In Figure 7.1 c and d, we can notice that, in the null models of *CountryCalls*, there are less subgraph classes than in the dataset. This is due to the fact that the configuration model is locally tree-like and its clustering coefficient goes down as the size increases. Another interesting difference between the original data and the corresponding null models is that the distributions in the null models are more "flat". This is particularly noticeable in *CountryCalls*, where the classes are also mainly concentrated in two specific areas in the frequency: mostly between 10 and 10^5, and after 10^8 (besides the classes that occurs zero times). The diversity of the classes distribution is again explained by the higher density of the datasets compared to the configuration model.

7.2.2 Resulting motifs

In this section we present the results of the neighborhood motif analysis on the considered datasets. We first illustrate the analysis on the single-layer and aggregated networks (subsection 7.2.2.1), and then on the multiplex ones (subsection 7.2.2.2). We also discuss some limitation of this analysis

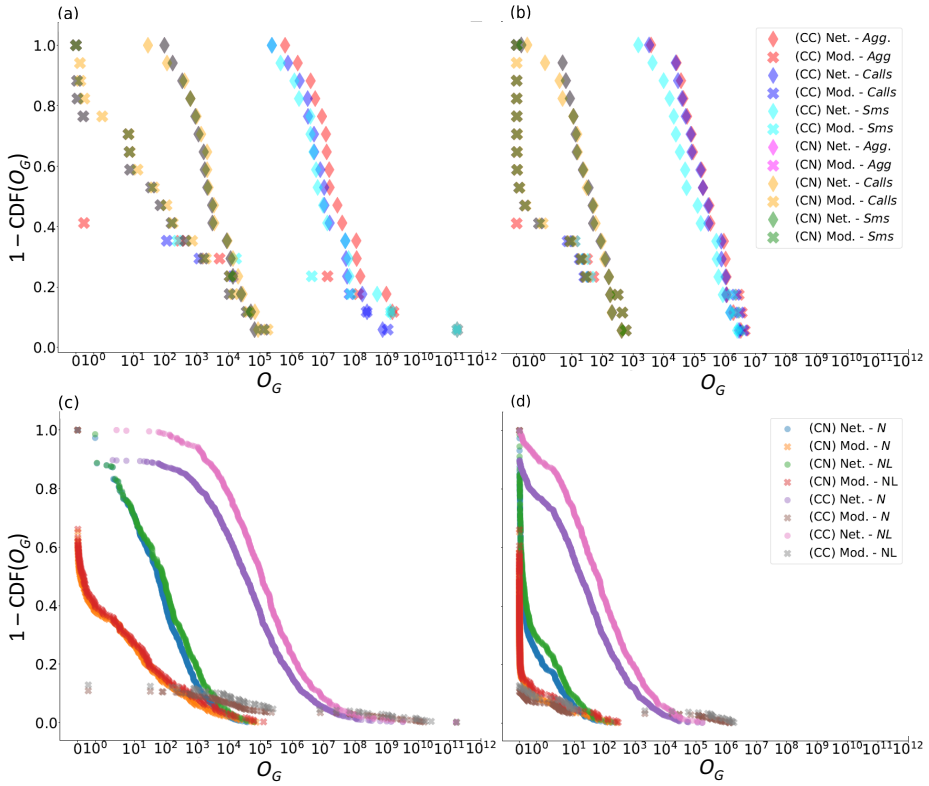

Figure 7.1: Occurrence frequency O_G of monoplex and multiplex subgraph classes (with size from 2 to 4) in data and models with respect to both Frequency F and Proportion P: a - F of monoplex subgraphs; b - P of monoplex subgraphs; c - F of multiplex subgraphs; d - P of multiplex subgraphs. When specified in the legend, N stands for multiplex network with neighborhood computed with respect to node isomorphism; NL for node-layer isomorphism; $Agg.$ for aggregated network. CC stands for *CountryCalls*, CN for *CopNet*. The analyzed datasets are indicated with $Net.$, and the models with $Mod.$.

and possible issues we have identified in the metric used to compare networks and compute the significance level of the subgraphs.

7.2.2.1 Monoplex motifs

The monoplex networks we consider are, in both datasets (*CopNet* and *CountryCalls*), the ones regarding calls, sms, and the corresponding aggregated network. All the possible monoplex motifs with size from 2 to 4 are illustrated in Figure 7.2.

Figure 7.3 shows the Z-Score of the monoplex motifs in the analyzed networks. From this figure, we can see that both Call, Sms and the aggregated network present similar patterns in the neighborhoods. To compute the Z-Score, the motif should appear in at least one of the realization of the null model. If a particular subgraph is present in the original network but has not appeared in the null model, then its estimated Z-Score goes to infinity, since its mean and standard deviation are zero. This means that subgraph is extremely rare in the null model, and, if it appeared in some of the realizations, we were not able to see it with our sampling. In this case, an higher amount of sampling could have helped.

All the motifs illustrated in Figure 7.2 appears in our datasets. However, the more dense ones did not appear in the random networks corresponding to *CountryCalls*. This is expected since the density of the configuration model decreases when the network size grows [New18]. Indeed, with a fixed degree sequence and a larger number of nodes, there are more possibilities of picking neighbors, with a less probability of formation of dense groups. Since *CopNet* is a smaller dataset, we could compute a Z-Score value for more motifs. The only one not appearing in the null model of all the networks is the one with 4 nodes completely connected (graph Q in Figure 7.2), while the graph with 4 nodes and 5 edges (graph P in Figure 7.2) is not appearing in the configuration model corresponding to Sms. Those are two most dense motifs, thus it is also normal to not to see them in the random networks' realizations.

There are no remarkable discrepancies in the motifs distribution computed with the Frequency F or the Proportion P_G. In *CopNet*, with the Proportion P, we can see an higher Z-Score of the most dense motif P, and a lower Z-Score of G. Overall, we have observed a lower variance when considering the proportion P, and this is the reason why the absolute values of the Z-Score are sometimes higher. In the Z-Score of the Frequencies F of *CopNet* (Figure 7.3 c), some of the Sms' and Calls' motifs have Z-Score shrinked towards 0 compared to the corresponding proportion P. Eliminating the effect due to the neighborhood size by computing the proportion P, makes an actual difference then, uniforming more the neighborhoods an also the motifs count.

The under-represented motifs (or anti-motifs) that emerged are the same for both datasets and are, as expected, the ones without edges between the neighbors (A, C, G in Figure 7.2).

The computed values of Z-Score are all very high. There is a huge gap between the values of *CopNet* and *CountryCalls*. Being *CountryCalls* significantly bigger than *CopNet*, its Z-Scores are noticeably high. As discussed in Section 2.6, it is known that the network size influences the Z-

Score. However, computing the Significance Profile to compare the networks, in our case, did not give easily interpretable results. In fact, for example, in the aggregated network of *CopNet*, the SP of subgraph J resulted to be ≈ 0.99, thus very close to the maximum value 1, while the other motifs had a SP with absolute value less than 0.005. Even though the subgraph J has a Z-Score significantly higher compared to the others, values of $Z - Score$ around 10^5 and 10^6 cannot be ignored, since they describe a characterizing pattern of the networks, occurring a high amount of times.

Motif J is composed of two edges connecting two neighbors not communicating to each other. J is also much more significant in *CountryCalls* compared to *CopNet*. This may be due to the fact that the networks of calls and sms in *CountryCalls* have a relatively low edge overlap, thus the aggregation leads to the appearance of non-communicating patterns in the neighborhoods. In the context of calls and sms, this may mean that, in that dataset, two people communicating through calls often do not use sms.

