Paper

A Novel Graph-modification Technique for User Privacy-preserving on Social Networks

Seyedeh Hamideh Erfani and Reza Mortazavi

School of Engineering, Damghan University, Damghan, Iran

https://doi.org/10.26636/jtit.2019.134319

Abstract—The growing popularity of social networks and the increasing need for publishing related data mean that protection of privacy becomes an important and challenging problem in social networks. This paper describes the (k,l)-anonymity model used for social network graph anonymization. The method is based on edge addition and is utility-aware, i.e. it is designed to generate a graph that is similar to the original one. Different strategies are evaluated to this end and the results are compared based on common utility metrics. The outputs confirm that the naïve idea of adding some random or even minimum number of possible edges does not always produce useful anonymized social network graphs, thus creating some interesting alternatives for graph anonymization techniques.

Keywords—graph-modification, social networks, privacy-preserving publication of data, graph anonymization, database security.

1. Introduction

In recent years, the extensive use of social networks, such as Facebook, Twitter or MySpace, in family or friendship communications, as well as in political, social, economic, educational, cultural and religious activities, has led many researchers to focus on various aspects of this highly utilized social communication tool (social network). Online social networks have provided many data sharing platforms. Due to the quickly increasing popularity of social network sites on the Web, disclosure of user identity becomes an important problem. Since the risk has a devastating impact on the daily life of the people involved (for example, disclosure of sensitive data, such as e-mails, instant messages, or private relationships), protecting user privacy has become an important and challenging problem in online social networks. Anonymization of the social network structure [1], [2] is a common approach to protecting user privacy.

A social network may be modeled as a graph, where vertices represent individuals, organizations or users, and edges represent connections, relationships between users or information flows [3]. Many researchers have focused on studying the problem of user privacy in online social networks [4]–[8]. To improve the security of social networks, various social network data anonymization techniques have been proposed [9]–[15]. These anonymization approaches are designed based on the principle of k-anonymity.

Anonymization methods are used in undirected social network graphs. Furthermore, these approaches anonymize a social network by inserting and/or deleting edges and vertices in the graph. Since the social network graph structure has been changed by anonymization methods, the utility-related value of social network decreases. For example, if the relationship between two users A and B is removed, user A cannot retrieve sensitive data of user B, and user B will not be able to share sensitive data with user A. If user A is removed, their existence is ignored in the social network. Therefore, developing anonymization methods to protect user privacy in social networks without experiencing the loss of information continues to be an outstanding problem that is still difficult to solve [16].

Ideally, an anonymized social network should protect the privacy of individuals with a minimum loss in its utility-related value, ensuring that an analysis based on anonymized social network data is very similar to that based on its original counterpart [17]. Therefore, the problem of maintaining the utility of data is very important in the process of anonymization of social network data. By referring high levels of utility of data, we mean strict preservation of the pure information bits carried by the original social network data. The information that is distorted and differs, significantly, from the original will probably provide unreliable analysis results. However, most of the existing anonymization techniques fail to generate anonymized social network data with a high degree of utility [17]. Thus, it is important to understand and model the utility of social network data being published by relying on utility-aware metrics [18].

Social networks are usually anonymized based on some computational privacy models. This paper applies the (k,l)-anonymity model which was initiallty defined by Feder *et al*. in 2008 [19]. The authors suggested to use edge addition to implement the model. However, this initial definition suffers from some practical issues that were later addressed in the improved version created by Stokes and Torra [20]. In this definition, a graph is called (k,l)-anonymous if for

