Universiteit
Leiden

# Master Computer Science

Measuring anonymity on labeled networks

| | |
|---|---|
| Name: | Antonis Mouratis |
| Student ID: | s3412660 |
| Date: | July 3, 2024 |
| Specialisation: | Artificial Intelligence |
| 1st supervisor: | Mark van der Loo |
| 2nd supervisor: | Frank Takes |
| 3rd supervisor: | Rachel de Jong |

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

## Abstract

Recently it has become possible for researchers to collect data on individuals and the connections between them. As the field of network science uses such network data to explain behavior in the real world, researchers often have a compelling interest in publishing this data. However, sharing possibly sensitive network data requires the reassurance of anonymity of the entities within it, resulting in a breach of privacy otherwise. This particular concept is the central object of study in the field of statistical disclosure control and has led to the development of several measures for assessing node anonymity. Nevertheless, the largest share of the work thus far is focused on an unlabeled network representation of the data. In this work, we present a novel algorithm that builds upon an existing algorithm for measuring $d$-$k$-anonymity We extend it by making it applicable to labeled networks. Our objective is to compare results produced with and without labeling, to see how anonymity is affected. For this purpose, we used graph models and real-world networks and applied four different types of binary labeling. Three of them are centrality-based and one of them is random. Our findings showed that anonymity decreases when one labels the nodes of the graph, as opposed to unlabeled nodes, in both the graph models and the real-world networks, and that the more nodes we have with different labels assigned to them, the more unique nodes we are expected to have.

# Contents

# 1 Introduction

In recent years there has been an apparent increase in human interactions due to the proliferation of social media, activities, and events. People from all over the world can meet and socialize with other people, that otherwise would not have the chance. This phenomenon has reshaped connections as social circles tend to become more and more interconnected [1].

These connections can only be visualized with the help of network science. A network consists of nodes that are interconnected to each other based on some information. In a social network, the nodes represent individuals and the edges that connect them show a type of relationship that they might have (e.g. family, friends, co-workers, etc.). Researchers are analyzing such networks to find potential patterns of connectivity between the nodes and structural properties that might yield interesting results about the individuals. For instance, Statistics Netherlands investigated segregation in the Dutch population. The researchers gathered information about everyone's neighbors, family, classmates, colleagues and housemates, and they used a network where nodes represent individuals, and links represent whether two nodes are neighbors, coworkers, schools [2, 3].

Such scientific research requires a large amount of data to work on, and its publication can result in a possible breach of privacy. Publishing such data introduces a grave re-identification risk, but also attribute disclosure. Thus, there is a need to properly *anonymize* the data before doing so. This can be done by altering it so that if an adversary wishes to de-anonymize the data, there won't be a breach of privacy issue. In the case of microdata and (aggregated) tabular data, this particular field of study belongs to statistical disclosure control [4, 5]. In our case disclosure control for network data since the information presented is based on the topology ('connectedness') of the network, meaning how the nodes are connected by edges, its structural properties give away some very crucial information about the nodes.

To clarify these principles we consider a small example of an attacker scenario. We consider a network that depicts the people of a small town. The nodes represent the people and the links (edges) represent a friendship between them. Now we consider a certain individual represented by a node $v$ within the network that has five friends. We refer to this node as *target node*. Thus, $v$ is connected to five other nodes. We refer to them as *neighbors*. This is a structural property of the target node in the network. In a hypothetical attacking scenario where an adversary wishes to identify that individual and possesses this particular structural information, they can use it to focus on looking for nodes with only five neighbors.

To improve their chance of finding that node we label the nodes according to their gender and we consider that their friends are all male. The difference now is that the adversary not only knows the structural properties of the node they are interested in but also possesses information about the gender of the individuals in the network. This way, the network provides more information for each node since the adversary can narrow their search down to a target node connected to five other nodes with male gender label. By filtering out those nodes from the dataset, they can gain information about the node. If they find a unique node we speak of re-identification.

Therefore, anonymizing the network data is crucial before publishing it. While there have been several ways to achieve it [6, 7, 8, 9, 10, 11, 12, 13], this thesis project is going to exclusively focus on $k$-anonymity, which is also the most well-known approach. With $k$-anonymity we refer to $k$-number of nodes in the graph that share a common characteristic. That means we have $k$ equivalent nodes and if an adversary has certain knowledge of the network structure surrounding a node, they have $1/k$ probability to re-identify it. Such a property could be the number of direct neighbors that corresponds to the degree of a node, or one could examine another extreme case, the $k$- automorphism introduced in [8], where nodes are only equivalent if they are theoretically indistinguishable due to symmetry in the graph structure, but such a measure does not apply to real networks since it is not often that we find much symmetry.

Although there has been significant work on unlabeled graphs over the recent years, showing promising results, there is a need for research on labeled graphs. Extending an approach from unlabeled to labeled networks presents open questions and challenges regarding the implementation. The purpose of this research is to investigate (1) how to extend the definition of $d$-$k$-anonymity to the situation of labeled graphs, and (2) how partial knowledge of labeling on the attacker's side affects the risk of disclosure.

Our work is an extension of the one in [14, 15, 16] which introduces the measure of $d$-$k$-anonymity, which is a parameterized version of $k$-anonymity, with $d$ referring to the radius of the neighborhood. We focus on the case of $d = 1$ which is the same as 1-neighbourhood isomorphism [10] and we apply various types of binary labeling on the graph to measure how anonymity is lost when one has more information about the neighborhood, other than only the structure. We test our approach on three graph models, namely Erdős–Rényi model [17], Barabasi model [18] and Watts–Strogatz model [19], and then on several real networks as well. To sum up, the main contributions of this thesis are as follows:

- Extend $d$-$k$-anonymity to the case of labeled graphs
- Examine various attacker scenarios
- Compute the anonymity of nodes in three graph models
- Compute anonymity in several real-world networks

The remainder of the thesis is structured as follows: first, we go through the already existing literature of $k$-anonymity approaches in Section 2, then we give the necessary background regarding graph theory and anonymity measures, as well as describe the three used graph models in Section 3. Next, in Section 4 we discuss the approach to solving our problem by presenting the algorithm and describing with a small example the difference between unlabeled and labeled $d$-$k$-anonymity. Section 5 contains a more elaborate description of the data used in the experiments and Section 6 the results of the experiments, both on graph models and real-world data. Lastly, Section 7 will conclude this thesis by summarizing the most important findings and discussing some suggestions for future work.

# 2 Related work

This section is focused on briefly describing well-known ***anonymity measures***, ***measuring anonymity***, and how the two can be combined with ***node-labeling*** of a graph. A more in-depth summary is given in [20].

## 2.1 Anonymity measures

To anonymize a graph one can choose between various measures. A first approach would be to focus on the nodes of the graph. By this, we refer to the case of $k$-anonymity introduced in [6, 7, 8, 9, 10]. A graph is $k$-anonymous if for each node there are at least $k$-1 equivalent nodes concerning the used measure. This is also the measure used in this thesis.

The measure can be put into three main categories as follows: two approaches introduced in [6, 7] that are based on degree. Two automorphism-based introduced in [8, 9], which categorize the nodes as equivalent if they have the same structural position in the graph. One that is based on isomorphism in [10] and considers the nodes equivalent if their neighborhoods are isomorphic and they have the same structural position in their respective neighborhoods.

A different approach is related to preserving edge privacy [11]. Another introduced in [12] is related to privacy in clusters with the nodes divided into them, forming supernodes connected to each other. What is more, we have another approach based on the Szemeredi Regularity Lemma, where they compute a partition of the nodes with the lemma making sure that such a partition exists. The graph is anonymized by randomizing the edges in every set, and the result is a graph maintaining the structural properties but the nodes are now indistinguishable. This approach has been introduced in [13] and it is shown that it can generate large anonymity groups with minimal information loss.

The mentioned measures focus on attacks based on structural knowledge but do not take into account attribute knowledge that an attacker might have. When an attacker possesses information about node attributes, such as the gender of the individual, it is likely that the re-identification risk becomes larger, which is the main focus of this thesis. Some approaches that extend these measures to protect graphs with this kind of scenario are [21, 22] which utilizes node labels, and [23] labeled edges. Our work utilizes centrality-based labeling which is not apparent in these works.

## 2.2 Measuring anonymity

There has been a lot of research around the topic of graph properties and how they are related to anonymity. The research in [7] delves into the distribution of anonymity within various real-world networks, presenting theoretical findings on ER graphs through a parameterized degree-based measure. Additionally, exploring the de-anonymizability of graphs and the preservation of graph properties post-anonymization, [24] concludes that the retention of degrees can be highly revealing. Moreover, in [25] it is investigated the relationship between graph size and privacy using the linkage covariance metric, revealing that larger graphs afford greater anonymity to nodes. What is closer to our work though is the one presented in [26] focuses specifically on the 1-neighbourhood of nodes, considering nodes equivalent if their 1-neighbourhoods prove to be isomorphic.

This thesis will focus on $d$-$k$-anonymity for $d = 1$. That means we examine the 1-neighborhood of the nodes in the graph and look for isomorphism. What is a new addition though is the use of labeling of the nodes. Real social networks often have a type of labeling for the individuals represented by nodes, like splitting them according to gender. As one might expect from real life, if an attacker obtains more information about someone, he is more likely to find him easier. That is also expected to be the case in our scenario with social networks. The question is how fast is the anonymity lost and if the type of labeling connected to structure (e.g. centrality-based) yields different results from completely random assignment.

# 3 Background

This chapter is split into segments. The first part provides a concise introduction to graph theory, giving key definitions and establishing the notation used throughout this thesis. The subsequent subsection describes anonymity measures within graphs. Within this, we explore the terminology about anonymity, discuss several $k$-anonymity measures, and conclude by giving a brief explanation of what binary labeling is and how anonymity is defined under binary labeling of the graph. In the last part of this section, we discuss the three graph models namely Erdős–Rényi, Barabasi, and Watts–Strogatz model that are going to be used later in the experiments in Section 5. We present the way they are built and discuss some of their properties.

## 3.1 Graph theory and notation

In this subsection, we take the example given in Section 1 and we get into more detail by giving the mathematical notation for what is discussed. For one, a characteristic that is of great importance when talking about a graph is ***directionality***. For instance, if we consider we consider two friends like our example, there is a directed link from one to another and the other way around. This makes the link between them an undirected link since the relation between them works both ways. If a graph contains only undirected links it is considered ***undirected***. The thesis is going to only focus on undirected graphs. Following our example, if an individual has for example five friends, we can translate it to a node having five direct neighbors (adjacent nodes) in the graph. The number of these direct connections is referred to as the ***degree*** of the node. These adjacent nodes, along with the edges between them, compose a subgraph called ***neighborhood*** of the node. If we examine the subgraph induced by the nodes with distance at most ***d*** from the node, we refer to ***d-neighborhood***. A small example of an undirected graph is shown in Figure 1, taken from [14] that illustrates what said so far.

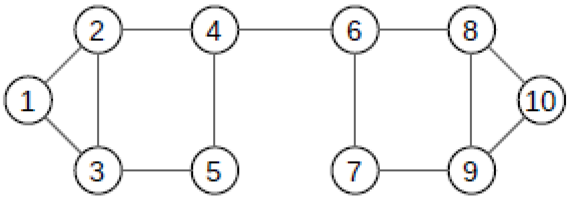


Figure 1: Example of a graph with 10 nodes, represented by circles, and 13 edges, represented by undirected lines

In the Figure, we can see that node 1 is directly connected to nodes 2 and 3. So, it has a degree of 2. Then, because nodes 2 and 3 are connected to 4 and 5 respectively, the 2-neighborhood of node 1 contains 4 nodes. Now, we are going to give more strict definitions to everything we established with this small example.

**Definition 3.1** (Graph)**.** We consider a graph $G = (V, E)$ as a set of nodes $V = \{1, 2, \ldots, |V|\}$, and edges $E$, where $|V|$ is the number of nodes in the graph. The set $E$ consists of pairs of nodes $(u, v)$, with $u, v \in V$.

In the case of undirected graphs we have the following definition:

**Definition 3.2** (Undirected Graph)**.** An undirected graph $G = (V, E)$ is a graph where $V$ represents a set of nodes, $E \subseteq V \times V$ is the set of edges, and $(v, v') \in E$ iff $(v', v) \in E$.

Now that we have formally defined a graph mathematically, we can give the rest of the definitions.

**Definition 3.3** (Degree)**.** Let $G = (V, E)$ be a graph. The degree of a node $v \in V$ is the number of nodes directly connected to $v$.

$$degree(v) = |\{(v, w) \ s.t. \ (v, w) \in E\}| \tag{1}$$

**Definition 3.4** (Distance)**.** Let $G = (V, E)$ be a graph. The distance between two nodes in a graph is the minimum number of edges in a path connecting them. Consequently, the distance between a node $v \in V$ and itself is 0.

**Definition 3.5** (Eccentricity)**.** The eccentricity $\epsilon(v)$ of a node $v$ is the greatest distance between $v$ and any other node $w$.

$$\epsilon(v) = \max_{u \in V} d(v, u) \tag{2}$$

**Corollary 3.0.1** (Diameter)**.** The diameter d of a graph is the maximum eccentricity of any node in the graph. That is, d is the greatest distance between any pair of nodes.

$$d = \max_{v \in V} \epsilon(v) = \max_{v \in V} \max_{u \in V} d(v, u) \tag{3}$$

**Definition 3.6** (Density)**.** Let $G = (V, E)$ be a graph with $|V|, |E|$ being the number of nodes and edges accordingly. Density is the ratio of the number of actual edges over the possible or potential edges of the graph and is given in Equation 4.

$$density = \begin{cases} \frac{|E|}{|V| \cdot (|V|-1)}, \text{if } G \text{ is directed} \\ \frac{2|E|}{|V| \cdot (|V|-1)}, \text{otherwise} \end{cases} \tag{4}$$

For the purposes of this thesis, we consider the density to be proportional to the fraction of the average degree over the square of the number of nodes of the graph as shown in Equation 5.

$$density \propto \frac{average\ degree}{|V|^2} \tag{5}$$

One can view a graph as follows: dense if the number of edges is close to the maximal number of edges and sparse if the number of edges is close to the minimal number of edges. A graph without edges has 0 density and a complete graph has 1.

Additionally, we give the definition of what an induced graph is. Given a $G = (V, E)$ graph, we consider an induced graph out of it to be a subgraph by selecting a subset of nodes $V' \subseteq V$ and adding all edges from $E$ between the chosen nodes.

**Definition 3.7** (Induced subgraph)**.** Let $G = (V, E)$ be a graph and a subset of nodes $V' \subseteq V$. The subgraph induced by $V'$ is defined as $G' = (V', E')$, where $E' \subseteq E$, such that $\forall v, w \in V'$, if $(v, w) \in E$ then $(v, w) \in E'$ as well.

When working on a real network, for instance, one that depicts social media like Facebook or X, there might be more influential individuals. A celebrity for example is very likely to have connections to many users since people tend to follow them online. That means the neighborhoods of many nodes might intersect because of that. Such nodes with this characteristic are often called central nodes of the graph. The extent of how important they are in the interconnections between the nodes can also be measured with something called ***centrality measures***. There are various centrality measures used in Network analysis like ***Pagerank*** [27], but in this thesis, we are going to focus on three other major ones, namely ***Closeness, Betweenness, and Degree centrality***.

We consider a graph $G = (V, E)$ and a node $v \in V$. **Closeness centrality** indicates how close a node is to all other nodes in the network. It is calculated as the reciprocal of the average of the shortest path length from the node to every other node in the network [28]. A more strict definition and equation are given in 3.8.

**Definition 3.8** (Closeness centrality)**.** Closeness centrality of a node $v$ is the reciprocal of the average shortest path distance to $v$ over all $n - 1$ reachable nodes.

$$C_C(v) = \frac{|V| - 1}{\sum\limits_{u \in V \setminus \{v\}} d(v, u)} \tag{6}$$

where $d(u, v)$ is the shortest-path distance between $v$ and $u$, and $|V| - 1$ is the number of nodes reachable from $v$.

**Betweenness centrality** is a widely used measure that captures a person's role in allowing information to pass from one part of the network to the other.

**Definition 3.9** (Betweenness centrality)**.** Betweenness centrality of a node $v$ is the sum of the fraction of all-pairs shortest paths that pass through $v$.

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)} \tag{7}$$

where $\sigma(s, t)$ is the number of shortest $(s, t)$-paths, and $\sigma(s, t|v)$ is the number of those paths passing through some node $v$ other than $s, t$. If $s = t, \sigma(s, t) = 1$ and if $v \in \{s, t\}, \sigma(s, t|v) = 0$

**Degree centrality** The degree centrality for a node $v$ is the fraction of nodes it is connected to over the number of nodes of the graph.

## 3.2 Graph labeling

Now that we know what a graph is and some necessary notation about it, we will introduce what labeling is more formally than already mentioned in the previous sections. In continuation of our example of an attacking scenario in Section 1, we wish to add labels to the nodes of the graphs such that they show whether a node represents a man or a woman. That can be done by using **binary labeling**. For instance, give the nodes representing men the number 0, and the ones representing women the number 1. A more formal definition is given below in Definition 3.10.