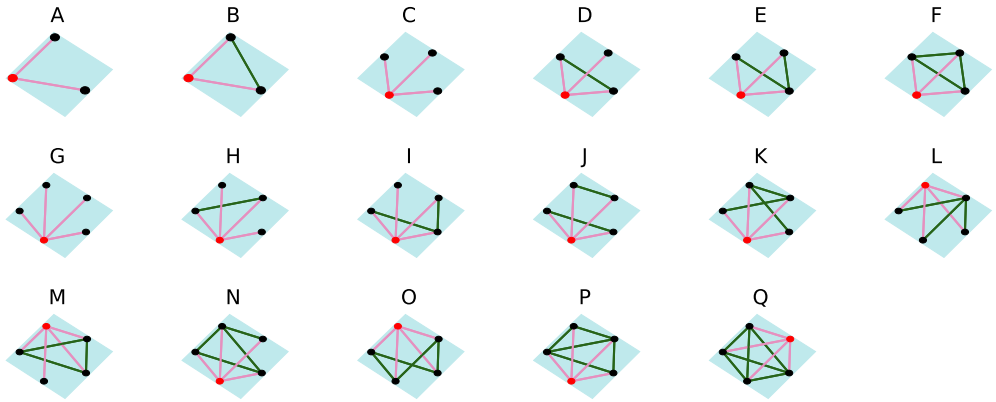

Figure 7.2: List of all the monoplex neighborhoods motifs with size from 2 to 4. The red node is the central node and the pink edges link the central node to the other nodes (in black) which are the neighbors. Thus, there are always edges between the central node and the other nodes. The green edges represent the edges present between the neighbors.

7.2.2.2 Multiplex motifs

We now discuss the results of the multiplex neighborhood motifs analysis with respect to both node and node-layer isomorphism.

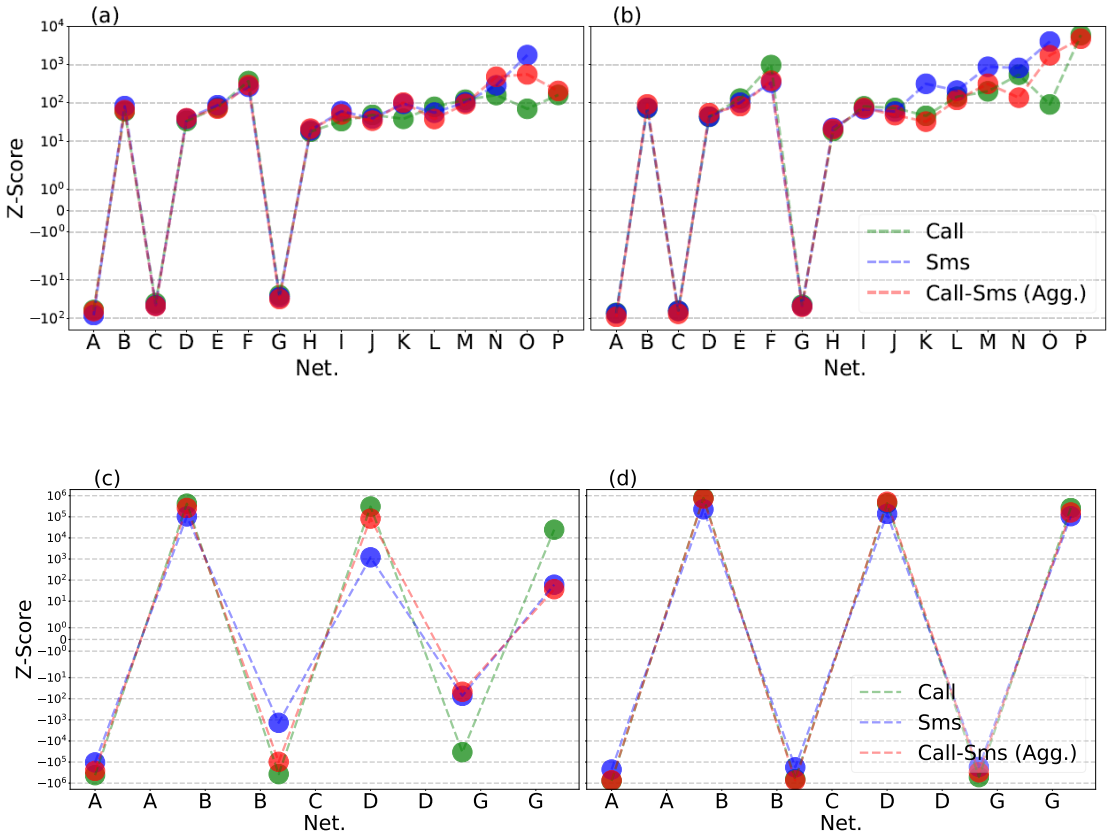

Figure 7.3: Z-Score of the monoplex motifs in the datasets *CopNet* and *CountryCalls* with respect to Frequency F and Proportion P: a) *CopNet* - Frequency; b) *CopNet* - Proportion; c) *CountryCalls* - Frequency; d) *CountryCalls* - Proportion. Each tick on the horizontal axis corresponds to a motif. The corresponding motifs are reported in Figure 7.2. The legend is the same for all the figures and is reported in *b* and *d*.

Figures 7.6 shows the Z-Score for the multiplex motifs appearing in both the original network and in the respective random ones. Similarly to what we have seen for monoplex motifs, we were not able to sample from the random networks all the subgraphs existing in the networks. Since the number of multiplex classes is much higher than the monoplex ones, the difference between the amount of discovered motifs in the datasets and in the random networks is significant.

Table 7.1 shows the number of discovered multiplex motifs for both the datasets and the respective configuration models. As expected, there is a

difference in the amount of classes between node and node-layer isomorphism. Since with node-layer isomorphism layers' labels are interchangeable, it is normal that the number of motifs with this type of isomorphism is almost half than the ones with respect to node-isomorphism.

There is also a lower amount of classes in the configuration model of *CountryCalls* than in *CopNet*. As in the monoplex case, it is easier to discover more classes in *CopNet*, since this dataset is significantly smaller than *CountryCalls*, in which we are able to sample only the most common subgraphs. Moreover, the configuration model of bigger networks is less dense than in smaller ones. For the same reason, the motifs that we have not sampled from the null models are the most dense ones. The presence of denser motifs can also be noticed in Figure 7.6, where dense motifs have a high Z-Score. Some of the motifs with a high Z-Score are shown in Figures 7.4 *a* and *b*. Even here, the motif with the maximum Z-Score of *CountryCalls* has less edges than the one of *CopNet*, as a confirmation of the higher density of the latter. Figures 7.4 *c* and *d* show instead the motifs with the minimum Z-Score from both datasets. As mentioned above, those subgraphs are among the most common in the configuration model and do not have any edge between the nodes, highlighting the sparse nature of this model.

The motifs for which we could compute Z-Score have in general a low amount of edges, while the ones not appearing in the null models have a lot

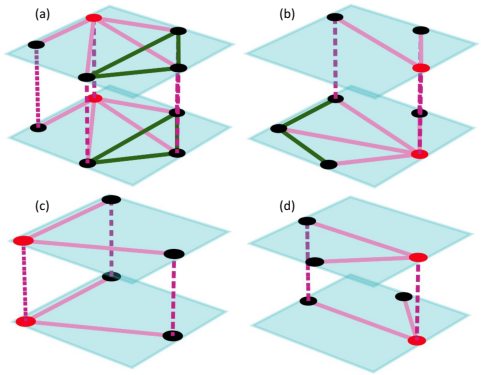

Figure 7.4: Motifs with the maximum estimated finite Z-Score in *CopNet* (a), and in *CountryCalls* (b), and with the minimum estimated finite Z-Score in *CopNet* (c), and in *CountryCalls* (d) The red node is the central node and the pink edges link the central node to the other nodes (in black), that are the neighbors. The green edges represent the edges present between the neighbors.

of edges. Figure 7.5 shows some of these dense subgraphs, which have an estimated infinite Z-Score.

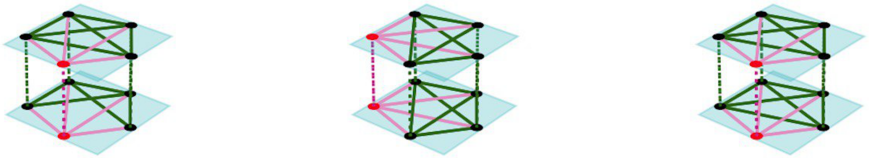

Figure 7.5: Some of the dense multiplex subgraphs that do not appear in the null models. The red node is the central node and the pink edges link the central node to the other nodes (in black), that are the neighbors. The green edges represent the edges present between the neighbors.

As can be noticed in Table 7.1, there are more subgraphs in *CountryCalls* compared to *CopNet*. This is due to the difference in size of the datasets. Given the variety of multiplex motifs, there exists also some with edges or nodes just in one layer. From Figure 7.6, we can notice there are only few differences in the distribution of classes with respect to the Frequency F and the Proportion P. Overall, especially in *CountryCalls*, the absolute values of the Z-Score with respect to P are higher than the ones with respect to F.