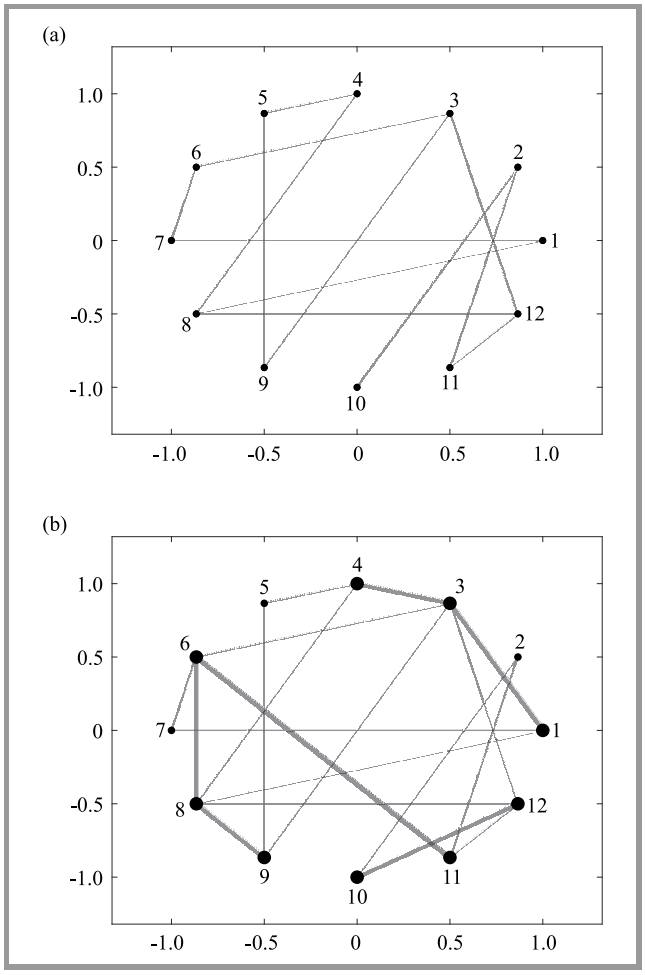

Fig. 1. The original graph (a) and its anonymized version based on edge addition with highlighted new edges (b).

every vertex v in the graph there exist at least $k-1$ other vertices that share at least l of their neighbors with v. The authors suggest to introduce some fake nodes or even to remove all risky nodes to produce an anonymous graph. Although the modification procedures are easy to realize, the usefulness of these suggestions is not evaluated by using utility measures commonly referred to in the graph anonymization literature. Unfortunately, there is the risk that node removal deletes all nodes of the graph, since removing some risky nodes (along with their connected edges) introduces new risky nodes. Additionally, introducing new nodes does not satisfy the privacy requirement in practice, since they do not correspond to distinct individuals [20]. Usually, the amount of changes made to the social graph is used to control the utility distortion in structural-based anonymization [21]–[24]. However, this metric does not consider the impacts on the social links' structure, which has a serious impact on the graph properties [25]. For instance, consider the graph shown in Fig. 1a. If an attacker knows exactly two neighbors of a victim node in the graph, his chance to identify the victim distinctly in the graph is, on average, more than 90% in average. This means that in order to remedy the risk based on naive node deletion, the algorithm has to remove almost all nodes in the original graph. However, adding some new edges, as shown in the Fig. 1b, deprives the attacker of this possibility.

In this paper, a mathematical model is applied to enforce the privacy requirement of (k,l)-anonymity to minimize the cost function related to the distortion of utility of the anonymization process. The aim of this study is to propose a general mathematical model to realize (k,l)-anonymity in social network graphs. To the best of our knowledge, this is the first general procedure that addresses the (k,l)-anonymity problem in graphs.

In summary, we offer the following main contributions:

1. We develop a general mathematical programming model to produce a (k,l)-anonymous graph for $l \geq 1$ and $k \geq 2$.

2. We linearize the anonymization constraints in our mathematical programming approach, which makes it solvable by mixed integer programming (MIP) solvers.

3. We show that the naïve idea of adding some random or even the minimum number of possible edges does not always produce useful anonymized social network graphs.

4. We compare the performance of the proposed approach with the existing approach. The results show that our approach changes slightly the key characteristics of the graph and produces more useful graphs.

The remainder of this paper is organized as follows. Section 2 reviews some related works. Section 3 defines the preliminary concepts that our work is based upon. The pro-posed method is described in Section 4 and is then evaluated in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Here, a number of graph-modification techniques are reviewed as graph anonymization methods. To preserve the privacy of users in the process of publishing data through anonymization methods, the process can be fully random or may be subject to some constraints, meaning that these methods are called random perturbation methods and constrained perturbation methods, respectively.