**Definition 3.10** (Binary labeling)**.** Let $G = (V, E)$ be a graph. Binary labeling assigns a binary attribute $b(v)$ to each node $v \in V$. The assignment is done by a function $f$:

$$f : V \longrightarrow \{0, 1\} \tag{8}$$

In our example, $f(v) = 0$ if $v$ is a man and 1 otherwise. While there are many other types of labeling, this thesis is going to only focus on binary.

## 3.3 Probability theory

In order to create random binary labeling for the nodes of the graph, we use a ***Uniform distribution***. The *Cumulative Distribution Function (CDF)* for *Uniform distribution* in an interval $[\alpha, \beta]$ is given below in Equation 9.

$$f(x) = \begin{cases} 0, x < \alpha \text{ or } x > \beta \\ \frac{1}{\beta-\alpha}, \alpha \leq x \leq \beta \end{cases} \tag{9}$$

What Equation 9 shows is that if $x \in [\alpha, \beta]$, then there is probability $\frac{1}{\beta-\alpha}$ for something to occur. In practice for this thesis, we choose a probability $r \in (0, 1)$ we sample a number from a Uniform distribution ($X \sim Uni(0, 1)$) and if the $X$ is greater than $r$ then we assign the label 1 to the node, otherwise 0.

Lastly, we give the Probability Mass Function (PMF) of the ***Poisson distribution*** for a discrete variable X and parameter $\lambda > 0$ in Equation 10.

$$f(x) = P(X = x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}, x = 0, 1, 2, \ldots \tag{10}$$

## 3.4 Measuring anonymity

When we are given a network such as the one mentioned as our example in Section 1 and we consider an attacking scenario where an adversary wishes to identify certain individuals within the network, we would like to measure how likely it is to happen. To do that several metrics have been developed as described in Section 2. What we are going to discuss here is how we can define anonymity mathematically.

We define an equivalence relation which gives a partition for the nodes of the graph (see Definition 3.11). Each node in the graph can only belong to one class: an **equivalence class**. nodes that belong to the same class are equivalent under a common characteristic, e.g. degree, which is the simplest one.

Now, if such a class consists of $k$ nodes, we call each ***node k-anonymous***, meaning that the larger the value of $k$, the higher anonymity is achieved for the nodes inside the class. If an equivalence class is of size 1, then it means it only contains one vertex, making it unique. Consequently, if a class is of size 2 then the nodes are almost unique and if an adversary that has enough information to determine that a target node is in this class, now has only a 50% chance of identifying his target in the network. On the opposite side if a class consists of all the nodes of the graph (i.e. $|V|$), then every node is equivalent to the rest, and thus all nodes are equally anonymous, achieving the maximum anonymity we can get. Accordingly, we call a graph ***k-anonymous*** if all equivalence classes in the equivalence partition have a size of at least k. This implies that all nodes are at least $k$-anonymous.

Consequently, nodes that belong to classes of size $k = 1$ are more susceptible in case of an attack. These nodes are referred to as ***unique*** [14, 26] and can be seen as $\frac{\# \, number \; of \; k{=}1 \; nodes}{\# \, nodes \; in \; the \; graph}$. The same applies to the nodes in classes of size $k = 2$, since if an adversary has enough information to determine that a target node is in this class, now has only a 50% chance of identifying his target in the network.

**Definition 3.11** (Equivalence relation)**.** Let $G = (V, E)$ be a graph We say that $\mathcal{R}$ is an equivalence relation on the subset $V \times V$ if it satisfies the following three properties:

- $(a, a) \in \mathcal{R}\ \forall a \in V$ (reflexive)
- $(a, b) \in \mathcal{R}$ iff $(b, a) \in \mathcal{R}$ (symmetric)
- if $(a, b) \in \mathcal{R}$ and $(b, c) \in \mathcal{R}$ then $(a, c) \in \mathcal{R}$ (transitive)

We denote $a\mathcal{R}b$ if $(a, b) \in \mathcal{R}$.

Using this framework for measuring anonymity we will now discuss in detail three categories of equivalence measures: degree-based, isomorphism-based and automorphism-based.

### 3.4.1 Degree-based

Using the example in Figure 1, we partition the nodes based on their degree as shown in Figure 2 from [14]. Nodes with the same color belong to the same class. The white-colored nodes represent the ones with a degree of 2 and the blue ones with a degree of 3. Thus, we have *6-anonymous* blue nodes and *4-anonymous* white nodes, making the graph *4-anonymous*.

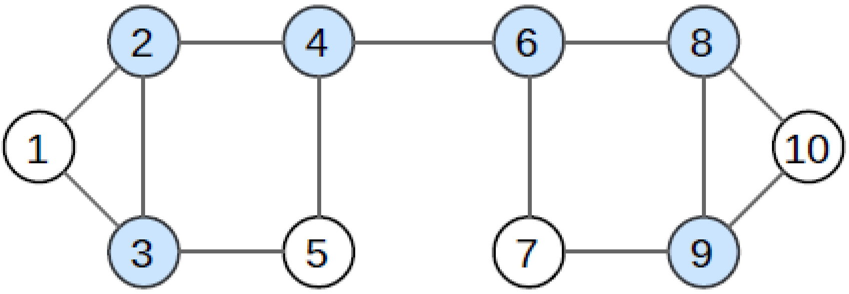


Figure 2: Graph with nodes colored according to their degree: nodes with the same color are in the same equivalence class. Blue nodes have degree 3, white nodes have degree 2

While in our example the nodes are not easily distinguished if their degree is given, there can be instances where the attacker might have additional information. Take, for instance, the comparison between nodes 3 and 6: while both share the same degree, node 3 participates in a triangle, whereas node 6 acts as a bridge connecting to node 4 on the left side. Notably, node 3 lacks connections to the other graph part. Thus, if an adversary gains access to such information, there is higher chance of distinguishing them becomes. Consequently, in many scenarios, a more strict measure is preferred. In [7], a configurable degree-based measure is proposed. This measure involves partitioning nodes with degree-based approach, followed by partioning them based on the degree distribution within various distance thresholds. However, despite its increased strictness compared to the much simpler scenario of a vertex's degree alone, this measure overlooks structural properties like triangles, which could still be discernible to a potential attacker.

### 3.4.2 Automorphism-based

A more strict example of measuring anonymity is introduced in [8], that is partition the nodes based on automorphism. An automorphism of a graph is considered to be a form of symmetry in which it is mapped onto itself while preserving the edge–node connectivity. A more formal definition is given in 3.12

**Definition 3.12** (Automorphism)**.** Let $G = (V, E)$ be a graph. An automorphism of the graph is a permutation $\sigma$ of $V$, $\sigma : V \longrightarrow V$, such that for a pair of nodes $v, u \in V$, there is an edge $(v, u) \in E$ iff the pair $(\sigma(v), \sigma(u)) \in E$.

In [8], a graph is considered $k$-anonymous if there are at least $k$-automorphic functions such that each node is mapped onto a different node for each of these automorphic functions. If a node $v \in V$ is mapped to a node $v' \in V$ by at least one existing automorphic function, we consider them to belong to the same **orbit**. Taking this into account the orbit of a node corresponds to the equivalence class of the vertex. Thus, for a graph to be $k$-anonymous with respect to automorphism, the orbits should all have a size of at least $k$. An example is shown in Figure 3.

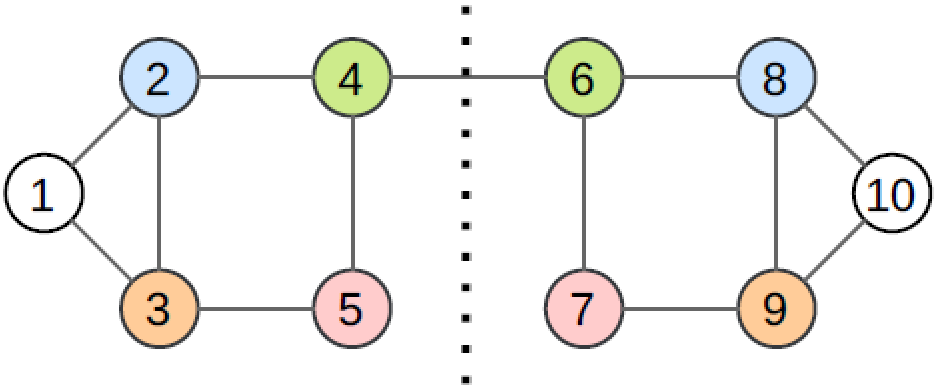


Figure 3: Equivalence classes with respect to automorphism. Figure taken from [14]

In the example illustrated in Figure 3, all nodes are 2-anonymous with respect to $k$-automorphism, and compared to the degree-based approach they can not be further distinguished. This shows the strictness of this measure; theoretically, nodes in the same equivalence class, or orbit, are indistinguishable even when all possible information about the graph is known. A disadvantage of this approach is that this implies that there should be symmetry in the graph for nodes not to be unique. For a graph to be $k$-anonymous with respect to automorphism, the graph should be symmetric in at least $k$-1 points. In this particular example, the symmetry is achieved with edge $(4, 6)$, making it 2-anonymous. However, in the case of real-world graphs, it is not likely that the graph is symmetric. Because of this, $k$-automorphism is often a too strict measure in practice, and a more practical one is needed.

### 3.4.3 Isomorphism-based

The level of strictness of $k$-automorphism, requires us to find a weaker measure, but not too weak like degree. That can be achieved with $k$-isomorphism. With this, we examine the **1-neighborhood** of a vertex, which is a *subgraph* induced from the node and its connection to other nodes. This concept is often referred to as *ego network* of a vertex. Conceptually, two graphs are isomorphic when they can be put on top of each other such that all nodes and edges coincide. A formal definition of graph isomorphism is given in 3.13

**Definition 3.13.** Let $G = (V, E), G' = (V', E')$ be two graphs. The graphs are called isomorphic if there is a bijection $\phi : V \longrightarrow V'$ such that $\forall v, u \in V$ it holds that $(\phi(v), \phi(u)) \in E'$, if $(v, u) \in E$.

An example of an ego network is shown in Figure 4 from [14]. We have isolated the 1-neighborhood of node 9. This particular subgraph contains node 9, all its direct neighbors, and all edges that connect them.

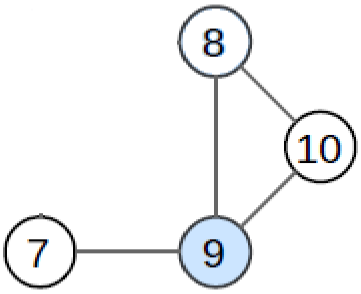


Figure 4: The 1-neighborhood of node 9

When one is working on a graph and wishes to partition the nodes based on their 1-neighborhood, one can check whether the degrees are equal and the connections between the neighbors are the same. That is also what makes it stronger than the degree-based approach, since this way we are able to identify triangles in a neighborhood.

First, we look at Figure 5. The graph is colored, with the colors indicating the four different equivalence classes created based on *isomophism on 1-neighborhood*. We can see that there are three classes of size 2 and one of size 4. The color red is used to illustrate the *d-neighborhoods* of node 3.

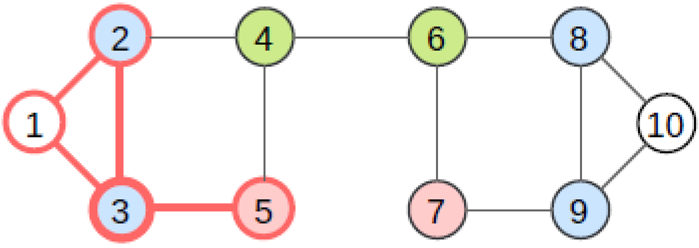


Figure 5: nodes are colored if they are equivalent with respect to 1-neighborhood isomorphism

Since this thesis aims to measure how anonymity is lost when we apply some kind of labeling as opposed to an unlabeled graph, we need to give a more strict definition for isomorphism.

**Definition 3.14** (Colored-isomorphism)**.** Let $G = (V, E)$ and $G' = (V', E')$ be two graphs, and two functions $f : V \longrightarrow \{0, 1\}, g : V' \longrightarrow \{0, 1\}$ as assignments of colors of the nodes (Definition 3.10). The graphs are called node colored-isomorphic if there is a bijection $\phi : V \longrightarrow V'$ i) $\forall (v, u) \in E$ it holds that $(\phi(v), \phi(u)) \in E'$, and ii) $f(v) = g(\phi(v)), \forall v \in V$.

To better illustrate what we mean with the Definition 3.14, we consider the elaborate example introduced in Figure 1. We choose to color the nodes based on their degree (see Figure 6), and randomly (see Figure 7). In both cases, we split the nodes using binary labeling, such that four nodes have the color green and 6 nodes have the color white. Then, we compute the equivalence classes in all three cases (i.e. unlabeled, degree-labeled, random-labeled) and determine the differences between them.

First, we apply binary labeling based on the degree of the nodes. The nodes with the higher degree value (3) have the color white, while the nodes with degree value 2 have the color green. Things are different now since in order for two subgraphs to be isomorphic, the colors between the nodes need to match, in addition to the isomorphism itself. By looking at Figure 6, we can see that the previous class that consisted of nodes [2,3,8,9] now is split into two ([2,8], [3,9]), meaning that now nodes 2,3,8,9 are more likely to be re-identified than before.

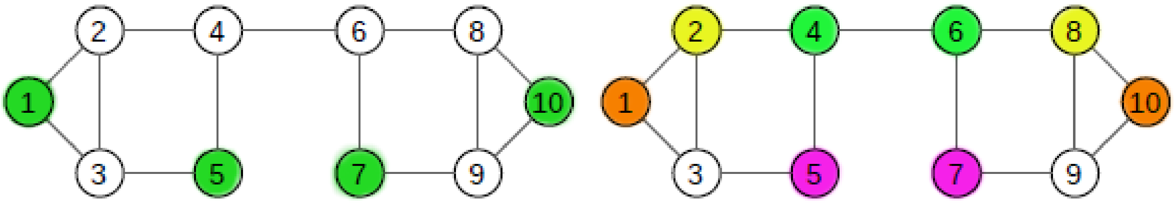


Figure 6: On the left side, the colors indicate the split based on degree-binary labeling. On the right side, nodes are colored if they are equivalent with respect to 1-neighborhood colored isomorphism

In this scenario, there was no huge difference made apart from this split. However, if one chooses to apply some other type of labeling then things could change. Now, we choose to use random binary labeling, meaning that the nodes are colored completely randomly. The split is again 4 and 6 nodes. Figure 7 shows that now we have a more significant effect. Most of the classes collapse and we now end up having 8 unique nodes. Nodes 1, 2, 3, 4, 5, 6, 7, 10 are now unique, and nodes 8, 9 are almost unique, meaning that anonymity is decreased a lot in comparison to the unlabeled version.

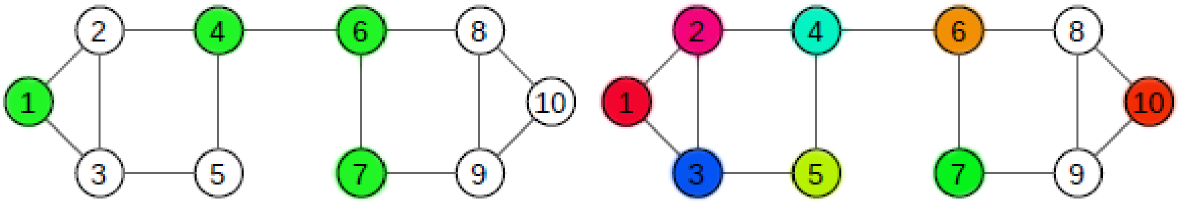


Figure 7: On the left side, the colors indicate the split based on random-binary labeling. On the right side, nodes are colored if they are equivalent with respect to 1-neighborhood colored isomorphism

With just a very small example of a graph with ten nodes, we can see the difference in anonymity between unlabeled and labeled graphs. What we are interested to see is how anonymity is lost in a random graph but with certain structural properties (graph models) and real networks with many more nodes.

## 3.5 Graph models

In the following subsections, we will briefly describe the three graph models used for the conducted experiments. We describe how the graphs are generated and how they can be used to help us interpret the results of real-world networks, due to their size and various properties that can be controlled by parameters

### 3.5.1 Erdős–Rényi

Erdős–Rényi (ER) graphs are considered to be the simplest random graphs that one can experiment on. Due to its tunable parameters, it is possible to control the size and density of the graph. A small example is given in Figure 8.

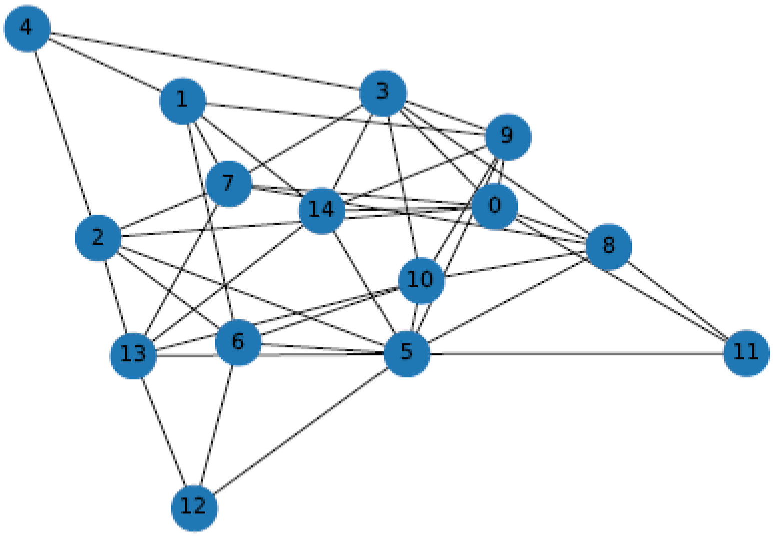


Figure 8: Example of ER graph with $|V| = 15$ and $p = 0.3$. Figure taken from [14].