A lot of multiplex motifs resulted to have a Significance Profile lower than 10^{-3}, while having an high Z-Score. As we have briefly mentioned for one of the monoplex motifs in subsection 7.2.2.1, subgraphs that have a Z-Score, for example, higher than one thousands should be considered as over-represented motifs. Indeed, in this case, the difference between their frequency in the original network and the mean time of appearence in the null model (or viceversa) is very high, and they cannot be ignored if our aim is to study recurrent patterns.

The issue of not having reliable values of Significance Profile generally emerges in large and dense networks, where the subgraphs occurring in the datasets are very different in comparison with the null model. With multiplex neighborhoods the problem is even more evident, since the number of possible subgraphs is significantly higher than the monoplex ones, and rescaling to length one the vector of the Z-Score values could not be the best way to compare networks of different size. Moreover, computing patterns in neighborhoods leads to extremely high Z-Score values in both the monoplex and multiplex case, further highlighting a possible issue related to the estimation of this values. This can be related the use of Z-Score itself, which assumes a normal distribution of the subgraphs, which may not be the case when it comes to networks' neighborhoods.

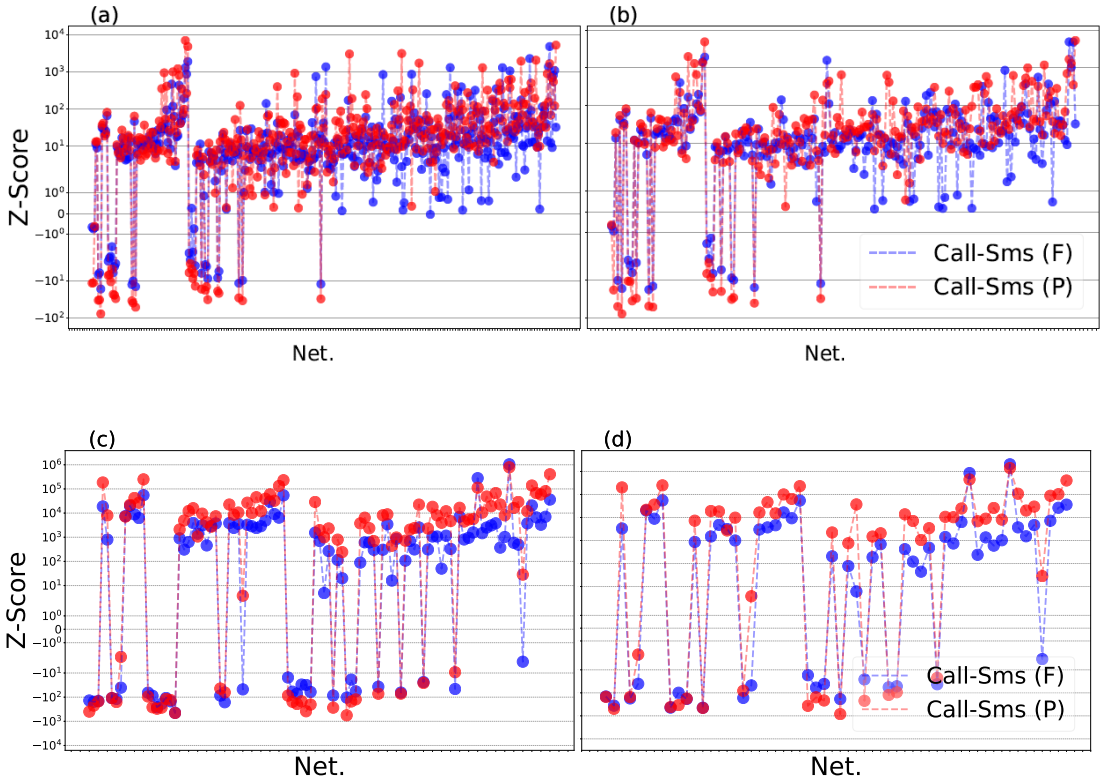

Figure 7.6: Z-Score of the multiplex motifs in the datasets *CopNet* and *CountryCalls* with respect to Frequency F (in red) and Proportion P (in blue), and node and node-layer isomorphism: a) *CopNet* - Node-isomorphism; b) *CopNet* - Node-layer isomorphism; c) *CountryCalls* - Node-isomorphism; d) *CountryCalls* - Node-layer isomorphism. The motifs are ordered by number of nodes and edges. Each tick on the horizontal axis corresponds to a motif. The legend is the same for all the figures and is reported in b and d.

	CopNet (Node)	**CopNet (Node-layer)**	**CountryCalls (Node)**	**CountryCalls (Node-layer)**
Data	582	343	847	450
Model	374	225	103	58

Table 7.1: Number of discovered multiplex subgraphs in the original datasets and in the respective null models, with respect to node and node-layer isomorphism.

7.3 Alternative approaches to network neighborhood analysis

In this chapter, we have analyzed the structure of neighborhoods in social networks, through the discovery of the motifs forming them. To find

neighborhood motifs we have performed an exhaustive search of induced subgraphs on the network of each neighborhood, when their size was limited, or we have uniformly sampled combination of nodes in the neighborhoods, computing the induced subgraphs composed of those vertices. However, different motifs finding (or subgraphs counting) algorithms exist [WR06; MSK12], and they can be used as an alternative even for this kind of analysis.

Since each random graph has a certain probability to occur, also the subgraph in it are. For this reason, instead of sampling from the null model, we could have estimated the probability for a certain subgraph to occur in the random network. In this way, we could have obtained an estimated Z-Score value for each subgraph occurring in the original network.

We could have also used different null models for our analysis, such the configuration model without edge overlap or also the *Erdős-Rényi* model. In particular, the latter could have been compared to the configuration model to understand the effect of the degree sequence when discovering neighborhood motifs. Another possible model would have been one in which we randomly shuffle the edges between a neighborhood. In this case, we could observe patterns in neighborhoods and the density of the original networks compared to the null model would not play a role.

Although the motifs analysis is mainly applied to monoplex networks, there are also few studies concerning multilayer networks [Bat+17; Tak+18]. Most of the methods in the literature are based on different kinds of sampling and can be useful in our case, especially in presence of large neighborhoods. Moreover, we have taken into account networks with unlabelled nodes and undirected edges. The amount of possible motifs increases when the edges have directions and the nodes are colored. Methods like [WR06] and [Kas+04] have been developed for the discovery of such motifs.

However, the conducted analysis is a simple method for the discovery of the over-represented and the under-represented subgraphs in neighborhoods, and it is easily extendable to higher-order neighborhoods, such as 2-hops or more. Furthermore, we could extend the analysis to labelled data, computing the neighborhood motifs for nodes with different features, for instance males and females, or students studying different majors. To do that, we could extract the label of the central nodes, and, in the case of genders, we could conduct a chi-square test with bootstrapping, where the null hypothesis is that each group has the same neighborhood motifs. This would mean drawing sample with replacement, in order to overcome the difference in the frequency of the groups. We could then compute the p-values for each relevant induced subgraph obtained. [Psy+17] contains an analysis of the role of gender using the *CopNet* dataset, also using motifs to study whether men and women tend to form triangles with nodes of the same gender group.

In a large-enough network, we could slightly modify our approach, in order to discover the actual over and the under-represented neighborhoods (instead of the patterns in them): we could treat the neighborhoods themselves as motifs and compute the Z-score and SP of those.

[SP09] analyzes the structure of neighborhoods of a mobile phone call dataset with a methodology very similar to ours, though considering only monoplex networks. The authors extract the neighborhoods of every node and compute the induced subgraphs present in them, focusing though only on the connected ones, instead of considering also the non-connected motifs as we do. This work also analyzes the different roles of each nodes in the neighborhood based on their position. However, it does not perform a proper motif analysis, since the found patterns are not compared with a null model, and, consequently, metrics like Z-score and SP are not considered for assessing the importance of the subgraphs, but only the frequency is used.

To conclude, some studies approached the analysis of neighborhoods from different angles. For example, [PS10] analyzes network neighborhoods of different orders based on the number of nodes, number of edges and density (the number of edges out of all the possible edges) to detect anomalies in a dynamic network, also comparing also the nodes to each other across different neighborhoods orders. In our case, if we want to compare single nodes to each other, we can extend our analysis by exploiting the multiplexity to see how the neighborhoods of single nodes change across different layers.