Random perturbation methods are based on introducing random noise to the original data. In the case of graphs, two main approaches exist: (a) rand add/del that randomly adds and deletes edges from the original graph, and (b) rand switch that exchanges edges between pairs of nodes. Random perturbation techniques are generally the simplest form of graph modification techniques and are less complex. Thus, they are able to deal with large networks, although

they do not offer privacy guarantees, but a probabilistic re-identification model [26] only. For example, Hay *et al*. [27] proposed a method, known as random perturbation, to anonymize unlabeled graphs using the rand add/del strategy that randomly removes p edges and then randomly adds p fake edges. Stokes and Torra [28] professed that a suitable selection of the eigenvalues in the spectral method can perturbate the graph while keeping its most significative edges. The authors in [29] presented a strategy which aimed to preserve the most important edges in the network, trying to maximize data utility while achieving the desired privacy level.

Several methods have been presented in the category of constrained perturbation methods, such as k-anonymity and extended k-anonymity. These methods provide privacy guarantees, but the level of privacy they offer may strongly depend on the knowledge of the adversary. The most basic knowledge of adversary is based on vertex degree [26]. The k-anonymity model was introduced in [30] and [31] for privacy preservation of structured or relational data. The k-anonymity model indicates that the attacker cannot distinguish between different k records, although he manages to find a group of quasi-identifiers. Therefore, the attacker can not re-identify an individual with a probability greater than $1/k$ [26]. There are some other models developed attempting to extend the k-anonymity model to overcome some particular disadvantages. Feder *et al*. [19] called a graph (k,l)-anonymous if, for every vertex in the graph, there exist at least k other vertices that share at least l of their neighbors. Severe weaknesses of that work were addressed by Stokes and Torra [20].

3. Preliminaries

In this section the preliminary concepts upon which this work is built will be introduced, including: graph anonymization and utility assessment of an anonymous graph.

3.1. Graph Anonymization

Similarly to other works, we model the social network as a simple graph, in which the degree of a vertex represents the number of its neighbors. Let $G = (V,E)$ be a social network graph, where V denotes the set of vertices and E represents the set of edges. As in [19], we assume, throughout this paper, that the social network graph is undirected, unweighted, and contains no self-loops, because this is an important category of graphs to study. Most of the social networks, such as Facebook, LinkedIn, Orkut and so on, allow only bidirectional links and are thus instances of such simple graphs.

Consider a simple graph and an attacker who knows that a target person and some number of their friends form a community. In the released graph, the attacker could find such a community to narrow down the set of nodes that might correspond to the target individual. The goal of the anonymization method is to prevent the attacker from identifying individuals based on their immediate neighbors. To achieve this, we use the (k,l)-anonymity definition introduced by Feder *et al*. in 2008. In this definition, it is required that for every node in the graph, some subset of its neighbors should be shared by other nodes. In this way, an attacker who knows some subset of the neighbors of the target individual and is even capable of pinpointing them in the graph, will not be able to distinguish the target individual from other nodes in the network that share this subset of neighbors [19]. More formally the (k,l)-anonymity property is defined as follows.

Definition 1 (k,l)-anonymity [19]. A graph $G = (V,E)$ is (k,l)-anonymous if for each vertex $v \in V$, there exists a set of vertices $U \subset V$ not containing v such that $|U| \geq k$ and for each $u \in U$ the vertices u and v share at least l neighbors.

3.2. Utility Assessment of an Anonymous Graph

We now discuss how to measure the utility loss of an anonymized graph based on topological features of the graph. For a given original graph $G(V,E)$, $\hat{G}(V,E)$ is the (k,l)-anonymous version, such that the utility of the anonymized graph should be as close as the utility of the original graph [25]. Zhou and Pei [23], [32] consider the total number of added edges and the number of vertices that are not in the neighborhood of the target vertex and are linked to the anonymized neighborhood for the purpose of anonymization. Another work in [21] controls the utility loss by preferring the largest frequent subgraphs during anonymization to impose less graph modification. The total cost of anonymization is still calculated by the amount of changes made during perturbation [25]. Although, based on this criterion, the structural properties of the social network graph are ignored, the number of changes to minimize the information loss of network graph is still important.

When a social network graph is anonymized, the resulting graph loses some utility compared with the original graph [33]. Some attempts have been made to preserve the structural properties of a social network graph during anonymization. Research done by [18], [34] considers local community structure as the subject of utility preservation. In [34], the authors proposed an approach in which the graph is partitioned by a local structure.