If one wishes to create an ER graph, one needs to know the number of graph nodes $|V|$ and a probability $p$. Then, the procedure goes like this:

- Iterate over all possible edges the graph can contain, which equals $\frac{|V| \cdot |V|-1}{2}$ for undirected graphs.
- Add each edge with a probability $p$.

By tuning the parameter $p$, one has control over the density of the graph. The higher the value, the more dense the graph is. If $p = 0$ we have a graph with no edges and if $p = 1$ we get a complete graph. Moreover, since the edges are added uniformly, there is no bias in which edges are added or not. As a result, we get a Poisson-shaped degree distribution, where $0 < p < 1$ and the expected degree of a node equals $p \cdot (|V| - 1)$.

However, the creation of ER random graphs has two downsides. It does not take into account two major properties observed in real networks.

- Due to the fact that they have a constant, random, and independent probability of two nodes being connected, ER graphs have a low clustering coefficient.
- They do not account for the formation of hubs. The degree distribution of ER graphs converges to a Poisson-shaped one instead of a power law observed in many real-world, scale-free networks.

Those two issues can be dealt with by using two other types of graph models, Barabási-Albert and Watts-Strogatz.

### 3.5.2 Barabási-Albert

The Barabási Albert (BA) model aims to recreate the power-law degree distribution that is often observed in real-world networks. This makes this model more realistic than the ER model and is achieved by taking into account two observations that often occur in real-world graphs: growth and preferential attachment. Growth in the context of a graph is done through the addition of nodes, which results in the creation of new edges. Preferential attachment signifies that nodes are more likely to connect to nodes with higher degrees compared to those with lower degrees. This contrasts with the equal likelihood of connecting to any vertex, which is the case in ER graphs. The formula for how to compute the probability $p_{attach}$ that a new node $w$ attaches to an existing node $v$ is given in Equation (11).

$$p_{attach}(v) = \frac{degree(v)}{2 \cdot |E|} \tag{11}$$

The procedure of how the BA networks are built in this thesis is described below:

- Start with $l$ nodes, where $0 < l < |V|$, that form a complete graph[1].
- Iteratively add the remaining $|V| - l$ nodes, by adding $l$ edges for each vertex. Here each node has a probability of $p_{attach}$ to be selected.

A small example of a BA is given in Figure 9 from [14].

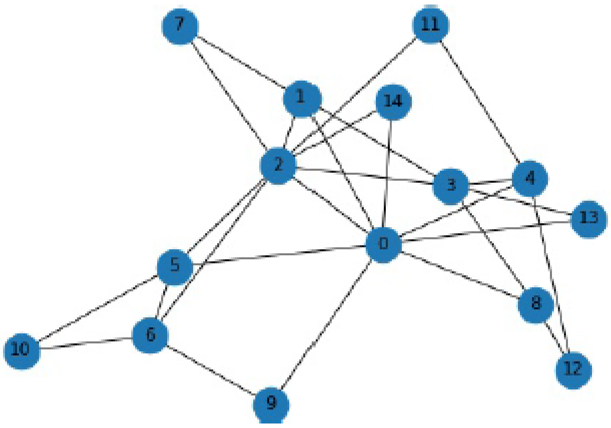


Figure 9: BA graph with $|V| = 15$ and $|E| = 2$

### 3.5.3 Watts-Strogatz

The Watts and Strogatz model was designed to be the simplest model that combats the low clustering issue of ER graphs. It accounts for clustering while maintaining the short average path lengths of the ER model. It does so by interpolating between a randomized structure close to ER graphs and a regular ring lattice. The Watts-Strogatz model uses three tunable parameters, namely $n, k, p$. The procedure for creating such a graph is given below:

- First create a ring over $n$ number of nodes.

[1] This starting condition can differ based on the implementation used. For BA graphs this is igraph [29] and for HK graphs networkx [30]

- Each node inside it is joined to its $k$ nearest neighbors.
- Shortcuts are created by replacing edges such that for each edge $(v, u)$ in the ring with probability $p$ replace it with a new edge $(v, w)$, where w is chosen uniformly from $V \setminus \{v\}$.

A small example of Watts-Strogatz graph is shown in Figure 10.

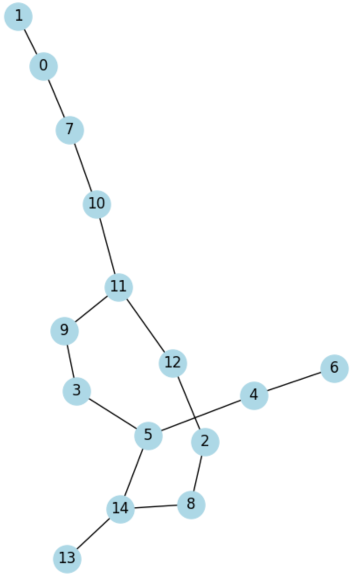


Figure 10: Example of Watts-Strogatz graph with $|V| = 15$ and $k = 2, p = 0.5$. Figure taken from [14].

Due to the way a graph is constructed following the Watts-Strogatz model, we start with a network that has a high clustering coefficient due to the regular lattice structure, and as $p$ increases and random rewiring occurs, the clustering coefficient decreases. As for the degree distribution, it initially follows that of a regular lattice, but as $p$ increases, it gradually shapes into a Poisson distribution.

# 4 Approach

The previous section discussed various measures of determining whether two nodes are equivalent or not. Such nodes share an equivalence relation that can be used to partition them into equivalence classes, and consequently compute their anonymity. The measures discussed are degree, isomorphism, and automorphism, with the first being the weakest, and the latest the strongest. However, it is not preferred due to its need for symmetry in the graph, which is unlikely to be prevalent in a real network.

The work of this thesis is an extension of the *d-k*-anonymity from [14] in the case of $d = 1$ on labeled graphs. This particular case also behaves the same as *1-neighborhood* isomorphism in [10]. This section thereafter will explain how the *d-k*-anonymity algorithm works, how the approach is extended on labeled graphs, and finally how the measure can be computed.

## 4.1 Definition of $d - k-$anonymity

The measure we focus on in this thesis considers the nodes equivalent if their neighborhoods are isomorphic and they have the same structural position in their respective $d$-neighborhoods. A formal definition from [14] is given below.

**Definition 4.1** ($d$-$k$-anonymity)**.** Given a graph $G = (V, E)$ and a distance $d > 0$, two nodes $v_1, v_2 \in V$ are equivalent with respect to $d$-$k$-anonymity if the following two properties hold:

- There is at least one isomorphism between the $d$-neighbourhoods of nodes $v_1$ and $v_2$
- There is at least one such isomorphism in which $v_1$ is mapped onto $v_2$ (see Definition 3.14)

The parameter $d$ allows the approach to adjust the strictness of the measure, and it also satisfies that for two nodes to be equivalent under $(d + 1) - k-$anonymity they need to be equivalent under $d$-$k$-anonymity for every $d$ value. The proof is shown in[16].

As mentioned in Section 3.4, we consider a vertex to be $d$-$k$anonymous if it is in an equivalence class, with respect to $d$-$k$-anonymity, of size $k$. Similarly, a graph is $d$-$k$anonymous if all equivalence classes in the graph are at least of size $k$, thus making each vertex $d$-$k$anonymous.

## 4.2 Computing $d$-$k$-anonymity

Let $G = (V, E)$ be a graph, $d$ be a distance and an equivalence relation. We want to partition the nodes into equivalence classes based on that relation. Thus, each class can contain one or more nodes $v \in V$. A profoundly simple approach, which we refer to as *naive algorithm*, would be, to begin with the class that contains all the nodes and group them with respect to an equivalence relation. The process is as follows: each node is compared to a node in an equivalence class until it finds the one it belongs in, and if it does not fit in any already existing one, a new equivalence class is created. The process is finished when the starting class is empty. This is done based on Definition 4.1. The remainder of this subsection first briefly describes the algorithms proposed in [14], and then explains the implementation of labeling on $d$-$k$-anonymity.

The computation of $d$-$k$-anonymity for each node requires an isomorphism check. To do so, we can use the notion of canonical labeling introduced in [31]. Canonical labeling can be viewed as a function that transforms a graph into a canonical form such that all isomorphic permutations of the graph will have the same canonical form. Thus, to determine if two graphs are isomorphic, one can just compute their canonical forms and if they are the same, then they are isomorphic. A strict definition of canonical labeling is given below:

**Definition 4.2** (Canonical labeling)**.** A canonical labeling is a function $C$ that transforms the graph into its canonical form such that for two given graphs $G = (V, E)$ and $G' = (V', E')$, any automorphic function $\gamma$ and graph $G^\gamma$ to which this function $\gamma$ is applied, the following two properties hold:

- $C(G^\gamma) = C(G)$
- $C(G) = C(G')$ if $G$ is isomorphic to $G'$ (see Definition 3.14)

Although with the help of canonical labeling we can determine whether two graphs are isomorphic or not by mapping the nodes of one graph to the other, we also need to see if there is one that maps the nodes onto each other.

This can be done by using the orbits of the $d$-neighborhoods. A pseudocode of the naive approach from [14] is given in Algorithm 1.

**Algorithm 1** NAIVE $d$-$k$-anonymity

```
 1: Input: Graph $G = (V, E)$, equivalence class eq = $V$, distance $d$
 2: eq_partition_new = {}
 3: if $d \leq 0$ then                                        ▷ Distance 0, no information is known
 4:     return{eq}
 5: end if
 6: for $v_1 \in$ eq do                                       ▷ For each vertex, find or create new eq
 7:     sub1 = get_neighborhood($G, v_1, d$)
 8:     cansub1 = $C$(sub1)
 9:     is_new_class = false
10:     for eq$_2 \in$ eq˙partition˙new do                    ▷ Check if vertex $v_1$ fits in already existing class
11:         $v_2 = v \in$ eq$_2$                               ▷ Get a vertex from $eq_2$
12:         sub2 = get_neighborhood($G, v_2, d$)
13:         cansub2 = $C$(sub2)
14:         if are_same(cansub1, cansub2, $v_1, v_2$) then     ▷ Check two equivalence criteria
15:             eq$_2$ = eq$_2 \cup v_1$
16:             is_new_class = true
17:             break
18:         end if
19:     end for
20:     if !is_new_class then                                 ▷ Vertex gets new class
21:         eq_partition_new = eq_partition_new $\cup \{v_1\}$
22:     end if
23: end for
24: return eq_partition_new
```

The main idea of the algorithm is the following: first, we input a graph and a set (equivalence class) of all its nodes (line 1). We iterate over all the nodes $v_1 \in V$, find their $d$-neighborhood, and compute the canonical labeling of it. Next, take a node $v_2$ from an equivalence class, do the same thing, and then compare the two canonical labelings to see if the 1−neighborhood of node $v_1$ is isomorphic to the graph of node $v_2$.

More specifically, we first check if the diameter $d$ of the graph is larger than 0 (line 3) because if not, all the nodes are considered equivalent and the algorithm is done. The for loop in lines 6 to 23 is used to iterate over all the nodes of the graph, and partitions them into equivalence classes. The right equivalence class for each node is determined with the for loop in lines 10-19. The computation of the neighborhoods and their canonical labeling is done in lines 7, 8, 12, and 13. We check for isomorphism that satisfies the two requirements on Definition 4.2 and if they hold for the two nodes, we add the new node to this class and continue with the next (lines 14-17). Otherwise, the algorithm continues the loop by comparing $v_1$ to a node in the other equivalence classes one by one, and if it is compared to all the existing equivalence classes and is still not partitioned, then a new class is created (lines 20-21). The algorithm is finished when every node is partitioned.

### 4.2.1 Colored $d$-$k$-anonymity

For the purposes of this thesis, the approach is modified such that we can work with labeled graphs. This time apart from the isomorphism checks, we also need to know if the colors of the corresponding nodes in the two graphs are the same. In order to do so, we create a dictionary that maps the nodes of the graph to a certain color. The coloring we use is binary and the function works as given in Definition 3.10.

Since the work of this thesis is an extension of the $d$-$k$-anonymity measure introduced in [14] of which the implementation has been done in C++ using the ***nauty*** framework [32], that is the case for this work as well. Nauty offers comprehensive computational capabilities facilitating the possibility of obtaining the orbits of nodes of a graph and the computation of canonical labeling of graphs. This computation can also be applied to colored (labeled) graphs. However, due to limitations, extra steps are needed for the requirements of the implementation of the $d$-$k$ algorithm. Since nauty is able to do an isomorphism check on labeled graphs by itself, we do not need to compare the color of every node. However, due to practical issues with the software (see Appendix 8.1 for a more detailed explanation of the implementation), we need to check the color of the first node of both graphs. A pseudocode is given in Algorithm 2.

**Algorithm 2** LABELED NAIVE $d$-$k$-anonymity

```
 1: Input: Graph $G = (V, E)$, equivalence class eq = $V$, distance $d$, dictionary $c$ ▷ $c$ contains the color for each node
 2: eq_partition_new = {}
 3: if $d \leq 0$ then
 4:     return{eq}
 5: end if
 6: for $v_1 \in$ eq do
 7:     sub1 = get_neighborhood($G, v_1, d$)
 8:     colors1 = get_colors(sub1, $c$)
 9:     firstElement1 = the first element of list colors1
10:     cansub1 = $C$(sub1)
11:     is_new_class = false
12:     for eq$_2 \in$ eq_partition_new do
13:         $v_2 = v \in$ eq$_2$
14:         sub2 = get_neighborhood($G, v_2, d$)
15:         colors2 = get_colors(sub2, $c$)
16:         firstElement2 = the first element of list colors2
17:         cansub2 = $C$(sub2)
18:         if are_same(cansub1, cansub2, $v_1, v_2$) and firstElement1==firstElement2 then
19:             eq$_2$ = eq$_2 \cup v_1$
20:             is_new_class = true
21:             break
22:         end if
23:     end for
24:     if !is_new_class then
25:         eq_partition_new = eq_partition_new $\cup \{v_1\}$
26:     end if
27: end for
28: return eq˙partition˙new
```

This time we have added a dictionary that contains the colors for every node as input for the algorithm. We create a list of the colors of the nodes of the neighborhood produced from lines 7 and 14 according to nauty format (lines 8 and 15). We store the first element of each list (lines 9 and 16). Because of the format of the lists that nauty uses to perform isomorphism checks (see Apendix 8.1), we only need to see if the colors of the nodes match for the first element of each list. With the addition of the equivalence condition in line 18 we can make sure that the colors of the nodes in the two graphs match. Thus, we have a more strict condition for isomorphism.

## 4.3 Iterative computation of $d$-$k$-anonymity

One can easily see that by nature, the naive approach can be very expensive since in the worst-case scenario each node is compared to all the equivalence classes, resulting in $O(|V|^2)$. This effect is countered by introducing an iterative approach that starts with distance $d = 1$ and iteratively increases it by one until it equals the chosen distance $d$. In this case, we use the property that two nodes are considered equivalent with respect to $(d + 1) - k-$anonymity, if they are equivalent with respect to $d$-$k$-anonymity. A pseudocode of the approach is given in Algorithm 3 taken from [14].

**Algorithm 3** ITERATIVE-$d$-$k$-anonymity

```
Input: Graph $G = (V, E)$, distance $d$
eq_partition = $\{\{0, 1, \ldots, |V|\}\}$        ▹ Equivalence partition contains one class
for $i = 1$ to $d$ do
    for eq in eq_partition do        ▹ Iterate over the classes, split them further
        eq_new = NAIVE-$d$-$k$-anonymity($G$, eq, $i$)        ▹ Call function in Algorithm 1
        eq_partition_new = eq_partition_new $\cup$ eq_new        ▹ Insert all the new classes
    end for
    eq_partition = eq_partition_new
    eq_partition_new = $\emptyset$
end for
return eq_partition
```

# 5 Data and Experiments

In Section 3.5 we gave a brief description of the three graph models and their properties used for our experiments. A more in-depth look is given in [14], which goes into more detail about the structural properties of the graphs such as 1) the number of components, the size of the largest component and diameter, and 2) the degree distribution and the triangles per vertex, over the average degree. This Section is focused on the evolution of the graphs.

The remainder of the section is structured as follows: first, a brief description and properties of the real networks are given so that the reader gets a better idea of their structure. Lastly, we discuss the criteria the evaluation is based on our experiments, and how we chose to combat randomness for the graph models experiments.