Chapter 8

Neighborhood Anonymization in Multiplex Networks

In this chapter, we are going to present an algorithm anonymizing a pair of multiplex neighborhoods by adding nodes and edges. This method is the core part for a full anonymization algorithm to prevent neighborhood attack on multiplex networks. We also discuss the features of multiplex networks that should be taken into account when developing an algorithm for this kind of systems, either to prevent neighborhood attacks or mitigate similar risks.

8.1 Neighborhoods pair anonymization

In this section, we illustrate an algorithm to make two multiplex neighborhoods with two layers isomorphic to each other, according to node isomorphism. A method similar to the one we present is necessary to build any anonymization algorithm, since two or more neighborhoods need to be modified to belong to the same isomorphism class. The anonymization algorithms presented in 3 first group together similar neighborhoods, and then make the neighborhoods in the same group isomorphic to each other. If we consider *k-anonymity* with $k = 2$ we just need to suppress unique neighborhoods, thus each group would contain only two neighborhoods. In this case, the algorithm we present would be enough to anonymize them. Otherwise, with $k > 2$, more neighborhoods need to be anonymized, but this algorithm would still be useful since neighborhoods can be made anonymous in pairs.

To minimize the number of modifications, the neighborhoods need to be aligned to find similar nodes and edges that are already present. Our

method contains a heuristic to align two multiplex neighborhoods based on matching maximal connected components, inspired to the seminal paper [ZP08], presented in 3. The connected components are extracted from the aggregated version of the multiplex network, and then sorted by taking into account multiplex features. After that, the matched connected components are anonymized by computing the difference of their adjacency matrices and consequently adding the missing edges.

A pseudocode version of the proposed algorithm is shown in Algorithm 1. Once the connected components are extracted from the neighborhoods, they are sorted based on various features. The idea is to anonymize first the bigger components, then the ones with a higher amount of nodes and edges. For this reason, we sort the components according to the following order: number of node-layer tuples, overlapping nodes (nodes shared between the layers), nodes in layer one, nodes in layer two, number of total intra-layer edges, number of overlapping edges, number of edges in layer one, number of edges in layer two.

Algorithm 1 Neighborhood pair anonymization

Input: multiplex neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_2$
Output: anonymized $\mathcal{N}_1$ and $\mathcal{N}_2$

Extract neighborhood components from $\mathcal{N}_1$ and $\mathcal{N}_2$;
Sort neighborhood component from $\mathcal{N}_1$ and $\mathcal{N}_2$;
while $\mathcal{N}_1 \,!\cong_0 \mathcal{N}_2$ **do**
    $Comp_1 \leftarrow$ next component of $\mathcal{N}_1$
    $Comp_2 \leftarrow$ next component of $\mathcal{N}_2$
    if $Comp_1 \cong_0 Comp_2$ **then**
        continue
    end if
    if $Comp_1 \cong_{agg} Comp_2$ **then**
        make $Comp_1$ and $Comp_2$ multiplex isomorphic
    else
        Sort nodes of $Comp_1$ and $Comp_2$
        Add missing nodes to $Comp_1$ or $Comp_2$
        Compute adjacency matrix difference $|A|_{Comp_1} - |A|_{Comp_2}$
        Add missing edges to $Comp_1$ and $Comp_2$
    end if
end while

We then match the sorted connected components from each network together, and anonymize them if they are not already isomorphic. However, if they are not isomorphic in the multiplex version (indicated with $\cong_0$),

but they are isomorphic in the aggregated network (indicated with $\cong_{agg}$), the anonymization consists only in adding the missing edges in the correct layers.

Otherwise, we anonymize the connected components by aligning the nodes and computing the difference of the adjacency matrices, to locate the missing edges (as it is done in methods presented in 3, such as [TP10] and [Liu+15]). To align the nodes of each connected component, we sort the vertices in descending order according to: degree in the aggregated network, overlap degree distribution, degree distribution in layer one, degree distribution in layer two. If necessary, we add nodes to the neighborhoods if one of them has less nodes than the other. Instead, if one of the components has less edges than the other, but the number of nodes in the neighborhoods is already the same, we do not add new nodes but merge the connected component with another existing node in the neighborhood. The algorithm continues to match connected components and anonymize them until the two multiplex neighborhoods are isomorphic.

Figure 8.1 shows two pairs of multiplex neighborhoods that have been anonymized with Algorithm 1. In Figure 8.1 *a*, the two neighborhoods differ of just one edge in layer 2. In this case, the algorithm just add the missing edge without any further modification (Figure 8.1 *b*). The algorithm works also not trivial examples: for instance, in Figure 8.1 *c* the two neighborhoods have a different amount of nodes, and the algorithm adds the missing nodes to the smaller neighborhood; In Figure 8.1 *d* there are nodes missing from one of the layers in both the neighborhoods. In this case, the algorithm adds the nodes to both neighborhoods and, when possible, adds a node-layer tuple aligned to an already existing node.

The presented algorithm anonymizes a pair of neighborhoods with respect to node isomorphism. To make them node-layer isomorphic, we can run the algorithm two times switching the layer's label in one of them, and then keep the anonymized version of neighborhoods pair with less modifications (in terms of added nodes and edges) from the original.

Since this algorithm is based on matching connected components in the aggregated network, it can also work in monoplex networks. The only modification needed is in the matching of the components and nodes in the component. Indeed, we do not have to sort by measures typical of multiplex networks such as number of nodes or edges in the layers.

Anonymization methods presented in 3 define a cost function to measure the number of modifications applied to the network. This function is of type $\alpha x + \beta y$, where x and y are the amount of modification, for instance in terms of nodes and edges, and α and β are the weight that the user can put on each type of modification. In our case, we can add even more variables

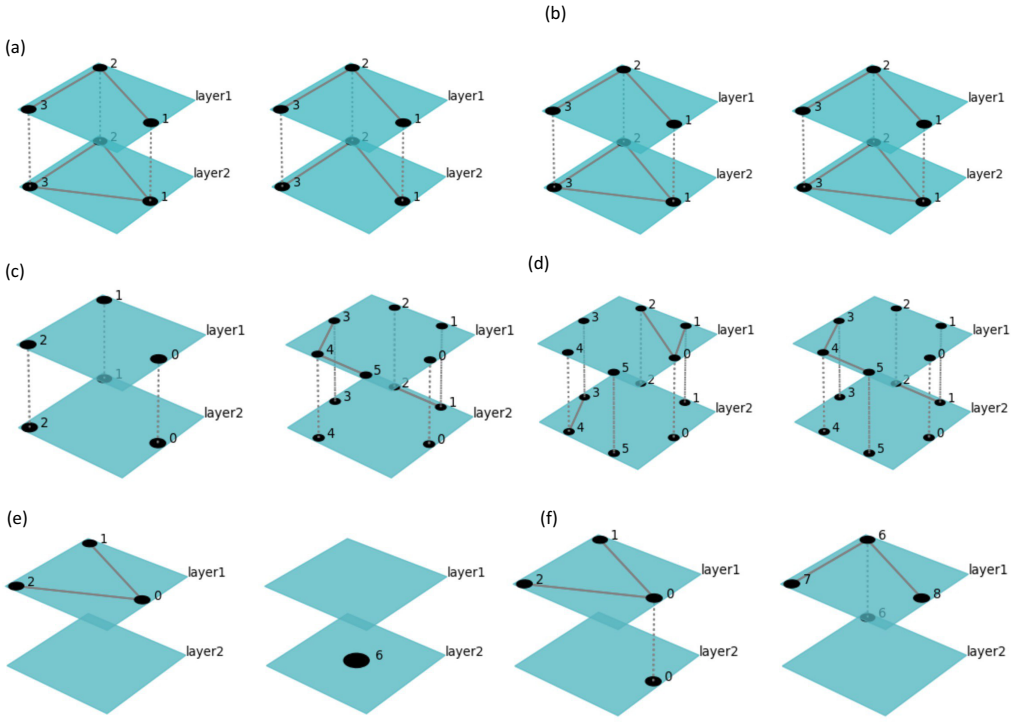

Figure 8.1: Three pairs of multiplex neighborhoods with two layers before anonymization (a, c, e) and after anonymization (b, d, f) with Algorithm 1.

to the cost functions. Those variables can correspond, for example, to the number of nodes added in the full network, number of nodes or edges added in only one-layer and number of overlapping edges added.