In this paper, we have used the average vertex degree (AVD), the average path length (APL), and the average closeness centrality (ACC) of the anonymized graph in comparison with the original graph, in order to evaluate the proposed method. These three indices, along with the degree distribution, are considered to be standard measures in graph-analysis studies. These criteria are defined as follows (Table 1).

Consider an unweighted and undirected graph G having the set of vertices V that $|V| = n$. Assume $d(v_1, v_2)$, where $v_1, v_2 \in V$ denote the shortest distance between v_1 and v_2. The degree of vertex v is denoted as $\deg(v)$.

Table 1
Notations used in the paper

Notation	Description
n	Number of graph vertices
k,l	Anonymization parameters
$\deg_i$	Degree of v_i
AVD	Average vertex degree
APL	Average path length
ACC	Average closeness centrality
CV	Candidate vertices
MIP	Mixed integer programming
PSPL	Pairwise shortest path length

Average vertex degree. The average vertex degree is defined as the average degree of all network nodes [35]. Then, the average vertex degree AVD_G is:
$AVD_G = \frac{1}{n} \cdot \sum_{v \in V} \deg(v)$.

Average path length. The average path length is defined as the average number of steps along the shortest paths between all pairs of reachable network nodes. The average path length APL_G is defined as:
$APL_G = \frac{1}{n \cdot (n-1)} \cdot \sum_{i \neq j} d(v_i, v_j)$.

Average closeness centrality. The closeness centrality of a node is a measure of centrality in a network, calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. The average closeness centrality in a graph is given by the average closeness centrality of all nodes.

4. The proposed Method

In this section, we describe how to satisfy (k,l)-anonymity in a given graph $G = (V,E)$, dividing the process into two main parts. First, solving the mathematical model that the anonymization problem is defined by, using a mixed integer programming in which the best edges are found to be added to G to produce the anonymous graph $\hat{G}$. Second, post-processing that removes unnecessary edges from $\hat{G}$. In Algorithm 1, the two main parts of the proposed method are shown. The function accepts a simple graph $G(V,E)$, anonymization parameters k and l as inputs and produces an anonymous graph $\hat{G}(V,E')$. The number of vertices is stored in n in line 1. The pairwise shortest path length matrix corresponding to graph G is computed in line 2. In line 3, for each newly added edge e_{ij} that connects v_i to v_j, its associated cost related to its addition is computed. The anonymous graph is produced in lines 4–10. For the case of $l = 1$ (line 4), the optimal solution is obtained by using the MIP mathematical model in line 5 that is introduced in Subsection 4.1. For the case of $l > 1$, before solving MIP problem in line 9 that yields G_2, a set of candidate vertices (CV) is computed using the *candidateVertices* function in line 8, which is shown in Algorithm 2. In line 10, all unnecessary added edges are removed from G_2 using the *postProcess* function (Algorithm 3), then its output is saved in $\hat{G}$, which is returned in line 11.

Algorithm 1: The pseudo-code of the proposed method

```
   Input: G(V,E): original graph, k,l: anonymization
          parameters
   Output: Ĝ(V,E'): anonymized graph
1  n = |V|
2  PSPL ← pairwise shortest path length matrix of
     graph G
3  compute costs(i,j),  ∀{v_i,v_j} ∈ E'
4  if l == 1 then
5  |  Ĝ ← MIPsolver(G,PSPL,k,l,n,costs,deg)
6  else
7  |  if l > 1 then
8  |  |  CV ← candidateVertices(G,PSPL,k,l)
   |  |    // Algorithm 2
9  |  |  G_2 ←
   |  |    MIPsolver(G,PSPL,k,l,n,costs,deg,CV)
10 |  |  Ĝ ← postProcess(G,G_2,k,l)
   |  |    // Algorithm 3
11 return Ĝ
```

4.1. Solving Mathematical Model

This paper uses edge addition as the graph-modification technique to produce the anonymous graph. In order to produce a useful graph, a general mathematical model of the problem is introduced that takes into account the k minimum number of different vertices sharing l of their neighbors. The model uses the following components:

Definition of sets: The indexes of vertices $v \in V$ in the graph are saved in S.