## 5.1 Graph models

The evolution of the graph models is related to the average degree in the case of the ER models or the value $m$ in the case of the BA and WS models. For the BA models $m$ accounts for $m$ different nodes that each new node is connected to. These nodes are plausible to connect to nodes with a high degree. This has as a result a skewed degree distribution that is often observed in empirical networks. As regards the WS models, $m$ accounts for small average path lengths and clustering. These graphs are generated by first connecting the nodes in a circle and then connecting each node to its $m \geq 2$ nearest neighbors. Next each edge is rewired with probability $p = 0.5$, which is also used in [14] and is a common value used for this type of example. Working with graph models they can have control over their densities, by tuning a parameter. The concept of increasing the number of edges while retaining the number of nodes constant in ER graphs is referred to *evolution of random graphs* in [17]. The values of $m$ we experiment with in this thesis are described as powers of number 2. More specifically, $m \in \{2^i | i = 1, 2, 3, \ldots, 8\}$, where $2^1 = 2$ accounts for a very sparse graph and $2^8 = 256$ for a very dense graph.

In the case of ER models, the way the edges are assigned is completely random. Thus, there is no preference about which edges should be added to the graph. In an attempt to explain their evolution, [17] describes it using five phases they go through. The phases are given below:

1. The graph only contains trees
2. Cycles are formed
3. The structure is reshaped abruptly making small components melt into a giant component with a more complex structure. Other components remain relatively small
4. All nodes tend to become connected with an almost certainty
5. The graph becomes asymptotically regular. Finally, the structure begins to stabilize and maintain the same structure until it is fully connected

## 5.2 Real-world networks

Here we will discuss the reasons behind the choice of our real-world networks. An interesting observation made in [14] is that anonymity tends to decrease the more we increase the density of a graph. That is why our choice of real-world networks consists only of sparse graphs. Furthermore, a notice should be made that directed graphs are turned into undirected ones for the purposes of the experiments since the implementation of the algorithm is only built for undirected graphs at this stage of the research.

Before moving on to the methodology followed for this thesis, we give a brief description of the datasets we experimented with and some structural properties of the networks they represent. The Copenhagen networks [33] dataset consists of three different networks, namely fb, sms, and calls. The twitch social networks [34] consists of four different networks, each one corresponding to a different country. More specifically, we have the PTBR networks which corresponds to Portugal, the ES network for Spain, the ENGB network for England, and the DE network for Denmark. The last dataset, Deezer [35] consists only of 1 network. Furthermore, we present a table for each dataset to give information about them regarding the number of nodes ($|V|$), the number of edges ($|E|$), the density of the networks, the gender of the individuals represented by 1s and 0s, and their directionality.

### 5.2.1 Copenhagen networks (Copnet)

Copnet consists of social networks that are built using data sampled from the social interactions of university students in Copenhagen. This particular study spanned a duration of four weeks, and the data was sampled every five minutes. These social interactions consist of phone calls, text messages, and information about Facebook friendships. The reasons why we chose this particular dataset are that, first, it includes three sparse networks with a relatively small of nodes, and second, it contains information about the individuals (gender) that we can use to label the nodes of the graph. In Table 1 we present some structural properties of the networks.

| Network | $\lvert V\rvert$ | $\lvert E\rvert$ | Density $\simeq$ | % of 1s | % of 0s | Directionality |
|---|---|---|---|---|---|---|
| sms | 568 | 24,333 | 0.07 | 21% | 79% | unidrected |
| calls | 536 | 3,600 | 0.01 | 23% | 77% | directed |
| fb | 800 | 6,429 | 0.01 | 24% | 76% | directed |

Table 1

*Note:* It is observed that not all the nodes in the Copnet dataset [33] have a gender assigned to them. More specifically, an amount of 25 (4%), 16 (3%), and 33 (4%) nodes in sms, calls, and fb are missing respectively. In order to fill all of them, we produced one using a Uniform distribution of $r = 0.5$, for each node that is missing the gender.

### 5.2.2 Twitch social networks

This particular dataset is built using data sampled from Twitch users, such as gamers that stream while they are playing, in a certain language. The data was collected in 2018 and it is primarily used for node classification and transfer learning. The nodes represent the users themselves and the edges represent mutual friendships within them. The reasons why this dataset is chosen are first, the networks contained are sparse and the number of nodes varies from around 2,000 to 10,000 (see Table 2), second, the graphs are undirected which means that the implementation can work on them without further work needed for the setup of the experiments, and last, the dataset contains information about the users (maturity) that can be used to label the nodes of the graph.

| Network | $\lvert V\rvert$ | $\lvert E\rvert$ | Density $\simeq$ | % of 1s | % of 0s | Directionality |
|---|---|---|---|---|---|---|
| DE | 9,498 | 153,138 | 0.003 | 60% | 40% | undirected |
| ENGB | 7,126 | 35,324 | 0.002 | 55% | 45% | undirected |
| ES | 4,648 | 59,382 | 0.006 | 29% | 71% | undirected |
| PTBR | 1,912 | 31,299 | 0.005 | 34% | 66% | undirected |

Table 2: Real-world networks for experiments. Here DE stands for Denmark, ENGB for England, ES for Spain and PTBR for Portugal.

### 5.2.3 Deezer network

The last dataset we experimented with is Deezer, which is based on the Deezer music streaming platform. This particular dataset was built in 2020 based on collected data from users of the public API. The nodes represent users from European countries and the edges are mutual follower relationships within them. The dataset is primarily used for the task of binary node classification to predict the gender of the users. This feature is retrieved from the name field of each user. The reasons why we chose to experiment with this network are that: first, it is a sparse graph with a lot more nodes compared to the rest of the networks (28,281), it is undirected, and it contains information (gender) of the users, useful to label the nodes of the graphs. In Table 3 we give the structural properties of the Deezer network.

| Network | $\lvert V\rvert$ | $\lvert E\rvert$ | Density $\simeq$ | % of 1s | % of 0s | Directionality |
|---|---|---|---|---|---|---|
| Deezer | 28,281 | 97,752 | 0.0002 | 43% | 57% | undirected |

Table 3

## 5.3 Experimental methodology

This thesis aims to examine how anonymity on unlabeled graphs is compared to labeled graphs when distance $d = 1$. For our experiments, we designed two different types of binary labeling. The first is random and the second is centrality-based. In the case of graph models, we used 5 different splits of the nodes for the random labeling, namely $10\% - 90\%, 20\% - 80\%, 35\% - 65\%, 45\% - 55\%$ and $50\% - 50\%$, by using a uniform distribution as mentioned defined in Equation 9. We conducted 10 experiments per graph using a unique seed for the generation of the graphs and in order to combat randomness in our results, we computed the minimum, average, maximum, and SEM (Standard Error of the Mean). More information and graphs are given in the next section and in Appendix 8.2. In the case of real-world networks, instead of using random labeling, the gender or maturity of the user provided in the datasets represented as 0s and 1s are utilized to label the nodes of the graph. This extra information provided from the datasets gives us the tool to test in practice the theoretical scenario discussed in Section 1. That means a potential adversary has an extra tool at his disposal. Apart from information about the structural properties of the nodes, now they contain information about their gender or whether the users are represented by nodes are mature or not. Regarding the centrality-based labeling, we split the nodes such that the $1\%, 5\%$, and $10\%$ of the nodes with the highest centrality value have the label 1, and the rest the label 0. The centrality measures we experimented with are *Closeness, Betweenness,* and *Degree* centrality.

## 5.4 Evaluation

In our work, we use the $1$-$k$ variant of the $d$-$k$-anonymity algorithm, narrowed down to undirected graphs. The algorithm partitions its nodes into classes based on isomorphism checks of the $d$-neighborhood between two nodes $v, w$ of the graph, with $k$ indicating the size of the classes meaning that the higher the value of $k$ of a class, the less likely the nodes in that class are to be re-identified. Our evaluation is based on the fraction of nodes that belong in the $k = 1$ and $k = 2$ classes.

Moreover, assigning labels on the nodes such as gender, gives a potential attacker an extra tool at their disposal apart from the structural properties of the nodes. Thus, to determine how anonymity is affected, one can focus on measuring the ratio of the number of unique nodes over the total number of nodes on the graph.

# 6 Results

First, we discuss the results from conducting experiments on the three graph models we have chosen regarding the ratio of unique nodes and the nodes in $k = 2$ classes. Next, we discuss the results from the experiments done on real networks. We attempt to see how anonymity is affected when one adds labels on the nodes of the graph in comparison to unlabeled graphs, and how that is related to the structure of the graph (sparse or dense graph).

The experiments on both the graph models and the real networks are conducted on a machine with 64 Intel Xeon E5-4667v3 cores @ 2.00GHz (128 threads), 1TB RAM and 9TB SSD local storage.

## 6.1 Anonymity in graph models

In this section, we will visualize and attempt to analyze the results in order to get assumptions and reasoning behind them. Due to the fact that there has already been significant work and analysis in [14] regarding the anonymity distribution of $d$-$k$-anonymity and how it tends to reshape with the increase of average degree, in this thesis we primarily shift our focus on discussing the results by examining the effect of adding labels on the nodes of the graph has in $1 - k-$anonymity.

For the experiments on graph models, we produce three types of graphs, namely ER, BA and WS of 1,000 and 10,000 nodes, by tuning a parameter $m$, where $m \in \mathcal{S} = \{2^i | i = 1, 2, 3, \ldots, 8\}$. In the case of ER graphs $m$ accounts for the average degree, in BA it accounts for $m$ different nodes that each node is connected to, and in WS it accounts for small average path lengths and clustering. For this purpose, we used a unique random seed for each graph model. As regards the random labeling applied, we did 10 independent experiments for each value of probability $r \sim Uni(0, 1)$ (see Section 3.3). The probabilities we used are [0.1, 0.2, 0.35, 0.45, 0.5] which split the nodes into $[10\% - 90\%, 20\% - 80\%, 35\% - 65\%, 45\% - 55\%, 50\% - 50\%]$ partitions accordingly.

Here, we will present the results for both graph models and real networks.

In order to present and visualize our results in the case of the graph models, we used line plots to illustrate anonymity within the graphs. First, we will be giving the ratios of $k = 1$ classes in both 1,000 and 10,000 nodes, and next, the ratios of $k \leq 2$ classes. As regards the real networks, we used a bar plot. The rest of the section goes into more detail.

### 6.1.1 Graphs of 1,000 nodes

Here we will present the results from the experiments conducted on graph models of 1,000 nodes, and then we will discuss our findings and their implications. The first graph model that we experimented on is the Erdős–Rényi model.

#### 6.1.1.1 Anonymity in Erdős–Rényi graph

The first graph model we are going to discuss is Erdős–Rényi. Figure 11 shows the ratio of nodes that belong in $k = 1$ classes over the total number of nodes, meaning that they are unique. The horizontal axis denotes the values of the average degree of the graph. The vertical axis denotes the fraction of unique nodes and employs a logarithmic scale to make the results more distinguishable. This type of plot will be used throughout the discussion of the results in graph models that show this particular fraction of nodes.

To begin with, it is evident that after having labeled the nodes of the graphs, the number of nodes that belong in $k = 1$ classes - representing unique nodes - is larger than the corresponding number without labels in the cases where average degree $\leq 16$. In the cases where the average degree is larger than 16, we can observe that all the nodes of the graph are unique even prior to labeling. Such results are to be expected since the graph is very dense but we are interested to see to what extent the results would differ between unlabeled and labeled graphs.

In the case of average degree equal to 2 where the graph is very sparse, most of the nodes are equivalent, even after applying labeling on the nodes of the graph. The same seems to be the case for an average degree equal to 4 as well, with a larger number of unique nodes being found after applying random labeling of equally splitting the nodes $(50\% - 50\%)$.

For the average degree equal to 8 and 16, there seems to be a more distinguishable pattern. That is, the fewer the number of nodes that have the same color, the fewer the nodes that are unique. One can see that this number is at its lowest when we split the nodes according to their centrality, to 1% and 99%. Following that, slightly above we have the $5\% - 95\%$ split of the nodes according to their centrality. The number of unique nodes is increased by

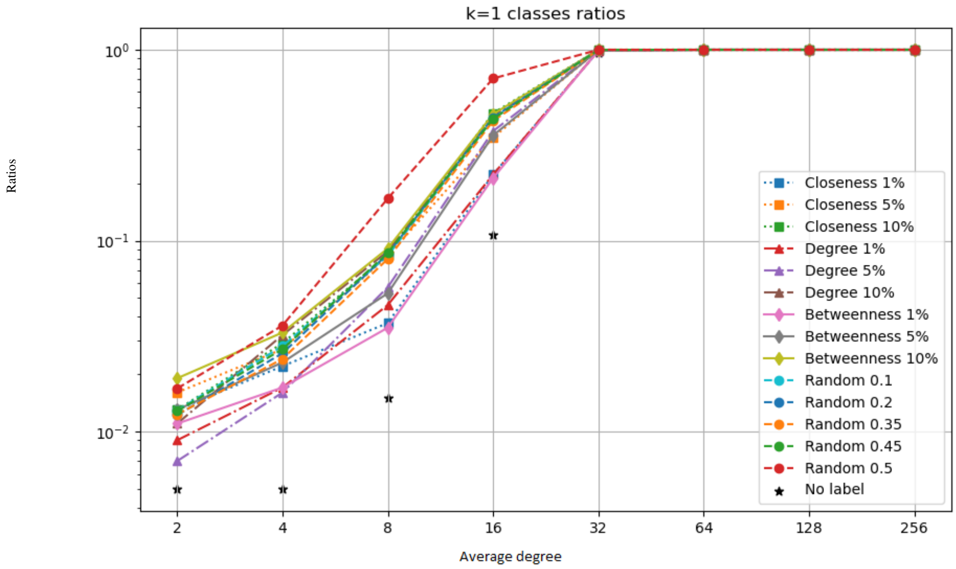


Figure 11: The ratio of $k = 1$ classes over 1,000 nodes for the ER graph model. Each marker corresponds to a type of binary labeling. Scattered points indicated by black asterisks show the ratio of nodes derived using $d$-$k$-anonymity without labeling.

approximately half an order of magnitude. Finally, the number of unique nodes derived from a $10\% - 90\%$ split seems to be the same as the Random split with the same percentage, with the increase being slightly increased once again. That seems to indicate that the more nodes there are that have the same color, the fewer unique nodes we get.

As for average degree $\geq 32$, we can see that all the nodes of the graph are unique, even without labels on them, thus the labeling does not have an effect on anonymity. Lastly, an interesting observation is the abrupt increase of $50\%-50\%$ Random labeling to the rest, with the values experiencing a considerable surge. In order to further examine this abrupt jump, we experimented with four more values for the probability $r$ of the Random labeling, namely $0.46, 0.47, 0.48, 0.49$, to see where exactly the jump happens (see Appendix 8.3). The results seem to indicate that it happens at 0.46 with the rest of the values to give approximately the same results as the lines tend to closely align. Further discussion on this particular observation later in the following section of the graph models with 10,000 nodes.

Similar results are observed when looking at the number of nodes that belong in classes of size $k \leq 2$. Figure 12 shows similar patterns with Figure 11. There is a resemblance in the pattern observed in both plots regarding the number of unique nodes resulting from labeling the nodes of the graph when the average degree is equal to 8 or 16. This time it is more distinguishable that the number of unique nodes that occur from $1\% - 99\%$ centrality-based split is slightly less than the $5\% - 95\%$ split, which is less than the $10\% - 90\%$. Moreover, the abrupt jump of the Random $50\% - 50\%$ seems to be the case here as well.

Of particular note is the marked increase in the number of nodes resulting from labeling in the case of an average degree equal to 2. Figure 12 depicts all types of labeling that seem to have approximately the same result which is they all seem to be half an order of magnitude larger than without labeling, as opposed to Figure 11.

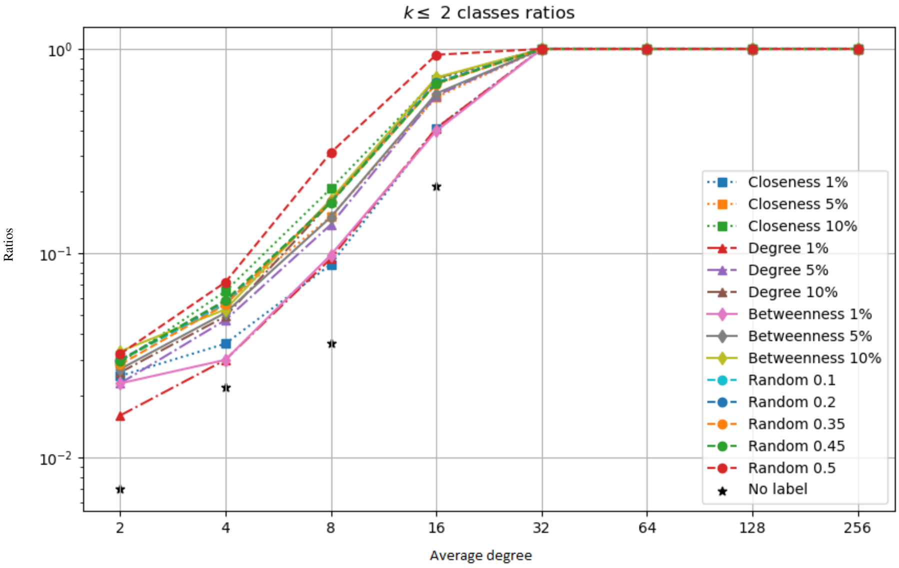


Figure 12: The ratio of $k \leq 2$ classes over 1,000 nodes for the ER graph model

#### 6.1.1.2 Anonymity in the Barabási-Albert model

Here we discuss the results of the Barabási model. We use the same kind of line plot with a logarithmic scale on the y-axis as well. The ratios of $k = 1$ nodes are given in Figure 13. The graph illustrates similar patterns in general as in the ER model. Notably, a significant deviation is observed in the rate of anonymity decrease. Anonymity decreases as a function of m happens faster, which seems to be the case from the early stages where $m = 2$. This is probably a result of a large diversity in degrees and triangles per vertex. A more in-depth analysis can be found in [14].