8.2 Preventing neighborhood attack on multiplex networks

The algorithm presented in Section 8.1 could be used as the main part for a full multiplex network anonymization algorithm. In this section, we discuss alternatives to the presented algorithm and how our method can be used in a complete algorithm to prevent neighborhood attack.

An alternative to the proposed method could include different ways of matching the two graphs and computing the missing edges or nodes to make them isomorphic. However, there are currently no methods similar, for example, to network alignment [Kuc+10; SXB08; Lia+09] for multiplex networks.

Our algorithm is a heuristic that takes a step towards (non-exact) multiplex graph matching. We can modify it by sorting the connected components and the nodes in them by prioritizing different measures. Furthermore, to match nodes, we could compute the correlation coefficient between a vector of measures of pair of nodes in the two neighbors and match the nodes with the highest correlation. The vector of measures could include, as an example, the degree in the two layers and the overlapping degree. It is anyway important to include multiplex measures while developing this kind of algorithm, because the difference or similarity between the different layers is what makes multiplex networks different from monoplex ones.

Incorporating the procedure proposed in Section 8.1 in a full network anonymization algorithm would mean adding nodes to the network. If we do not want to add new nodes, we could pick already existing nodes and incorporate them in a neighborhoods. Those nodes could be low degree nodes, as proposed for instance in the method presented in Section 3.3.2. However, if the nodes that join the neighborhoods have neighbors themselves, then the neighborhoods may have additional edges as well.

As most of the anonymization methods presented in Chapter 3 do, before anonymizing a pair of neighborhoods, all the neighborhoods in the network need to be organized in groups. Each group needs to contain neighborhoods similar to each other, which then would be made isomorphic to each other. It is important to group together neighborhoods that do not differ significantly, since, in this way, the modification to the original data would be low. In the case of simple networks, sorting the neighborhoods by number of nodes and edges could be sufficient, but in a multiplex case there is always the diversity between the layers to take into account. We can group neighborhoods with a heuristic similar to the one of sorting connected components. Indeed, if we sort the neighborhoods, we can group the ones that are closed to each other in the sorted list. An alternative would be to develop new distance measures for multiplex networks, such as graph edit distance [Gao+10] (discussed in Chapter 2) or graph kernels [Vis+10; She+11]. Graph kernels in particular would be very handy, since, with them, we can compute the distance between two networks, which could be also all the extracted networks' neighborhoods. In general, tools existing for simple graphs needs to be extended to multiplex networks to be used for this kind of algorithm, but also for other purposes.

Chapter 9

Conclusion

We now conclude this study by going through all its steps, summarizing its results (Section 9.1), and discussing the possible future work to extend our analysis (Section 9.2).

We have analyzed the uniqueness and structure of the neighborhoods in complex social networks. Our aim has been to quantify the risk related to neighborhood attack when social network data are shared, and to estimate the difficulty of the anonymization problem to prevent identity disclosure caused by this kind of attacks. We have conducted this study on multiplex networks in different settings, depending on whether the layers' labels are present or not, or the data from different layers are aggregated into a single one. We have defined multiplex neighborhoods and identified the differences that this definition could cause in the uniqueness value.

To understand how different networks behave in terms of anonymization difficulty, we have used three network models to generate networks with different features. For every model, we have varied parameters such as the number of nodes, average degree and, in the multiplex case, amount of edges overlapping between different layers. We have measured the number of unique neighborhoods in each of the networks, identifying the uniqueness' trend in function of various parameters.

Since network models are not always similar to real-world social networks, we have then analyzed the uniqueness of neighborhoods in some social network data of human communications, such as phone calls and text messages. We have compared the results of this analysis with the ones derived by the study of network models, to determine whether models can be used as a proxy for empirical data in terms of uniqueness, and which model is more suitable to approximate social networks.

To further improve the understanding of the difference between data and models, which always have a random ingredient in them, we have

conducted an analysis of the recurrent patterns in social networks, through an analysis of network motifs in neighborhoods. The recurrent structures in neighborhoods are the reason why neighborhoods are similar to each other and, consequently, possible to anonymize without radically modifying the data.

Finally, we have proposed an algorithm to make a pair of multiplex neighborhoods isomorphic. This method can be used as the core building block for a full multiplex network anonymization algorithm, fitting in the framework of existing anonymization methods. We have also discussed the essential features of multiplex networks to take into account during the development of this kind of algorithms.

9.1 Summary of results

This thesis identifies and quantifies the threat related to neighborhood attack in multiplex networks, while discussing possible solutions to address this problem. We have adopted a scientific approach for the understanding of an engineering problem as the anonymization against neighborhood attack. Our goal was to gain a more in-depth knowledge of the behaviour of networks in terms of anonymization difficulty and how network's structure and features affect this. This understanding can give us an idea on the amount of modification we should do to the data to make them anonymous (at least from a neighborhood attack viewpoint) and, consequently, possible to share.

Mainly, we have seen that the uniqueness of neighborhoods is generally higher when networks have a clear local structure and nodes are organized in densely connected groups (e.g. communities). This structure is typical of real-world data, in contrast to completely random situations, where the uniqueness clearly decreases with the growing network size and average degree. On the other hand, structures that connects all the nodes in the same way, such as ring lattices, are already anonymous, since it is not possible to distinguish between nodes.

The main obstacle for existing anonymization algorithm (that we have surveyed in Chapter 3) is the existence of unique neighborhoods, since they need to be anonymized even in the most basic settings (e.g. *k-anonymity* with $k = 2$) to not to be easily re-identified. Furthermore, the distribution of the occurrence frequency of neighborhoods in data shows that the number of classes occurring one time is the highest fraction, while a lower amount of classes are occurring more times. This means that the uniqueness of classes is the hardest to suppress, while, for instance, it would be easier to

make $3-anonymous$ (k-*anonymous* with $k=3$) neighborhoods that are occurring two times.

To understand how multiplex networks influence the uniqueness of neighborhoods, and to avoid the ambiguity that this representation can cause, we have first defined multiplex neighborhoods in Chapter 4. We have identified two different ways of defining neighborhoods in multiplex networks, and we have called them *Non-Inclusive Multiplex Neighborhood*, $\mathcal{N}_{\subset}$, and *Inclusive Multiplex Neighborhood*, $\mathcal{N}_{\subseteq}$. The choice of using one of another definition depends on the attackers' knowledge we want to model. The two neighborhood definitions influence the uniqueness value in multiplex networks because they have different information incorporated in them. In this thesis, we have mostly focused on *Non-Inclusive Multiplex Neighborhood*, since it is a more realistic approach when it comes to neighborhood attacks.

Non-Inclusive Multiplex Neighborhoods are built by extracting the neighborhood of the target node separately from every layer and then combining them into a multiplex network. Conversely, each layer of Inclusive Multiplex Neighborhoods includes the nodes that are neighbors of the target node in at least one layer and the edges between them. We have discussed that the uniqueness of $\mathcal{N}_{\subseteq}$ is always higher (or, at least, equal) than the uniqueness of $\mathcal{N}_{\subset}$. We have also shown that aggregating the layer of a multiplex network into a single-layer network does not always lower the re-identification risk when $\mathcal{N}_{\subset}$ is used. Indeed, the aggregated network could contain information about links that we cannot observe with $\mathcal{N}_{\subset}$. This happens especially when the edge overlap is low. This is explained by the fact that, if two layers would share a lot of edges, then it is likely that two nodes that are part of a neighborhood in one layer, would also be part of it in another one, not adding significant information to the neighborhood when aggregated. Instead, if the edges of two networks are mostly independent (i.e. with low overlap), then the aggregation may reveal additional links, that an attacker can use during the re-identification process.

On the other hand, the uniqueness of $\mathcal{N}_{\subseteq}$ is always higher (or equal) than the one of aggregated neighborhoods $\mathcal{N}_a$, and, depending on the hypothesis and settings of the neighborhood attack we want to protect against, this should be taken into account when sharing network data.

The variation of the uniqueness of neighborhoods is strongly influenced by the network structure, which makes the probability of having equal or similar neighborhoods higher or lower. To represent different network structures, we have conducted simulations with the multiplex version of three network models (Erdős-Rényi, Watts-Strogatz, Random Geometric Graph), showing that the best strategy to share data to minimize the risk of neighborhood attack depends on various variables. As it was expected,

in a multiplex setting, not sharing layers' labels helps, since the amount of available information is always lower (or, at the extreme, equal) than when we share them. The difference in the uniqueness between those two types of situation vanishes when the fraction of overlapping edges between the two layers is close to one, since, concerning neighborhoods, the network is the same as its aggregated version.

We have identified a linear trend in the variation of the uniqueness of neighborhoods, in function of the average degree and network size. The uniqueness depends on those two features when the networks are sparse (i.e., the average degree is not very high relative to the number of nodes). In networks with nodes organized in densely connected groups, the uniqueness presents an almost flat behaviour, meaning that the network size does not radically influence it. This trend characterizing graphs with a clear nodes disposition is due to the fact that, despite the size, the nodes would always be locally densely organized, allowing the formation of a variety of neighborhoods. However, if the structure is completely regular, the uniqueness of neighborhoods dramatically decreases. Indeed, in structures like ring lattices nodes are all anonymous, since they have already the same neighborhoods.

The growth in the possibility of neighborhoods formation in locally dense networks compared to sparse ones is explained by the locally tree-like structure of the latter. This peculiarity reflects in a low probability of having strongly connected neighborhoods and consequently, less diversity in the neighborhood classes.

The tendency of the uniqueness to depend less on the network size as the organization in the network increases is also confirmed by the uniqueness variation in function of the edge overlap, in both Watts-Strogatz and Random Geometric Graph. The uniqueness peak in both networks is with an edge overlap around 0.75. This value corresponds to a network architecture with a more pronounced local structure than a more random one, similar to Erdős-Rényi graphs. Indeed, when the edge overlap is equal to 0, the networks behave like an Erdős-Rényi one, while, when the overlap increases, the typical local structures of the models appear. The uniqueness goes down again when reaching value 1 of total overlap, since there is a loss in the diversity between the two layers.

The behaviour of models such as the Random Geometric Graph approximates quite well the uniqueness of empirical datasets. Real-world social networks are indeed locally dense, since humans (or other social entities) strongly tend to connect with a relatively small number of people. Even in data, the uniqueness increases when the average amount of connections per nodes increases, as a confirmation that the trend that we have seen in

the models is also valid for real data. The difference between locally dense social networks and random models that are locally tree-like can also be noticed with the analysis of neighborhood motifs (Chapter 7). Comparing empirical networks with a randomized version of the network with the same degree sequence (the configuration model) shows the variety of classes that are present in dense networks compared to sparse random ones.

Neighborhoods are structures that can be a serious threat when sharing data. Easier anonymization approaches such as degree anonymization are not enough to protect network data privacy since, as we have shown in Chapter 5, the nodes with a unique degree are not sufficient to explain unique neighborhoods. We have also presented equations to determine the expected fraction of unique degree nodes in both monoplex and multiplex Erdős-Rényi networks, taking into account the amount of edge overlap between different layers. The fraction of unique degree nodes can explain the fraction of unique neighborhoods only if the network is sparse and small, or, in a monoplex case, when is completely connected. In this case, the uniqueness would go to zero, since all the neighborhoods would have the same structure. However, this is not always the case in a multiplex network, because the amount of edge overlap should also be taken into account (at least if we consider as average degree the one of the aggregated network, as we do).

The linear trend of the uniqueness tells us the approximate uniqueness value that a network would have with a given size and average degree. Primarily, it can give us an idea on "how far" a network is from the area in which all the nodes are anonymous ($U_{\mathcal{N}_\tau} = 0$). This information can be used to modify the data by, for example, thresholding the edges or increasing the size, to make neighborhoods anonymous. After the data have been shared, it would also be possible to share the amount of thresholding that has been done, allowing data analysts to estimate actual features of the network such as the average degree. Modifying data in this way would not imply the use of an algorithm that adds, for example, fake edges. This approach is also useful for the design of data collection studies. In fact, it would be possible to estimate network measures that are needed to have a completely anonymous network that can be shared in the future (e.g. number of participants). Therefore, in this case, there would be no more need of anonymization algorithms that introduce noise to the data.

Anonymizing multiplex neighborhoods is a non-trivial problem that can be addressed in the future and, as a first step, we have proposed a heuristic to anonymize a pair of multiplex neighborhoods in Chapter 8. Our method shows that features that are peculiar of multiplexity, such as overlapping edges or overlap degrees, should be considered when adapting any algorithm

to multiplex networks. In particular, the multiplexity should be taken into account when it comes to anonymization, where keeping utility while minimizing data modifications is crucial. However, the choice of using an algorithm should be preceded by an analysis of the situation. Applying an anonymization algorithm to a network where neighborhoods are all unique could modify the data entirely. Since the goal is to share data where all the neighborhoods are anonymous, then it is needed to understand the distance of the data from the area with zero uniqueness. If the data are close enough to this area, small modifications of those could be sufficient to share them safely. On the other hand, if they are in an area with maximum uniqueness that is far from the uniqueness' transition from 0 to 1, applying any method would imply losing data utility.

9.2 Future work

In this thesis, we have studied the problem of privacy and anonymization of neighborhoods in multiplex networks. This work can be expanded in multiple directions, for example by studying different systems or network structures, or analyzing different neighborhood attacks' scenario.

Although we have found the existence of a linear trend in the uniqueness of neighborhoods with definition $\mathcal{N}_{\subset}$, future work can be directed towards a more in-depth study of neighborhoods of type $\mathcal{N}_{\subseteq}$, to understand the differences in such a situation. The choice of the neighborhood definition may affect the behaviour of both network models and data, leading to different results than the presented ones.

Moreover, a systematic study with systems with more than two layers and that are not fully interconnected would be an interesting direction. Increasing the number of layers would also most probably lead to a growth in the uniqueness value, because of more information being available. This analysis could bring new insight in the uniqueness variation depending on the amount of overlapping edges, since they can be of different types: there are edges shared among all the existing layers, but also some appearing in just two out of all the present layers. Alternative models could also be considered to study the uniqueness with a network structure different than the already analyzed ones. We have also focused on networks with sparse regime, but it would also be interesting to explore the dense regime of multiplex networks, in order to understand if there is any substantial difference compared to the monoplex case.

We have seen that the uniqueness of neighborhoods decreases with the network size and average degree. To further explore this trend with real-

world data, it would be possible to take subsets of data from the same datasets, while keeping the average degree, and compute the corresponding uniqueness. In the same way, one can vary the average degree keeping the network size, by thresholding the edges' weights to lower the amount of connections (for instance considering the connections between people happening at least a certain amount of times). Similarly, we can compare the behaviour of the data with the one of models in terms of uniqueness transition. Network models present only a relatively narrow area in which the neighborhoods are neither all anonymous or all identifiable, while real-word social networks may behave differently. The analysis on how some real-world data would behave could give a better overview that can be used for data sharing purposes in order to keep the anonymity of the subjects.

It would be also interesting to consider and analyze the uniqueness of neighborhoods that go beyond 1-hop, extracting the neighbors of neighbors. This would increase the uniqueness value and the re-identifiability, until the situation where, at the limit, the attacker would know the identity of the nodes in the full network.

Following the same framework of our equation for unique degree nodes (in Chapter 5), it is possible to extend them to different network models. This would allow to check, in other multiplex networks, whether the degree uniqueness could explain at an higher (or lower) percentage the uniqueness of neighborhoods.

The whole analysis of this thesis is done with unlabelled graphs, thus considering *k-anonymity* as privacy definition. More advanced definitions such as *l-diversity* can also be taken into account when doing an analysis of the uniqueness with labelled data. This study can be done also on network models, for example assigning a limited amount of labels to nodes according to a certain probability, and measure how the uniqueness varies compared to a situation where nodes' labels are not present.