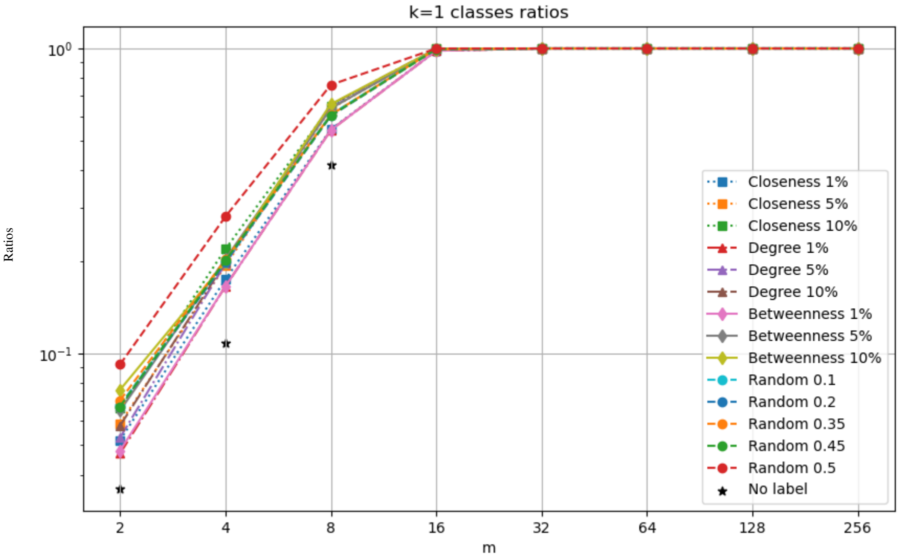


Figure 13: The ratio of $k = 1$ classes over 1,000 nodes for the BA graph model

To provide a more detailed insight, the initial observation one can make is that the proportion of $k = 1$ nodes without labels initiates slightly above 0.001. This has as a result the faster decline of anonymity where as $m$ increases to 4 and 8. That sharp jump is approximately one order of magnitude, and in the case of $m = 16$, all the nodes are unique. Interestingly, the disparity between pre-labeling and post-labeling remains relatively consistent. The number of unique nodes demonstrates a steady incline. Furthermore, the pattern where the 1% – 99% split results in the fewest unique nodes, with the 5% – 95% split being slightly more, with subsequent splits displaying a similar pattern.

Last but not least, a similar pattern observed is the clear distinction of the Random 50% – 50% split from the rest of the subsequent splits. The proportion of $k = 1$ nodes seems to be higher than the rest reminiscent of the ER model but that maybe due to $m$ accounting for different things for the BA and ER models. In the following section where we discuss the graph models with 10,000 nodes, we notice that such a sharp increase is not evident, which might indicate that the observed behavior may be attributed to specific structural properties. Now, we will give the results from the proportion of nodes that belong in $k \leq 2$ classes in Figure 14.

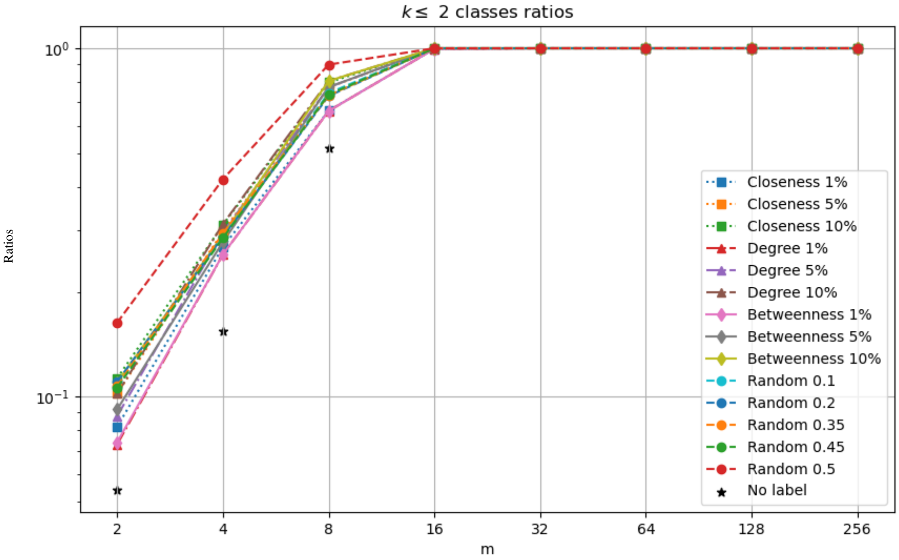


Figure 14: The ratio of $k \leq 2$ classes over 1,000 nodes for the BA graph model

The proportion of nodes that belong in classes of $k \leq 2$ is slightly higher than the $k = 1$. Thus, no notable disparity between the results of $k = 1$ and $k \leq 2$ is discernible, warranting further analysis.

#### 6.1.1.3 Anonymity in Watts-Strogatz

The last graph model we experimented with is the WS model. Here we notice a decrease in anonymity when looking at the proportion of unique nodes compared to the ER model for the values of $m$ larger than four. The results are shown in Figure 15.

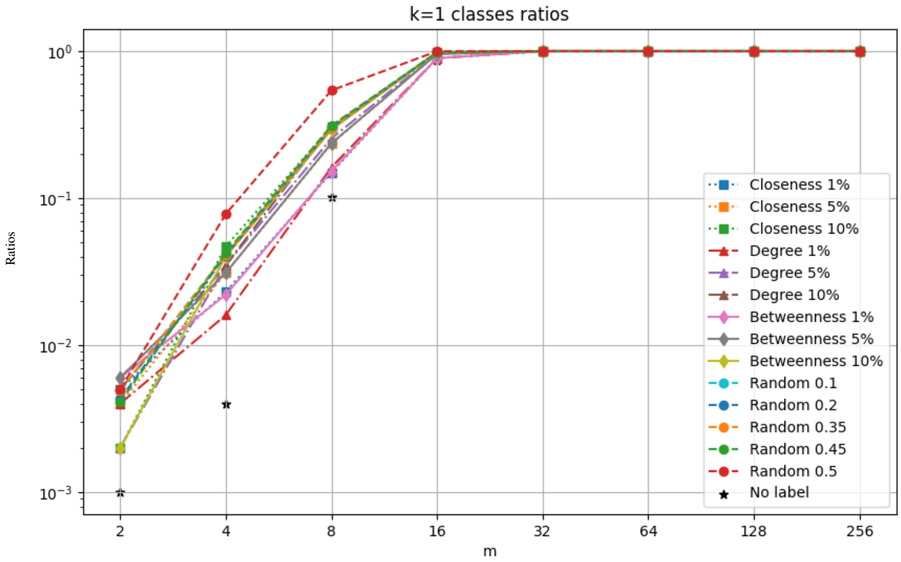


Figure 15: The ratio of $k = 1$ classes over 1,000 nodes for the WS graph model

To be more precise, we can see that for $m$ equal to 8 and 16, there is a notable surge of unique nodes increasing by a margin of approximately one order of magnitude, which results in all the nodes being unique when $m = 16$. Moreover, similar trends in labeling patterns are also evident here, wherein the splits that have fewer nodes with a different color resulting in fewer unique nodes, and the Random 50% – 50% split having more unique nodes than the subsequent splits.

Conversely, a significant change observed is the decrease in the number of nodes that fall in $k = 1$ classes when $m = 2$. More specifically, we can see that it is less than the one in Figure 11 by approximately half an order of magnitude, ranging from 2 to 6 unique in the case of labeling and only one unique node without labeling out of 1,000 nodes. That is particularly interesting observation, especially when compared to the results from the ER and BA models. The cause of this change could be the result of the structural properties in WS model when it is highly sparse. As described in Section 3.5.3, the graph initially starts as a ring and the edges are then rewired according to a probability $p$ given. Thus, when we keep the average degree of the graph to 2, the nodes of the graph can be split into categories according to their structural positions:

- Nodes that still maintain the ring structure
- Nodes that tend to form a line graph
- Nodes that belong in the ring are also the start of the line graph
- Leaf nodes with degree one

This can be seen in Figure 10 where nodes 2, 5, 8, 14, 3, 9, 11, 12 form a ring, nodes 11, 10, 7, 0, 1 form a line graph, nodes 5, 4, 6 form another line graph, with nodes 11 and 5 being the start of the line. Lastly, nodes 13, 6 and 1 are leaf nodes with degree value 1. As a result, the equivalence between the nodes heavily relies on the degree of the nodes, making most of the nodes equivalent and a small portion of them unique.

It is worth noting that the proportion of $k = 1$ nodes in the cases of Closeness and Betweenness centrality based 10% labeling is less than the 5% when $m = 2$. This difference is a relatively small one since the 5% gives us 40 unique nodes and the 10% gives us 20 unique nodes. Although it might be coincidental, our intuition behind it is the following. The most central nodes tend to be close to each other and with one or more of them being the nodes that link the ring with the line graphs. Thus, a slight change in the coloring of the nodes can of course lead to a decrease in anonymity. When we slightly increase the percentage of different colored nodes, there is a small chance that we get marginally improved results because we might change the color to central nodes that are connected to nodes that became unique after we applied a lower percentage labeling on the nodes. Going back to the example of a WS model given in Figure 10, we now give two colored versions of the graph in Figure 16 that show the betweenness and closeness centrality distribution with the darker colors indicating nodes with higher centrality value, and the lighter colors the opposite.

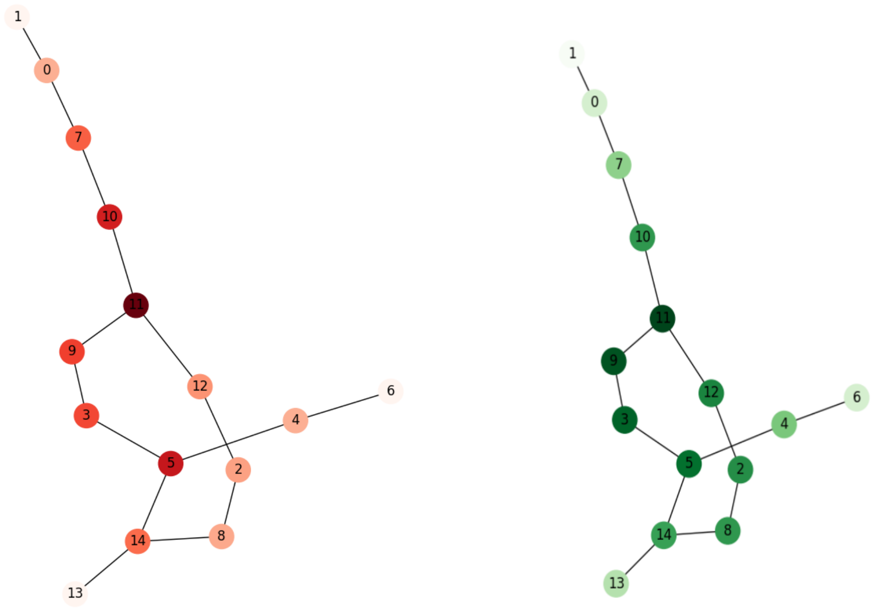


Figure 16: Colored version of the Watts-Strogatz graph used as an example in Figure 10. The color red shows the betweenness centrality distribution and the color green shows the closeness centrality distribution of the nodes.

In Figure 16 we can see that if we label the nodes such that node 11 has a unique label and the rest of the nodes share another common label, node 5 is unique. However, when we give node 5 the same label as node 11, the node is no longer unique as it is equivalent to node 11. This is just a small example to help us showcase our intuition behind the cause of the results. Now, we will present the results for the ratios of $k \leq 2$ classes in Figure 17.

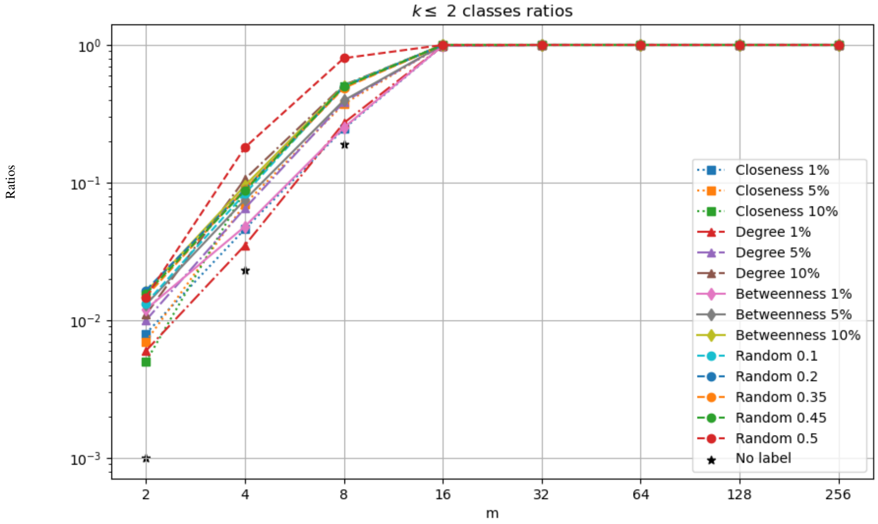


Figure 17: The ratio of $k \leq 2$ classes over 1,000 nodes for the WS graph model

The graph depicts minor changes compared to Figure 15 when $m = 8$. What is important to highlight is that for $m = 4$ the number of unique nodes is increased by an order of magnitude for both before labeling and all types of labeling. This effect might be related to the structure of the WS model which has started to evolve and many nodes do not have a degree value of 2 as is the case for $m = 2$. Thus, large classes can collapse and produce more unique nodes.

The most important aspect that should be highlighted in the plot is the number of unique nodes for $m = 2$. This has only changed by one node in the case of no labeling, and a few for all the types of labeling. This should be the effect of the structure of the WS model which makes the majority of the nodes to be equivalent with respect to $d$-$k$-anonymity.

### 6.1.2 Graph of 10,000 nodes

Now, we move to the experiments on graph models with 10,000. Here, we used the same experimental setup, with the only difference being the number of nodes on the graphs is now 10,000. We can now have a more clear picture of what is happening and make some more solid assumptions.

#### 6.1.2.1 Anonymity in Erdős–Rényi

The first graph model we experimented with was the Erdős–Rényi one. Figure 18 shows the results.

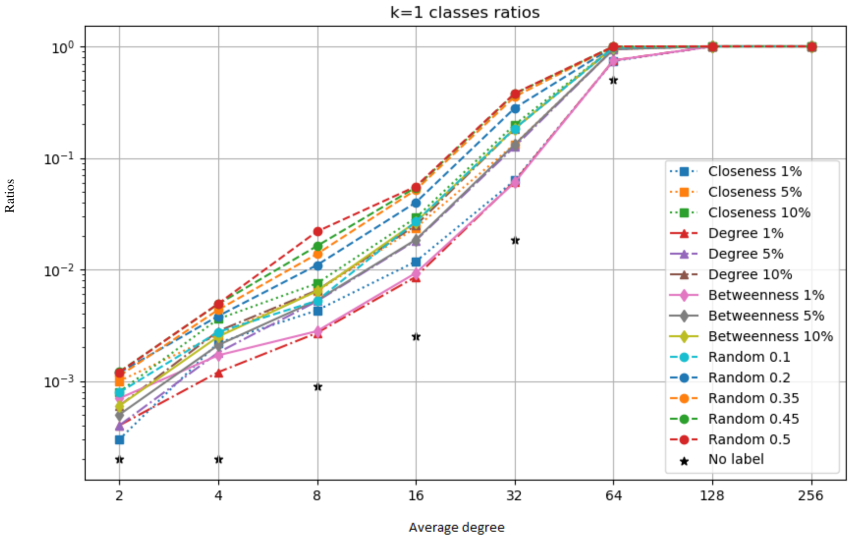


Figure 18: The ratio of $k = 1$ classes over 10,000 nodes for the ER graph model

The first and most striking difference that can be observed is the lower rate that anonymity decreases. While in the case of 1,000 nodes, the proportion of unique nodes without labeling is around the 0.005 margins when the average degree is equal to 2 and 4, in the case of 10,000 nodes it falls in around 0.0005, which is an order of magnitude less. The same behavior can be observed in the different types of labeling applied. This means that the number of unique nodes has remained the same, even after increasing the number of nodes in the graph from 1,000 to 10,000.