Nodes' labels can also be used to extend our neighborhood motifs analysis. For instance, as already discussed in Chapter 7, nodes' metadata such as genders can be considered to see which kind of motifs are associated to different types of nodes. Additionally, we have identified problems in our motif analysis related to both sampling and error estimation. The computed error resulted to be very low and the employed uniform sampling did not allow us to see some of the motifs in the random networks which occurred in the datasets. To remedy this problem, we could estimate the probability of occurrence of each subgraph in the null model. In our analysis, we have used the configuration model with degree overlap, but different alternatives can also be used, such as the configuration model without considering the overlapping degrees, or the Erdős-Rényi model. The

configuration model has the same degree sequence as the data, therefore its degree uniqueness does not differ from the original network. We could expand our analysis to understand more in-depth how the degree uniqueness influences the neighborhoods' uniqueness when the clustering coefficient is sparse, comparing the uniqueness of the Erdős-Rényi model and the configuration model.

Furthermore, the computed Z-Score and Significance Profiles strongly depend on the network size. Even though this is known for network motif analysis, with neighborhood motif analysis the values become almost meaningless, making comparison problematic, since most of the times, especially in multiplex networks, the Z-Score is very high while the SP very low. If one conduct the analysis just considering the SP, then most of the subgraphs would not emerge, making the final results different and, probably, not reliable. For this reason, alternative measures for computing the significance of patterns in networks, and, in particular, in smaller local structures like neighborhoods, should be developed to obtain meaningful results that are also comparable to each other.

The presented algorithm to anonymize a pair of neighborhoods can also be modified and compared with alternatives. In particular, other possibilities to align two multiplex neighborhoods and find their edit distance can be considered. Moreover, this method should be tested along with a full network anonymization algorithm, to understand how much data are modified with the proposed heuristic. The test set should include different scenario based, for example, on the structure of networks or different amount of edge overlap. As we have seen, network structure and other features can radically modify the uniqueness and, consequently, the hardness of the anonymization problem. A prior analysis of the network and its features is needed to understand the situation, and, eventually, develop methods that are specific to each scenario. The understanding of the situation could also reveal that varying the size or the number of edges (e.g. by thresholding or randomly removing them), even for a limited amount, would be enough to make a certain dataset anonymous.

To conclude, multiplexity is a characteristic that can be found in an increasing amount of data, and can bring additional problems in terms of privacy compared to simple social networks. Methods to preserve privacy and anonymity in this kind of data should be developed, both addressing neighborhood attacks and other threats. To maximize the utility and performance of those methods, new basic tools for multiplex networks already existing for normal graphs need to be created, in order to be used as building blocks of privacy-preserving or anonymization algorithms. Examples of those tools could include the development of a graph edit distance function for

multiplex networks, or methods to group similar networks together, like graph kernels, with the aim of anonymizing networks without radical modifications. The development of these methods can also give us an idea of the actual distance between the existing neighborhood classes, and consequently an indication of the convenience of anonymizing the dataset. For example, the neighborhoods of a dataset could be similar to each other even though they are all unique. In this case, anonymizing them would require only small modification. Instead, if all the neighborhoods are very distant to each other, anonymizing them would mean to decrease the data quality and, therefore, their utility.

Bibliography

[Bar+16] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[Bat+17] Federico Battiston et al. "Multilayer motif analysis of brain networks". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.4 (2017), page 047404.

[BC78] Edward A Bender and E Rodney Canfield. "The asymptotic number of labeled graphs with given degree sequences". In: *Journal of Combinatorial Theory, Series A* 24.3 (1978), pages 296–307.

[BJ00] Horst Bunke and Xiaoyi Jiang. "Graph matching and similarity". In: *Intelligent systems and interfaces*. Springer, 2000, pages 281–304.

[BW00] Alain Barrat and Martin Weigt. "On the properties of small-world network models". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 13.3 (2000), pages 547–560.

[CFL10] James Cheng, Ada Wai-chee Fu, and Jia Liu. "K-isomorphism: privacy preserving network publication against structural attacks". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM. 2010, pages 459–470.

[Che+13] Sean Chester et al. "Complexity of social network anonymization". In: *Social Network Analysis and Mining* 3.2 (2013), pages 151–166.

[Che+14] Rui Chen et al. "Correlated network data publication via differential privacy". In: *The VLDB Journal – The International Journal on Very Large Data Bases* 23.4 (2014), pages 653–676.

[D+14] Cynthia Dwork, Aaron Roth, et al. "The algorithmic foundations of differential privacy". In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pages 211–407.

[D+15] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. "Unique in the shopping mall: On the reidentifiability of credit card metadata". In: *Science* 347.6221 (2015), pages 536–539.

[D+18] Erik Davis, Sunder Sethuraman, et al. "Consistency of modularity clustering on random geometric graphs". In: *The Annals of Applied Probability* 28.4 (2018), pages 2003–2062.

[De +13] Yves-Alexandre De Montjoye et al. "Unique in the crowd: The privacy bounds of human mobility". In: *Scientific reports* 3 (2013), page 1376.

[Dwo06] Cynthia Dwork. "Differential Privacy". In: *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*. ICALP'06. Venice, Italy: Springer-Verlag, 2006, pages 1–12. ISBN: 3-540-35907-9, 978-3-540-35907-4. DOI: 10.1007/11787006_1. URL: http://dx.doi.org/10.1007/11787006_1.

[Dwo08] Cynthia Dwork. "Differential privacy: A survey of results". In: *International Conference on Theory and Applications of Models of Computation*. Springer. 2008, pages 1–19.

[EG60] Paul Erdős and Tibor Gallai. "Gráfok előírt fokszmú pontokkal". In: *Matematikai Lapok* 11 (1960), pages 264–274.

[ER60] Paul Erdos and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pages 17–60.

[For96] Scott Fortin. *The graph isomorphism problem*. Technical report. Citeseer, 1996.

[Fos+18] Bailey K Fosdick et al. "Configuring random graph models with fixed degree sequences". In: *SIAM Review* 60.2 (2018), pages 315–355.

[FP73] Harary Frank and Edgar M Palmer. *Graphical enumeration*. 1973.

[FW15] Amin Milani Fard and Ke Wang. "Neighborhood randomization for link privacy in social network analysis". In: *World Wide Web* 18.1 (2015), pages 9–32.

[Gao+10] Xinbo Gao et al. "A survey of graph edit distance". In: *Pattern Analysis and applications* 13.1 (2010), pages 113–129.

[GJ02] Michael R Garey and David S Johnson. *Computers and intractability*. Volume 29. wh freeman New York, 2002.

[Hay+07] Michael Hay et al. "Anonymizing social networks". In: *Computer science department faculty publication series* (2007), page 180.

[Hay+08] Michael Hay et al. "Resisting structural re-identification in anonymized social networks". In: *Proceedings of the VLDB Endowment* 1.1 (2008), pages 102–114.

[Hay+09] Michael Hay et al. "Accurate estimation of the degree distribution of private networks". In: *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. IEEE. 2009, pages 169–178.

[HSS03] Elad Hazan, Shmuel Safra, and Oded Schwartz. "On the complexity of approximating k-dimensional matching". In: *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*. Springer, 2003, pages 83–97.

[JK07] Tommi Junttila and Petteri Kaski. "Engineering an efficient canonical labeling tool for large and sparse graphs". In: *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics*. Edited by David Applegate et al. SIAM, 2007, pages 135–149.

[JKM08] Krzysztof Juszczyszyn, Przemysław Kazienko, and Katarzyna Musiał. "Local topology of social network based on motif analysis". In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer. 2008, pages 97–105.

[Kar+03] Hillol Kargupta et al. "On the privacy preserving properties of random data perturbation techniques". In: *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE. 2003, pages 99–106.

[Kar+11] Márton Karsai et al. "Small but slow world: How network topology and burstiness slow down spreading". In: *Physical Review E* 83.2 (2011), page 025102.

[Kas+04] Nadav Kashtan et al. "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs". In: *Bioinformatics* 20.11 (2004), pages 1746–1758.

[Kas+13] Shiva Prasad Kasiviswanathan et al. "Analyzing graphs with node differential privacy". In: *Theory of Cryptography*. Springer, 2013, pages 457–476.

[Kiv+12] Mikko Kivelä et al. "Multiscale analysis of spreading in a large communication network". In: *Journal of Statistical Mechanics: Theory and Experiment* 2012.03 (2012), P03005.

[Kiv+14] Mikko Kivelä et al. "Multilayer networks". In: *Journal of complex networks* 2.3 (2014), pages 203–271.

[Kiv17] Mikko Kivelä. *Multilayer networks library for python (pymnet)*. 2017.