Following that, the findings depicted in Figure 18 show that anonymity decreases at a lower rate. In the case of 1,000 nodes, there is a jump from an average degree equal to 4 to 8, of half an order of magnitude, and from that point, until the average degree is equal to 32, the proportion of unique nodes is raised with a rate of approximately an order of magnitude. In the case of 10,000 on the other hand, though, this proportion seems to increase more steadily with a rate of half an order of magnitude throughout and gradually reaches the point where all the nodes are unique when the average degree is 128, which is 4 times more compared to the 10,000 nodes. The same findings apply to all the types of labeling we experimented with.

The cause behind this finding might stem from the density of the two graphs compared. We consider the formula given in Equation 5. If we choose to keep the average degree a constant $m_0$ and we only change the number of nodes, the graph with the larger number of nodes is more sparse. Thus, it is expected that the graph with 10,000 nodes will have fewer unique nodes than the graph with 1,000 nodes. That is our intuition behind the origins of the results.

On the opposite side, the patterns where the number of unique nodes occurs less often when the number of nodes that have different colors is larger is also apparent here, and the Random type of labeling produces more unique nodes as well. This time though, the Random 50% – 50% does not have such a sharp jump that makes it distinguished from the rest, and we can see that the Random 20% – 80% is slightly above the other types of centrality-based labeling. Following that, the more we increase the number of nodes with different colors, the more unique nodes tend to occur after applying the $d$-$k$-anonymity algorithm. We will now present the results for the fraction of nodes that belong in $k \leq 2$ classes in Figure 19.

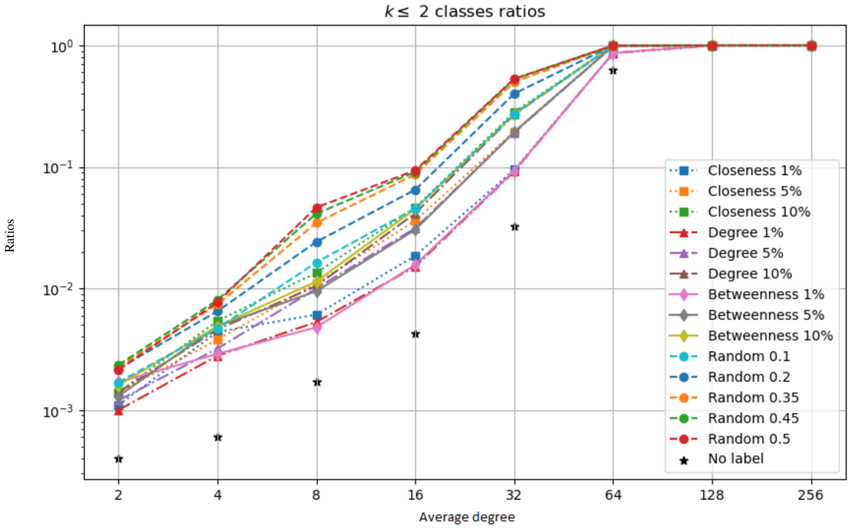


Figure 19: The ratio of $k \leq 2$ classes over 10,000 nodes for the ER graph model

We can see similar patterns with Figure 12 and Figure18). The number of unique nodes gradually increases with the average degree. The types of labeling with fewer nodes of different colors produce more unique nodes, and at an average degree of 64 almost all nodes are unique and anonymity is lost. Furthermore, the difference in the number of unique nodes in comparison to Figure 18 is at most half an order of magnitude which is observed when the average degree is equal to 8. In the rest of the cases, it is minor and not significant.

#### 6.1.2.2 Anonymity in Barabási

Akin to the results observed from the comparison of 1,000 and 10,000 node graphs cases for the ER model, the results for the BA model seem to follow a similar pattern with a small difference.

To begin with, the fraction of unique nodes without labels over the total number of nodes of the graph is half an order of magnitude less in the case of 10,000 nodes as shown in Figure 20 compared to the one for 1,000 nodes in Figure 13, when $m = 2$. More precisely, in Figure 20 we can discern that it is slightly less than 0.01, whereas in Figure 13 it is above the margin of 0.01.

The rate of the increase of unique nodes is approximately the same in both cases, as the case of 1,000 nodes seems to be slightly more than half an order of magnitude, compared to the case of 10,000 nodes which seems to be around half an order of magnitude. This has as a result the loss of anonymity (all nodes become unique) when $m = 16$ and $m = 32$, for the 1,000 and 10,000 node graphs respectively. A prominent difference though that is worth noting is that this time there is no sharp jump in the number of unique nodes produced from the Random 50% – 50% labeling as it is observed in Figure 13.

In Figure 21 we present the results for the $k \leq 2$ classes. We notice the same patterns as in Figure 20 with no significant differences. That means we do not have a remarkable rise of unique nodes. The results seem to be approximately the same as $k = 1$, just slightly higher.

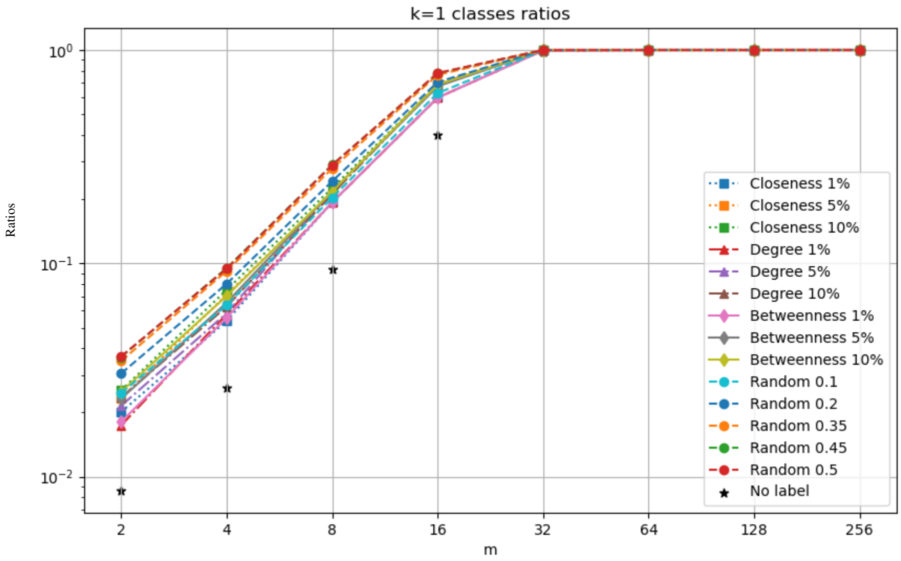


Figure 20: The ratio of $k = 1$ classes over 10,000 nodes for the BA graph model

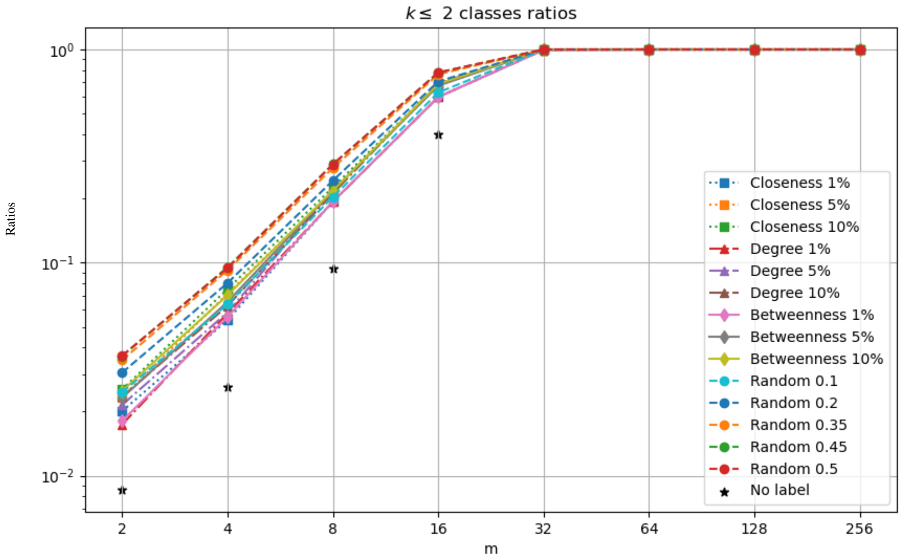


Figure 21: The ratio of $k \leq 2$ classes over 10,000 nodes for the BA graph model

#### 6.1.2.3 Anonymity in Watts-Strogatz

Lastly, we present the results for the WS model of 10,000 nodes in Figure 22.A common characteristic that both Figures 20 and 22 share is that when $m = 16$ almost every node becomes unique, and when $m = 32$ they all become unique.

It is imperative to note here that there are no unique nodes in the cases of without labeling and with Closeness 5% labeling. This observation is only made in this type of graph and the intuition behind it is again the overall structure of the extremely sparse ($m = 2$) WS model as explained in the case of 1,000 nodes in Section 6.1.1.3. Moreover, similar to the case of WS model with 1,000 nodes, the rest of the types of labeling yield 2 to 6 unique nodes.

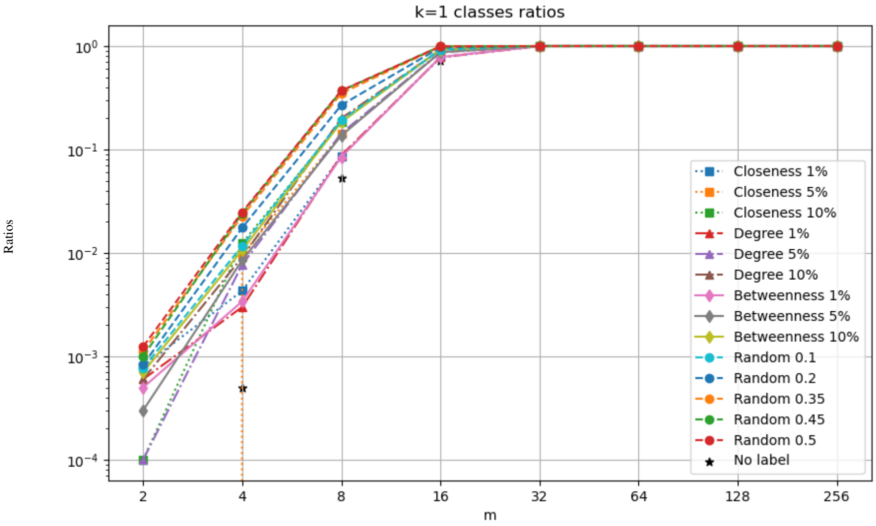


Figure 22: The ratio of $k = 1$ classes over 10,000 nodes for the WS graph model

On the other hand, several differences are also distinct. First, for $m$ equal to 2 and 4, the proportion of unique nodes is an order of magnitude less than in the case of 1,000 nodes. An interesting observation though is the jump from $m = 4$ to $m = 8$ which is approximately two orders of magnitude, making it have slightly fewer unique nodes than the graph of 1,000 nodes.

What is more, the lines indicating the types of labeling exhibit a clearer pattern compared to Figure 13, which makes it easier to discern the pattern that the types of labeling with more nodes of different colors tend to produce more unique nodes when the $d$-$k$-anonymity algorithm is applied on the graph. Now we will present the results for the $k \leq 2$ classes.

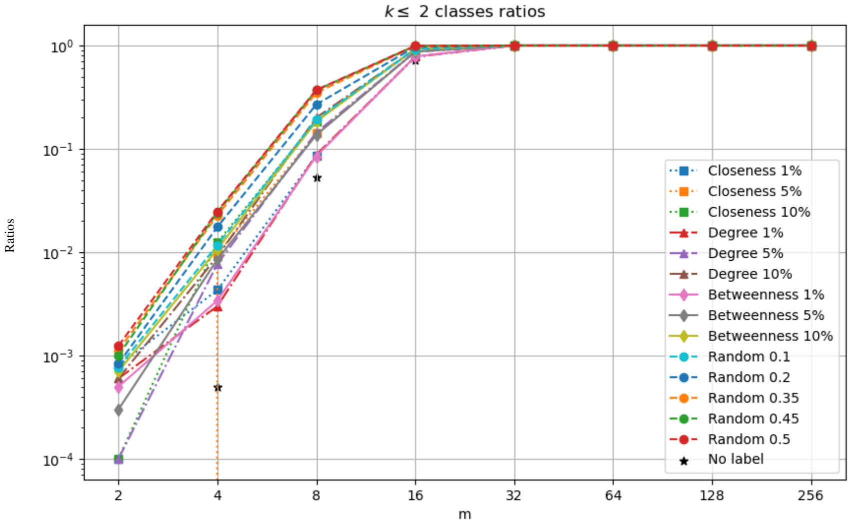


Figure 23: The ratio of $k \leq 2$ classes over 10,000 nodes for the WS graph model

Figure 23 depicts the same patterns as in Figure 22 with a marginal increase in the number of unique nodes when $2 < m \leq 8$. What is worth noting is the results yielded for $m = 2$. More specifically the number of unique nodes without labeling and with Closeness 5% labeling is again 0, and for the rest of the types of labeling, it has remained the same as it is for $k = 1$. This is most likely due to the structure of the WS model when $m = 2$ as already explained in detail.

To sum up, looking at the findings from the graph models we can see some common patterns, as well as variations. The first and foremost observation made is that the number of unique nodes is larger across all the graph models, and values of average degree and $m$, after applying labeling on the nodes of the graphs, in comparison to prior labeling. Such a result was to be expected.

Another characteristic that all of the results share is that when one wishes to label the nodes to apply the $d$-$k$-anonymity algorithm, is that the more the nodes with different color are, the more unique nodes tend to be yielded. To elaborate, for instance, in the case of centrality-based labeling, Closeness, Betweenness, and Degree centrality 1% produce more unique nodes than 5%, which produces more unique nodes than 10%. However, that does not seem to be the case when $m = 2$ in the WS model, where due to the ring structure and average degree of 2, the results show that changing the color in more nodes might result in slightly fewer unique nodes.

Last but not least, an intriguing question that arises from our findings is how fast the number of unique nodes grows when one raises the density of the graph. It is crucial to note the significant decline of unique nodes when we raised the total number of nodes in the graph from 1,000 to 10,000. It is clear that anonymity is increased and the cause behind it might lie in the density, as we kept the average degree the same, and only increased the number of nodes. Thus, the figures for the graph models of 10,000 nodes represent more sparse graphs than the ones of 1,000 nodes.

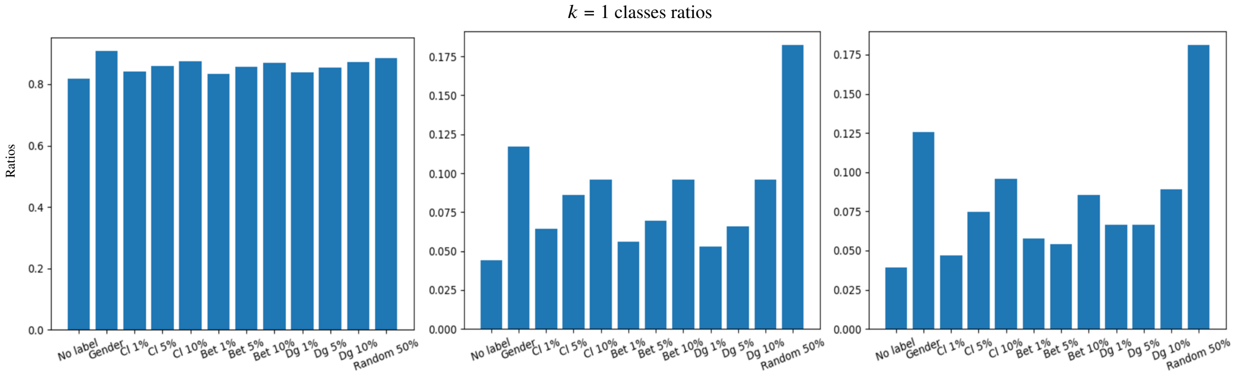
### 6.1.3 Real-world networks

After having discussed the results from the graph models, we now move on to the analysis of the results of the real-world networks. Here the kind of plot that we will use throughout this section is a bar plot for each network, where the horizontal axis denotes the type of labeling and the vertical axis denotes the ratio of unique over the number of nodes of the graph. The y-axis used is linear instead of logarithmic since the results are easily distinguishable.

We note that because of the observation made in the graph models that the Random 50% split tends to do a sharp jump, raising the number of unique nodes by a large portion, the addition of a Random 50% split is incorporated to conduct experiments in the real-world networks as well.

#### 6.1.3.1 COPNET

The first dataset of real-world networks we will discuss is the COPNET networks. We present the results in Figure 24.



((a)) FB network ((b)) SMS network ((c)) Calls network

Figure 24: The bars here indicate the fraction of $k = 1$ unique nodes over the number of nodes on the graph.

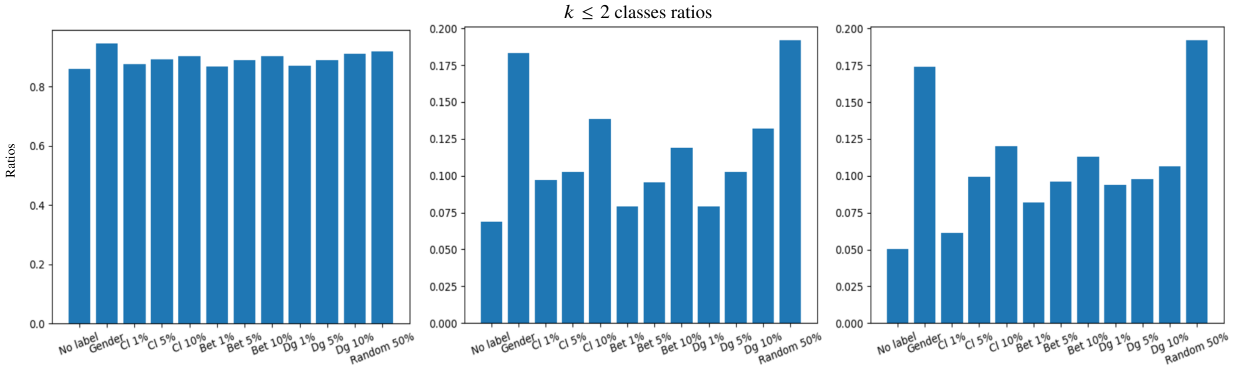
Figure 24 depicts the results for all the networks in the COPNET dataset. We can discern similarities, differences, and common patterns in all of them.

To begin with, the number of unique nodes after having applied labeling on the nodes is larger than the one before labeling, which is to be expected and is in accordance with the results in the graph models. Moreover, another characteristic that both the graph models and the real-world networks share is that the more nodes we have with different colors, the more unique nodes are expected to be produced by the $d$-$k$-anonymity algorithm.

To elaborate, we can observe that in Figure 24(b) and Figure 24(c) the proportion of $k = 1$ nodes over the total number of nodes of the graph, without labels, is slightly less than 0.05. Gender labeling, which is a variant of Random labeling produces more unique nodes than centrality-based ones. The cause behind it is in accordance with our findings so far, since the Gender labeling splits the nodes of the SMS and calls networks into 21% – 79% and 23% – 77% respectively. Meaning, we have a larger portion of nodes with different color than the centrality-based labelings. What is more, a similar pattern can be observed regarding the centrality-based labelings since the 1% produces fewer unique nodes than 5% which produces fewer than 10%. The only difference worth noting is that in the Calls network, Betweenness centrality 5% produces barely fewer unique nodes than 10%, but that might lie in the selection of nodes colored, just like it was the case with the WS model.

Last but not least, we should mention that the Random 50% seems to produce approximately 4 times the amount of unique nodes when compared to without labels and around two-thirds the amount compared to Gender labeling.

What is particularly interesting to note is the huge difference between the findings of the FB networks and the other two networks. More specifically, we can see that over 80% of the nodes are unique even without applying labels on the nodes of the graph, which is vastly different from the other two networks. The largest number of unique nodes produced can be found in Gender labeling and the lowest in the 1% centrality-based labelings. The variations observed though are marginal. However, our intuition is that since most of the nodes are already unique without labels, we can expect such findings. Now we present the results for the $k \leq 2$ classes in Figure 25.



((a)) FB network ((b)) SMS network ((c)) Calls network

Figure 25: The three real-world networks from the COPNET dataset showing the ratio of $k \leq 2$ classes over the number of nodes of the graphs

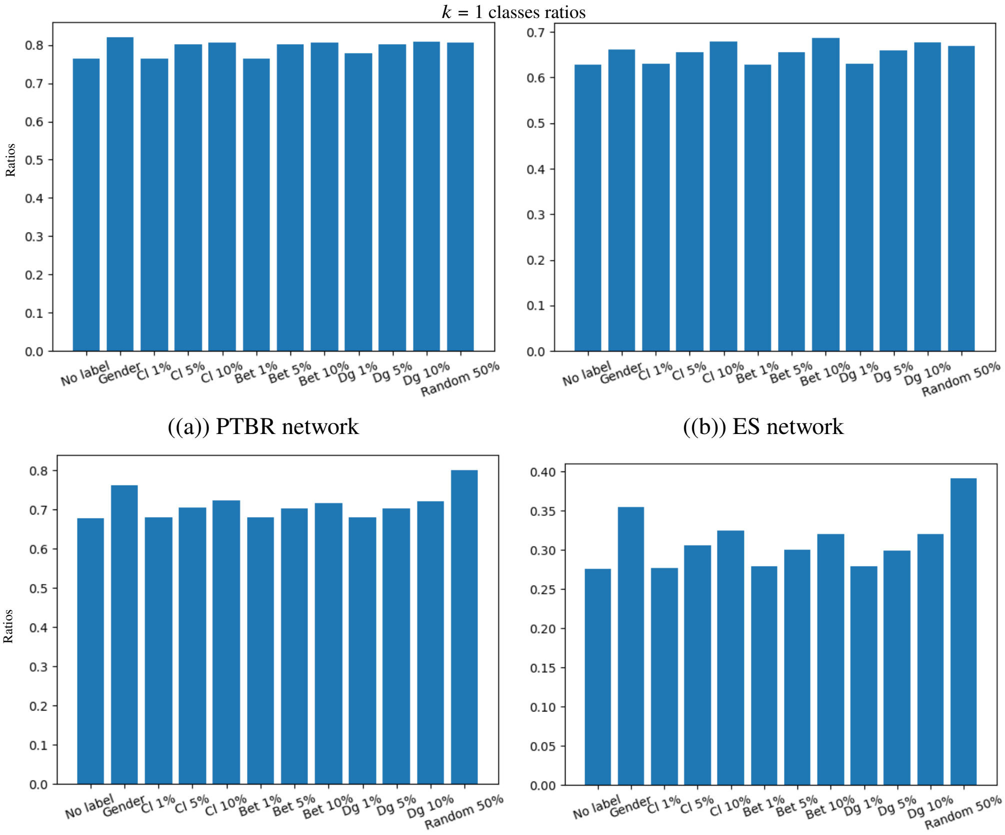
The graphs depict that in the case of FB, we have a small increase of nodes that is not significant. As we described the amount of unique nodes is over 80% in all cases, and the increase here is a minor one.

On the other hand, particularly intriguing observations can be found in SMS and calls networks. To further explain, if we look at the Gender labeling we can see that there is an increase of around 0.05 in the ratio of unique nodes. Moreover, the ratio in all cases (with and without label) is increased by a margin of 0.05 in the SMS network, whereas in calls it has a relatively minor increase from the $k = 1$ case.

Lastly, it should be noted that the number of unique nodes produced by Betweenness centrality 5% is now less than the ones produced by Betweenness 1%, meaning that in summation we end up having more unique nodes if we have more nodes with different color.

#### 6.1.3.2 Twitch

Here we present the results of the four languages spoken in Twitch. First, we give the results for the $k = 1$ classes in Figure 26.



((a)) PTBR network

((b)) ES network

((c)) DE network

((d)) ENGB network

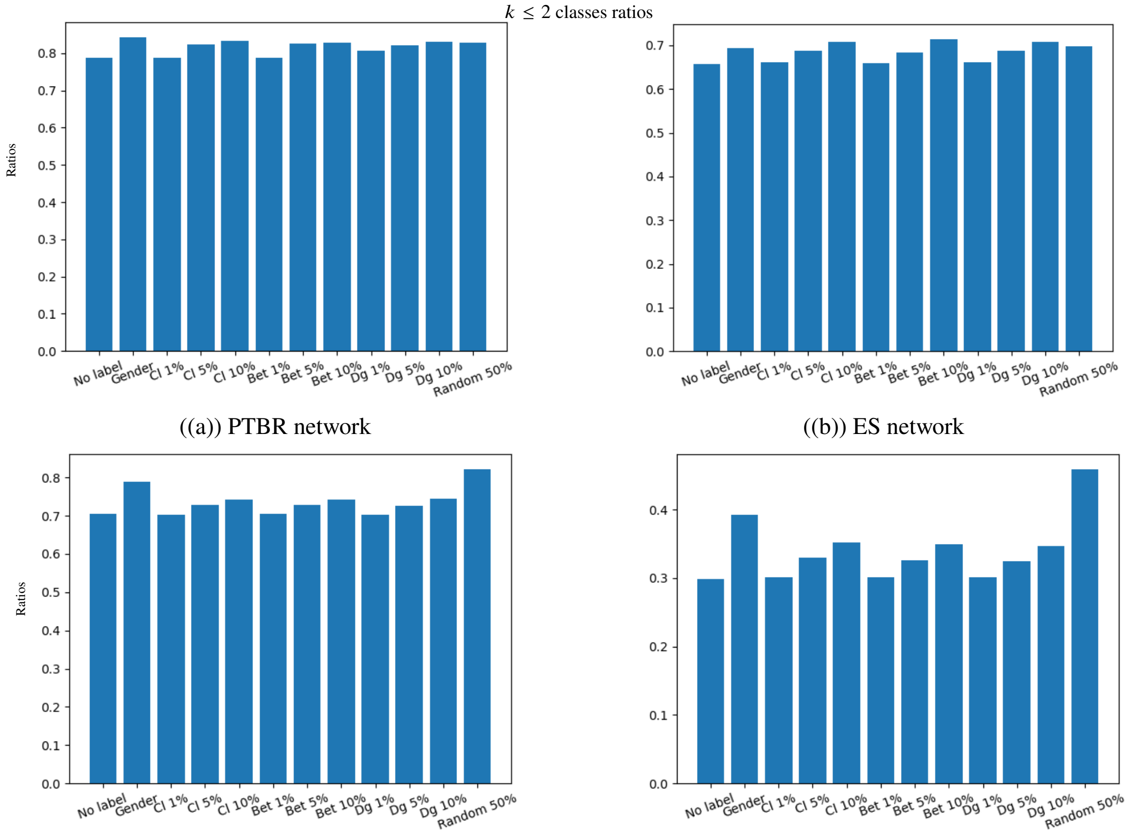
Figure 26: The real-world networks from the Twitch musae dataset showing the ratio of $k = 1$ classes over the number of nodes of the graphs

The graph depicts that over 60% of the nodes are unique. The crest in each bar plot is reached by a different type of labeling. In PTBR we have the maximum value for Gender, in ES the Betweenness 10%, and in DE it is the Random 50%. However, it can be easily discerned that the values for each type of labeling are very close to each other, with the highest being a 10% margin in the DE network, where the percentage of unique nodes is slightly less than 70%.

Regarding the ENGB network we can see that the proportion of unique nodes ranges between around 0.27 and 0.4, with 0.27 being the nodes produced without labeling and 0.4 being the nodes produced by Random 50%. The amount of unique nodes in this case is lower by an approximately two-thirds margin.

What is a shared characteristic across all the Figures though is again that the number of $k = 1$ nodes is the lowest in the case of without labels and that the pattern of more nodes with the same color results in fewer unique nodes persists also here, except the case of PTBR and ES. We now present the results for the $k \leq 2$ classes.

We can see that there is a marginal increase of unique nodes compared to $k = 1$, which does not yield any significant difference in the results. Thus, there is no need for further analysis and investigation since the results are almost the same as in Figure 26.



((a)) PTBR network

((b)) ES network

((c)) DE network

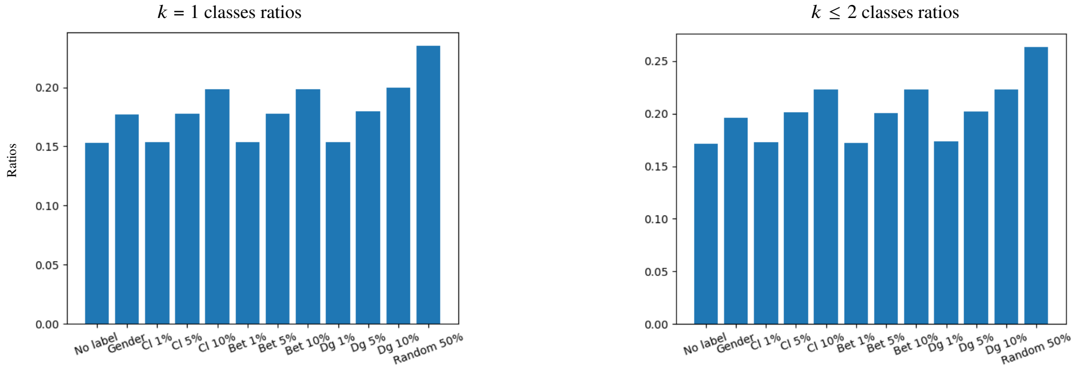
((d)) ENGB network

Figure 27: The real-world networks from the Twitch musae dataset showing the ratio of $k \leq 2$ classes over the number of nodes of the graphs

#### 6.1.3.3 Deezer

The last dataset we experimented with is the Deezer network. We present the results for both the $k = 1$ and $k \leq 2$ classes in Figure 28. A first observation is that in Figure 28(a) is that the ratio of the nodes that belong in $k = 1$ classes, falls in the interval from 0.15 to approximately 0.23. The lowest value is again in the case of without labels and the peak is at the Random 50% labeling. Moreover, the pattern of more nodes with the same colors producing fewer unique nodes is also apparent here. What is more, the increase in the case of $k \leq 2$ is a relatively small one, being around 0.05, which is not a significant raise.

What is particularly interesting to note is that the worst-case scenario corresponds to around 23% o unique nodes. The cause behind it might be that we have a network of 28,281 nodes with a density value of 0.0002, which means that the graph is a lot more sparse than the rest of the networks.



((a))

((b))

Figure 28: The ratio of $k = 1$ classes on the left side and the ratio of $k \leq 2$ ratios on the right side for the Deezer network

To sum up, the main assumptions from our experiments on the graph models seem to be validated by our experiments on the real-world networks. The number of unique nodes produced after having applied node labeling is larger than the one prior to labeling. Consequently, when we apply node labeling, the number of unique nodes is proportional to the number of nodes that have different colors, meaning that the more nodes we have with different colors when we add labels to the nodes, the more unique nodes we expect to get. Last but not least, It is observed both in graph models and real-world networks that a more dense graph tends to have more unique nodes than a more sparse graph.

# 7 Conclusion

In this thesis, we went through the existing literature on anonymity measures and specifically focused on $d$-$k$-anonymity. After discussing the algorithm and its properties we introduced an extension of the algorithm, which can be applied on labeled graphs.

Our algorithm is implemented in C++, and it is built to be applied on labeled undirected graphs. We conducted experiments for the case of $d = 1$ on graph models of 1,000 and 10,000 nodes, and real-world networks and compared the results of unlabeled and binary-labeled graphs.

First and foremost, our findings showed that anonymity decreases when one labels the nodes of the graph, as opposed to unlabeled nodes, in both the graph models and the real-world networks. The only case where the number of unique nodes without labeling and with labeling is the same as when all the nodes of the graph are unique.

Furthermore, our results show that in both the graph models and the real-world networks, when the graph is very dense the majority of the nodes are unique even before labeling. Due to this, the number of unique nodes with any type of labeling is very close to the one without labeling.

Another interesting observation from our results is that the more nodes we have with the same label assigned to them, the fewer unique nodes we have after we apply the algorithm. This is to be expected since when we are working with subgraphs of a main graph and we wish to see if they are colored-isomorphic, the labels between the nodes need to match. Thus, the chances for this to happen are higher when more nodes have the same assigned label in the original graph.

It is also observed that in the case of graph models and real-world networks with approximately the same properties (e.g. number of nodes, number of edges, density), anonymity can differ a lot from one graph to another. In the case of the graph models, we saw that the Barabasi-Albert model produces approximately an order of magnitude more unique nodes than the Erdős–Rényi model and that the Watts-Strogatz model produces the fewest of all. The same pattern is observed in the networks of the COPNET dataset and the networks of the Twitch dataset.

Last but not least, our findings show that there is a significant increase in unique nodes approximately 5 to 10% when $k \leq 2$ compared to only when $k = 1$ in the case of the graph models, whereas in the real-world networks it is a marginal one that does not affect the results significantly.

Some interesting directions this work can be extended to are, first, to extend the implementation to directed graphs, and second, to experiment with more types of labeling than binary node labeling.

# References

[1] S. P. Christensen, *Social media use and its impact on relationships and emotions*. [Theses and Dissertations, 6927], Brigham Young University, 2018.

[2] M. van de Water, “CBS: People with Dutch parents and high incomes live most in their own bubble.” https://www.volkskrant.nl/, 2024. [Online; accessed 23-February-2024].

[3] E. Bokányi, E. M. Heemskerk, and F. W. Takes, “The anatomy of a population-scale social network,” *Scientific Reports*, vol. 13, no. 1, p. 9209, 2023.

[4] L. Willenborg and T. De Waal, *Elements of statistical disclosure control*, vol. 155. Springer Science & Business Media, 2012.

[5] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Nordholt, K. Spicer, and P. de Wolf, *Statistical Disclosure Control*. vol. 2, Wiley New York, 2012.

[6] K. Liu and E. Terzi, “Towards identity anonymization on graphs,” in *Proceedings of the 2008 Association for Computing Machinery SIGMOD international conference on Management of data*, pp. 93–106, 2008.

[7] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, “Resisting structural re-identification in anonymized social networks,” *Proceedings of the 2008 Very Large Database Endowment*, vol. 1, no. 1, pp. 102–114, 2008.

[8] L. Zou, L. Chen, and M. T. Özsu, “K-automorphism: A general framework for privacy preserving network publication,” *Proceedings of the 2009 Very Large Database Endowment*, vol. 2, no. 1, pp. 946–957, 2009.