[KP18] Mikko Kivelä and Mason A Porter. "Isomorphisms in multilayer networks". In: *IEEE Transactions on Network Science and Engineering* 5.3 (2018), pages 198–211.

[KSV11] Bruce Kapron, Gautam Srivastava, and S Venkatesh. "Social network anonymization via edge addition". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE. 2011, pages 155–162.

[Kuc+10] Oleksii Kuchaiev et al. "Topological network alignment uncovers biological function and phylogeny". In: *Journal of the Royal Society Interface* (2010), rsif20100063.

[Li+] Chongjie Li et al. *An Improved Label-bag based Graph Anonymization based on Utility*.

[Li+14] Chongjie Li et al. "Label-bag based graph anonymization via edge addition". In: *Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering*. ACM. 2014, page 1.

[Lia+09] Chung-Shou Liao et al. "IsoRankN: spectral methods for global alignment of multiple protein networks". In: *Bioinformatics* 25.12 (2009), pages i253–i258.

[Liu+15] Chuan-Gang Liu et al. "K-anonymity against neighborhood attacks in weighted social networks". In: *Security and Communication Networks* 8.18 (2015), pages 3864–3882.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. "t-closeness: Privacy beyond k-anonymity and l-diversity". In: *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE. 2007, pages 106–115.

[LM14] Wentian Lu and Gerome Miklau. "Exponential random graph estimation under differential privacy". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2014, pages 921–930.

[LT08] Kun Liu and Evimaria Terzi. "Towards identity anonymization on graphs". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM. 2008, pages 93–106.

[Mac+06] Ashwin Machanavajjhala et al. "l-Diversity: Privacy Beyond k-Anonymity". In: *22nd International Conference on Data Engineering (ICDE'06)*. IEEE. 2006, page 24.

[McK83] Brendan D McKay. "Applications of a technique for labelled enumeration". In: *Congressus Numerantium* 40 (1983), pages 207–221.

[Mil+02] Ron Milo et al. "Network motifs: simple building blocks of complex networks". In: *Science* 298.5594 (2002), pages 824–827.

[MSK12] Ali Masoudi-Nejad, Falk Schreiber, and Zahra Razaghi Moghadam Kashani. "Building blocks of biological networks: a review on major network motif discovery algorithms". In: *IET systems biology* 6.5 (2012), pages 164–174.

[MW90] Brendan D. McKay and Nicholas C. Wormald. "Uniform generation of random regular graphs of moderate degree". In: *J. Algorithms* 11.1 (1990), pages 52–67.

[NA13] Mohd Izuan Hafez Ninggal and Jemal H Abawajy. "Neighbourhood-pair attack in social network data publishing". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer. 2013, pages 726–731.

[New02] Mark EJ Newman. "Random graphs as models of networks". In: *arXiv preprint cond-mat/0202208* (2002).

[New06] Mark EJ Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pages 8577–8582.

[New18] Mark Newman. *Networks*. Oxford university press, 2018.

[NS09] Arvind Narayanan and Vitaly Shmatikov. "De-anonymizing social networks". In: *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE. 2009, pages 173–187.

[OWK14] Rina Okada, Chiemi Watanabe, and Hiroyuki Kitagawa. "A k-anonymization algorithm on social network data that reduces distances between nodes". In: *Reliable Distributed Systems Workshops (SRDSW), 2014 IEEE 33rd International Symposium on*. IEEE. 2014, pages 76–81.

[Pen+03] Mathew Penrose et al. *Random geometric graphs*. 5. Oxford university press, 2003.

[Pen+16] Mathew D Penrose et al. "Connectivity of soft random geometric graphs". In: *The Annals of Applied Probability* 26.2 (2016), pages 986–1028.

[PMS18] Beatrice Perez, Mirco Musolesi, and Gianluca Stringhini. "You are your Metadata: Identification and Obfuscation of Social Media Users using Metadata Information". In: *arXiv preprint arXiv:1803.10133* (2018).

[PS10] Michael D Porter and Ryan Smith. "Network neighborhood analysis". In: *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*. IEEE. 2010, pages 31–36.

[Psy+17] Ioanna Psylla et al. "The role of gender in social network organization". In: *PloS one* 12.12 (2017), e0189873.

[RMT15] Luca Rossi, Mirco Musolesi, and Andrea Torsello. "On the k-Anonymization of Time-Varying and Multi-Layer Social Graphs." In: *ICWSM*. 2015, pages 377–386.

[Ros95] Guido Rossum. *Python Reference Manual*. Technical report. Amsterdam, The Netherlands, The Netherlands, 1995.

[SF83] Alberto Sanfeliu and King-Sun Fu. "A distance measure between attributed relational graphs for pattern recognition". In: *IEEE transactions on systems, man, and cybernetics* 3 (1983), pages 353–362.

[She+11] Nino Shervashidze et al. "Weisfeiler-lehman graph kernels". In: *Journal of Machine Learning Research* 12.Sep (2011), pages 2539–2561.

[SP09] Alina Stoica and Christophe Prieur. "Structure of neighborhoods in a large social network". In: *Computational Science and Engineering, 2009. CSE'09. International Conference on*. Volume 4. IEEE. 2009, pages 26–33.

[Sto+14] Arkadiusz Stopczynski et al. "Measuring large-scale social networks with high resolution". In: *PloS one* 9.4 (2014), e95978.

[Swe02] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pages 557–570.

[SXB08] Rohit Singh, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection". In: *Proceedings of the National Academy of Sciences* (2008).

[SY13] Entong Shen and Ting Yu. "Mining frequent graph patterns with differential privacy". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pages 545–553.

[SZ15] Wolfgang E Schlauch and Katharina A Zweig. "Influence of the null-model on motif detection". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*. IEEE. 2015, pages 514–519.

[Tak+18] Frank W Takes et al. "Multiplex network motifs as building blocks of corporate networks". In: *Applied Network Science* 3.1 (2018), page 39.

[TC12] Christine Task and Chris Clifton. "A guide to differential privacy theory in social network analysis". In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society. 2012, pages 411–417.

[TP10] BK Tripathy and GK Panda. "A new approach to manage security against neighborhood attacks in social networks". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. IEEE. 2010, pages 264–269.

[Vis+10] S Vichy N Vishwanathan et al. "Graph kernels". In: *Journal of Machine Learning Research* 11.Apr (2010), pages 1201–1242.

[Wan+14] Yang Wang et al. "Resisting label-neighborhood attacks in outsourced social networks". In: *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*. IEEE. 2014, pages 1–8.

[Wei] Eric W Weisstein. "Binary Search". In: *From MathWorld - A Wolfram Web Resource*.

[Wer06] Sebastian Wernicke. "Efficient detection of network motifs". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 3.4 (2006), pages 347–359.

[Won+11] Elisabeth Wong et al. "Biological network motif detection: principles and practice". In: *Briefings in bioinformatics* 13.2 (2011), pages 202–215.

[WR06] Sebastian Wernicke and Florian Rasche. "FANMOD: a tool for fast network motif detection". In: *Bioinformatics* 22.9 (2006), pages 1152–1153.

[WS98] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world' networks". In: *nature* 393.6684 (1998), page 440.

[Wu+10] Ye Wu et al. "Evidence for a bimodal distribution in human communication". In: *Proceedings of the national academy of sciences* (2010).

[YH02] Xifeng Yan and Jiawei Han. "gspan: Graph-based substructure pattern mining". In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE. 2002, pages 721–724.

[YW09] Xiaowei Ying and Xintao Wu. "On link privacy in randomizing social networks". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2009, pages 28–39.

[ZCÖ09] Lei Zou, Lei Chen, and M Tamer Özsu. "K-automorphism: A general framework for privacy preserving network publication". In: *Proceedings of the VLDB Endowment* 2.1 (2009), pages 946–957.

[Zha+17] Jun Zhang et al. "Privbayes: Private data release via bayesian networks". In: *ACM Transactions on Database Systems (TODS)* 42.4 (2017), page 25.

[ZP08] Bin Zhou and Jian Pei. "Preserving privacy in social networks against neighborhood attacks". In: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE. 2008, pages 506–515.

[ZP11] Bin Zhou and Jian Pei. "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks". In: *Knowledge and Information Systems* 28.1 (2011), pages 47–77.