[9] J. Cheng, A. W.-c. Fu, and J. Liu, “K-isomorphism: privacy preserving network publication against structural attacks,” in *Proceedings of the 2010 Association for Computing Machinery SIGMOD International Conference on Management of data*, pp. 459–470, 2010.

[10] B. Zhou and J. Pei, “Preserving privacy in social networks against neighborhood attacks,” in *2008 Institute of Electrical and Electronics Engineers 24th International Conference on Data Engineering*, pp. 506–515, Institute of Electrical and Electronics Engineers, 2008.

[11] L. Zhang and W. Zhang, “Edge anonymity in social network graphs,” in *Proceedings of the 2009 International Conference on Computational Science and Engineering*, vol. 4, pp. 1–8, Institute of Electrical and Electronics Engineers, 2009.

[12] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, “K-symmetry model for identity anonymization in social networks,” in *Proceedings of the 2010 International Conference on Extending Database Technology*, pp. 111–122, 2010.

[13] G. Minello, L. Rossi, and A. Torsello, “k-anonymity on graphs using the szemerédi regularity lemma,” *Institute of Electrical and Electronics Engineers Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1283–1292, 2020.

[14] R. G. de Jong, M. P. van der Loo, and F. W. Takes, “Algorithms for efficiently computing structural anonymity in complex networks,” *Association for Computing Machinery Journal of Experimental Algorithmics*, vol. 28, pp. 1–22, 2023.

[15] R. G. de Jong, M. P. van der Loo, and F. W. Takes, “The effect of distant connections on node anonymity in complex networks,” *Scientific Reports*, vol. 14, no. 1, p. 1156, 2024.

[16] M. van der Loo, *Topological anonymity in networks*. Statistics Netherlands, 2022.

[17] P. Erdös, “On the evolution of random graphs,” *Publ Math Inst Hungarian Acad Sci*, vol. 5, p. 17, 1960.

[18] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[19] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[20] S. Ji, P. Mittal, and R. Beyah, “Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey,” *Institute of Electrical and Electronics Engineers Communications Surveys Tutorials*, vol. PP, pp. 1–1, 12 2016.

[21] S. Sharma, P. Gupta, and V. Bhatnagar, “Anonymisation in social network: a literature survey and classification,” *International Journal of Social Network Mining*, vol. 1, pp. 51 – 66, 01 2012.

[22] M. Yuan, L. Chen, S. Y. Philip, and T. Yu, “Protecting sensitive labels in social network data anonymization,” *Institute of Electrical and Electronics Engineers Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633–647, 2011.

[23] S. Das, Ö. Eğecioğlu, and A. El Abbadi, “Anonymizing weighted social network graphs,” in *2010 Institute of Electrical and Electronics Engineers 26th International Conference on Data Engineering*, pp. 904–907, Institute of Electrical and Electronics Engineers, 2010.

[24] S. Horawalavithana, J. G. A. Flores, J. Skvoretz, and A. Iamnitchi, “Behind the mask: Understanding the structural forces that make social graphs vulnerable to deanonymization,” *Institute of Electrical and Electronics Engineers, Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1343–1356, 2019.

[25] S. V. Vadamalai, *Lost in the Crowd: Are Large Social Graphs Inherently Indistinguishable?* University of South Florida, 2017.

[26] D. Romanini, S. Lehmann, and M. Kivelä, “Privacy and uniqueness of neighborhoods in social networks,” *Scientific reports*, vol. 11, no. 1, p. 20104, 2021.

[27] “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks and Integrated Services Digital Network Systems*, vol. 30, no. 1, pp. 107–117, 1998. Proceedings of the 1998 International World Wide Web Conference.

[28] J. Golbeck, *Analyzing the Social Web*. Elsevier Inc., Newnes, 01 2013.

[29] G. Csardi and T. Nepusz, “The igraph software,” *Complex systems*, vol. 1695, pp. 1–9, 2006.

[30] G. Varoquaux, K. J. Millman, *et al.*, *Proceedings of the 2010 Python in Science Conference*. Lulu.com, 2010.

[31] B. D. McKay and A. Piperno, “Practical graph isomorphism, II,” *Computing Research Repository*, vol. abs/1301.1493, 2013.

[32] “Practical graph isomorphism,” *Journal of Symbolic Computation*, vol. 60, pp. 94–112, 2014.

[33] P. Sapiezynski, A. Stopczynski, D. D. Lassen, and S. L. Jørgensen, “The Copenhagen Networks Study interaction data,” 11 2019.

[34] B. Rozemberczki, C. Allen, and R. Sarkar, “Multi-scale attributed node embedding,” 2019.

[35] B. Rozemberczki and R. Sarkar, “Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models,” in *Proceedings of the 2020 Association for Computing Machinery International Conference on Information and Knowledge Management*, p. 1325–1334, Association for Computing Machinery, 2020.

[36] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx,” tech. rep., Los Alamos National Lab.(LANL), Los Alamos, New Mexico (United States), 2008.

# 8 Appendices

## 8.1 Nauty coloring

Although Nauty software provides a useful tool to perform isomorphism checks on labeled graphs (colored nodes), there is a limitation. The software is not able to recognize whether the colors of the nodes match to each other. It only recognizes whether two nodes belong in the same partition. Due to the nature of our experiments, we need to know if the nodes have the same color because in real-world networks they can represent the gender of the individual. Thus, it is important that we are aware of this information when we split the nodes into equivalence classes with the $d - k-$algorithm. That is according to the documentation, we need to use two vectors **lab** and **ptn**, the first representing the nodes and the second representing the partitions. Two partitions are split with the number 0 in a cell of the vector **ptn**. A small example taken from the documentation is given below to illustrate this.

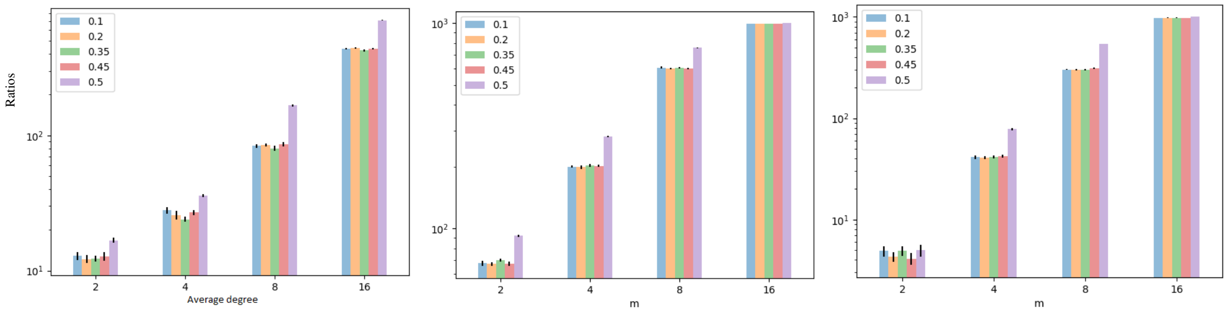
**Example:** We are given a graph of 9 nodes, each one represented by a number from 0 to 8. Now we split the nodes into partitions as follows :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **lab** = | 2 | 3 | 5 | 6 | 1 | 0 | 4 | 7 | 8 |
| **ptn** = | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

The result of this choice of vectors is the following partition: $[\{2\}, \{3\}, \{0, 1, 5, 6\}, \{4, 7, 8\}]$.

The same limitation is present in the Python library [36] where the function ***is_isomorphic*** does not take into account if the colors assigned to the nodes are the same. It only takes into account if the nodes belong in the same partition. So in order to deal with this limitation we had to add an if condition to check whether the nodes of the two graphs have matching colors

## 8.2 Statistical analysis for random labeling



((a)) ER graph model ((b)) BA graph model ((c)) WS graph model

Figure 29: Bar plot of Average-Standard Error of the Mean (SEM) for the Random labeling on graph models with 1,000 nodes. The colored bars show the average ratio of $k = 1$ nodes over 10 experiments for each type of labeling, and the black bar in each colored bar indicates the SEM

| p | 0.1 | | | 0.2 | | | 0.35 | | | 0.45 | | | 0.5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| m | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max |
| 2 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.02 |
| 4 | 0.02 | 0.03 | 0.03 | 0.02 | 0.03 | 0.04 | 0.02 | 0.02 | 0.03 | 0.02 | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 |
| 8 | 0.07 | 0.08 | 0.10 | 0.08 | 0.09 | 0.09 | 0.07 | 0.08 | 0.10 | 0.07 | 0.09 | 0.10 | 0.15 | 0.17 | 0.18 |
| 16 | 0.40 | 0.44 | 0.47 | 0.40 | 0.44 | 0.47 | 0.40 | 0.42 | 0.46 | 0.40 | 0.43 | 0.46 | 0.69 | 0.71 | 0.73 |

Table 4: Table that shows the minimum, max and average values for all the experiments of Random labeling for each value of average degree in the ER graph model with 2 decimals precision

| p | 0.1 | | | 0.2 | | | 0.35 | | | 0.45 | | | 0.5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| m | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max |
| 2 | 0.06 | 0.07 | 0.08 | 0.06 | 0.07 | 0.07 | 0.07 | 0.07 | 0.08 | 0.06 | 0.07 | 0.07 | 0.09 | 0.09 | 0.10 |
| 4 | 0.18 | 0.20 | 0.21 | 0.17 | 0.20 | 0.22 | 0.19 | 0.20 | 0.22 | 0.18 | 0.20 | 0.22 | 0.27 | 0.28 | 0.29 |
| 8 | 0.58 | 0.61 | 0.65 | 0.57 | 0.60 | 0.61 | 0.58 | 0.61 | 0.64 | 0.58 | 0.60 | 0.62 | 0.75 | 0.76 | 0.77 |
| 16 | 0.99 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table 5: Table that shows the minimum, max and average values for all the experiments of Random labeling for each value of average degree in the BA graph model with 2 decimals precision

| p | 0.1 | | | 0.2 | | | 0.35 | | | 0.45 | | | 0.5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| m | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max | Min | Avrg | Max |
| 2 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 |
| 4 | 0.03 | 0.04 | 0.05 | 0.04 | 0.04 | 0.05 | 0.03 | 0.04 | 0.05 | 0.04 | 0.04 | 0.06 | 0.07 | 0.08 | 0.09 |
| 8 | 0.29 | 0.30 | 0.31 | 0.28 | 0.30 | 0.33 | 0.27 | 0.30 | 0.33 | 0.30 | 0.31 | 0.34 | 0.52 | 0.54 | 0.56 |
| 16 | 0.97 | 0.98 | 0.98 | 0.96 | 0.98 | 0.99 | 0.97 | 0.98 | 0.99 | 0.97 | 0.98 | 0.99 | 0.99 | 1.00 | 1.00 |

Table 6: Table that shows the minimum, max and average values for all the experiments of Random labeling for each value of average degree in the WS graph model with 2 decimals precision

## 8.3 Supplementary plots

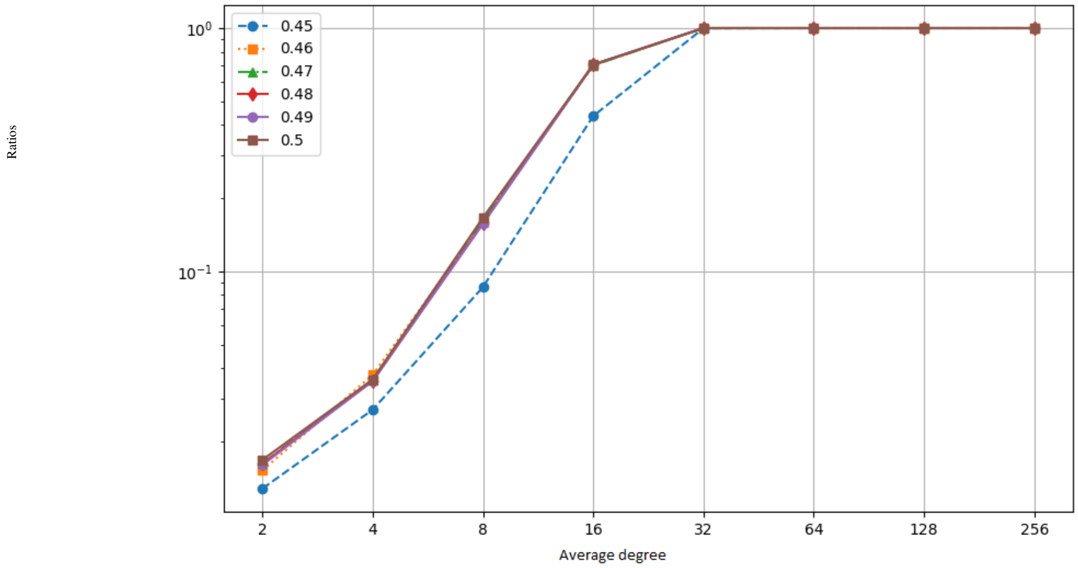


Figure 30: The ratio of $k = 1$ classes for random labeling over 1,000 nodes for the ER graph model

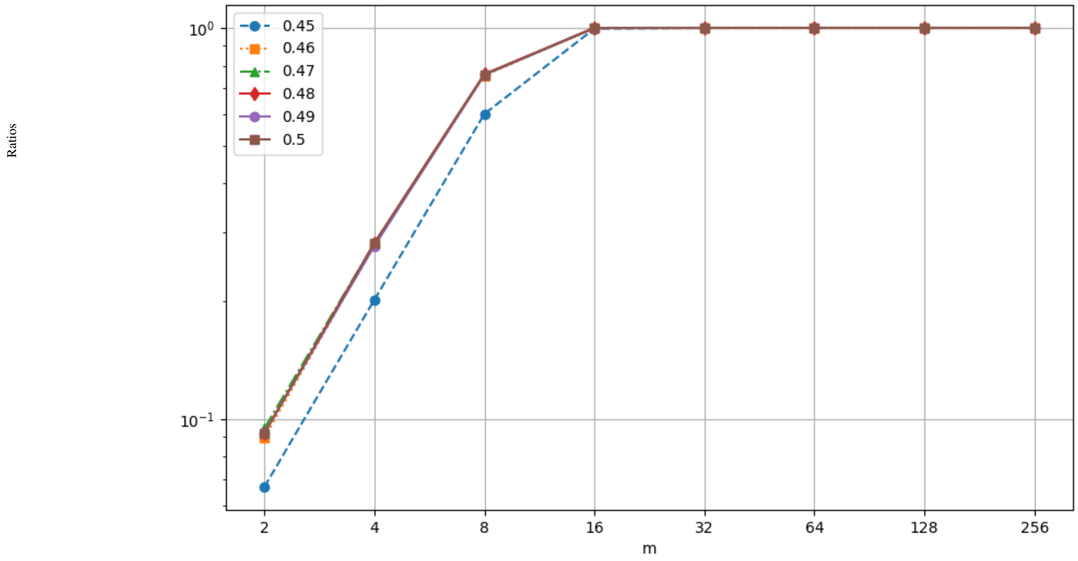


Figure 31: The ratio of $k = 1$ classes for random labeling over 1,000 nodes for the BA graph model

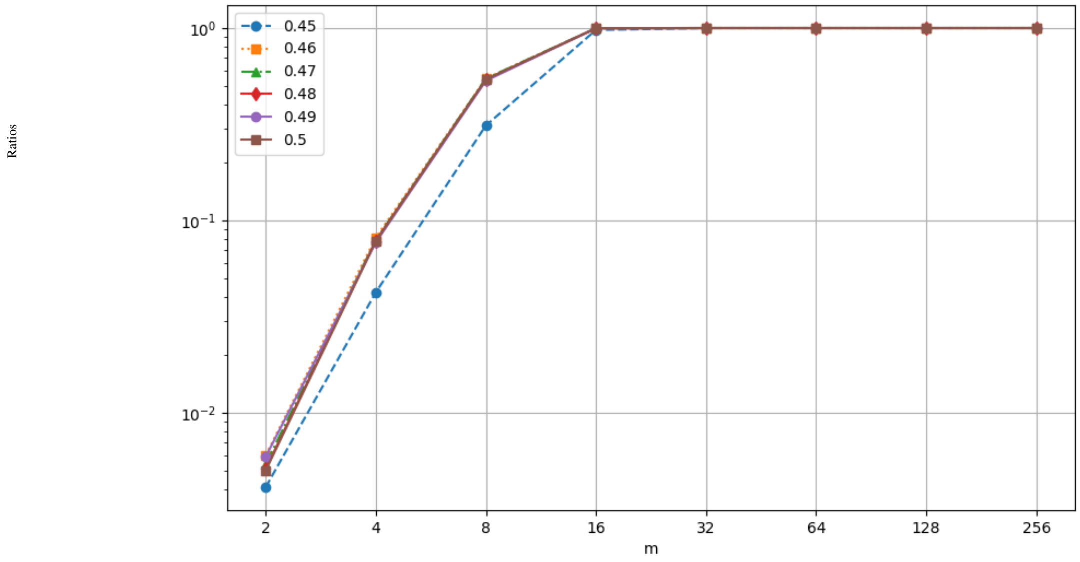


Figure 32: The ratio of $k = 1$ classes for random labeling over 1,000 nodes for the WS graph model