Fixed parameters and constants: Three parameters n,k, and l represent the number of vertices, and k and l of the anonymization model, respectively. Parameter c_{ij} is the cost of adding the edge e_{ij} that connects v_i to v_j. The cost matrix $C = [c_{ij}]$ is symmetric, i.e., $c_{ij} = c_{ji}$. It is equal to 0 for connected vertices in the original graph, and assumes a positive value for not connected ones. Therefore, to reduce the amount of data passed to the solver, only the upper triangular part of C is applied in practice. Additionally, deg_i denotes the degree of v_i in the original graph.

Independent problem variables: The solution consists of connected vertices. The binary decision variable x_{ij} determines the connectivity of v_i and v_j in the produced anonymized graph, where $x_{ij} = 1$, if and only if the related vertices are to be connected. In order to decrease the space complexity of the final model, we only consider x_{ij} for $j > i, i, j \in S$, since the graph is undirected ($x_{ij} = x_{ji}$) and loop free ($x_{ii} = 0$).

Objective function: The objective function is to minimize the aggregate cost of change with respect to the original graph, i.e.:

$$\min_{x_{ij}} \sum_{i,j\in S, i<j} c_{ij}x_{ij}.$$

Constraints: The constraints fall into two categories: original edge preserving constraints, and anonymization constraints.

1. Original edge preserving constraints: none of the existent edges in the original graph cannot be removed, i.e.:

   $$x_{ij} = 1, \quad \forall e_{ij} \in E\ .$$

   In order to speed up the computation, the warm-start strategy is used in which $x_{ij} = 1$ for all of the connected vertices v_i and v_j.

   Anonymization constraints: assume that $N_l(j)$ is the l neighbors of v_j, i.e. $N_l(j) = \{S_j | \#S_j = l, \forall i \in S_j, (v_i, v_j) \in E\}$. These constraints enforce that each l neighbors of a vertex are also the neighbors of at least k different vertices in the anonymized graph:

   $$\sum_{w\in S\setminus S'_j} \left(\prod_{i\in S'_j} x_{iw} \right) \geq k \quad \forall S'_j \in N_l(j), j \in S\ .$$

   This function is non-linear due to the multiplication of x-variables. It is required to replace each nonlinear term $\prod_{i\in S'_j} x_{iw}$ with a new variable $z(i_1, i_2, \ldots, w)$ where $i_n \in S'_j$ and add linearization constraints in the following manner:

   $$\begin{aligned}
   &0 \leq z(i_1, i_2, \ldots, w) \leq x_{i_1 w}\\
   &0 \leq z(i_1, i_2, \ldots, w) \leq x_{i_2 w}\\
   &\vdots\\
   &0 \leq z(i_1, i_2, \ldots, w) \leq x_{i_{|S'_j|} w}\\
   &z(i_1, i_2, \ldots, w) \geq x_{i_1 w} + x_{i_2 w} + \cdots + x_{i_{|S'_j|} w} - (|S'_j| - 1)
   \end{aligned}$$

   In the case of $l = 1$, as we introduced in [36], for each k the constraints can be simplified to the constraints that impose the minimum degree of each vertex to be at least k, i.e.:

   $$\sum_{i<j} x_{ij} + \sum_{j<i} x_{ji} \geq k \quad \forall i \in S, deg_i < k\ .$$

   It is also notable that for a fixed w, $\prod_{i\in S'_j} x_{iw} = \prod_{i\in S''_j} x_{iw}$ where S''_j is defined in the following:

   $$S''_j = S'_j - \{i | i \text{ is connected to } w\}.$$

2. Run-time constraint: according to Definition 1, we call the graph $G = (V, E)$, (k, l)-anonymous, if for each vertex $v_i \in V$, there exist at least l vertices, which are simultaneously connected to v_i and also to at least $k - 1$ other vertices. In other words, if $v_j \in V$ connects to p vertices and also q other vertices connect to those p vertices, so that $|p| \leq l$ and $q < k$, then the graph G is not (k, l)-anonymous. In this situation, the MIP solver for finding the minimum cost (k, l)-anonymous graph needs to connect the vertex v_j to at least $l - p$ number of best vertices, so that all of the l vertices are connected to k other vertices. The MIP solver selects the best vertices (vertices with the minimum cost of adding their corresponding edges to the original graph) from all vertices in the graph. Therefore, for large and dense graphs and in the case of $l > 1$, implementation of the proposed method will take a lot of time. In this paper, in order to solve the problem in a reasonable time, we have decided that for all combination of l vertices of the graph G, we propose a set of vertices to the MIP solver that is called candidate vertices. These CV corresponding to l vertices are obtained from the union of their adjacent vertices. The CV is obtained by using Algorithm 2 (line 8 of Algorithm 1). The *CandidateVertices* function gets the original graph G, the PSPL matrix corresponding to graph G and anonymization parameters k and l. The *CV* is generated as the output of the function. In Algorithm 2, for all combination of l vertices of the graph G, a sorted vector D is produced in line 2 in which PSPL$(v_{i_k}, :)$ is the i_k-th row in the pairwise shortest path length matrix of G. Since the main diagonal of the PSPL matrix is zero, the first l elements in sorted vector D will be zero. In line 3, *validIdx* is the index of vertices obtained from $(l+1)$-th component of D and for threshold number ($Th \geq k$) of it. That Th is a positive number that determines the number of candidate vertices for all combinations of l vertices. *ValidIdx* introduces vertices that are considered for connecting to vertices v_{i_1}, v_{i_2}, ..., v_{i_l}. In line 4, the CV is obtained as a matrix with $l+1$ dimensions that $CV(i_1, i_2, \ldots, i_l, validIdx) = 1$. For example, for $l = 2$ and $Th = k$, D is obtained by Eq. (1) that the CV would be a $n \times n \times n$ matrix that for each 2-combination (v_1, v_2), $v_1, v_2 \in V$, the vertex v is a candidate vertex for connecting to both v_1 and v_2 provided by $\mathrm{CV}(v_1, v_2, v) = 1$, for $v \in validIdx$. Therefore, CV proposes candidate vertices to the MIP solver and the MIP solver finds the best vertices from v to connect them to the vertices $\{v_{i_1}, v_{i_2}, \ldots v_{i_l}\} \in V$.

   $$\begin{aligned}
   &D = sortIndex(\mathrm{PSPL}(v_{i_1}, :) \cdot \mathrm{PSPL}(v_{i_2}, :)), \forall i_1, i_2 \in S.\\
   &validIdx = D[3 : 3 + k].\\
   &\mathrm{CV}(i_1, i_2, validIdx) = 1.
   \end{aligned} \tag{1}$$

In this study, two different variants of the general mathematical model have been applied to the original graph. The difference between these variants consists in the different cost functions that are to be minimized in the objective function. Specifically, the following models are considered:

Algorithm 2: The pseudo-code of the candidateVertices function

Input: $G(V,E)$: original graph, PSPL: pairwise shortest path length matrix, k,l: anonymization parameters
Output: CV candidate vertices

```
1 foreach l-combination of v_i ∈ V, ∀i ∈ S do
2     D ← sortIndex(PSPL(v_{i_1},:) * PSPL(v_{i_2},:) * ··· * PSPL(v_{i_l},:)), ∀i_1, i_2, ..., i_l ∈ S
3     validIdx ← D[l+1 : l+1+Th]
4     CV(i_1, i_2, ..., i_l, validIdx) ← 1
5 return CV
```

- Model 1. This model is an approach to the implementation of the Feder *et al*. [19] anonymization model that tries to minimize the number of added edges. In this model, it is assumed that all edges cause the same level of destruction in the graph, therefore $c_{ij} = 1, \forall i, j \in S$.

- Model 2. It is interesting to add new edges that minimally change APL of the original graph, which is an important property of G. This model tries to add edges that change APL minimally. It is hard to compute the amount of change in APL for a large number of sets of candidate edges, since these edges reinforce the value for other edges. Therefore, the model approximates the total costs of the number of newly added edges by aggregating their individual effects on APL. More precisely, if c_{ij} is the amount of change in the APL of the original graph caused by the addition of e_{ij} to G, the total value of change in the APL for the set of the newly added edges $E' \subset V \times V \setminus E$ is approximated by $\sum_{e_{ij} \in E'} c_{ij}$.

4.2. Post-process

As mentioned before, in the mathematical model presented for finding an anonymous graph, for the anonymization parameter $l = 1$, the MIP solver will achieve the optimal solution (because it checks the addition of all possible edges that are not existent in the original graph and selects the edges with the minimum cost). If the anonymization parameter is $l > 1$, however, in order to solve the mathematical problem in a reasonable time frame, instead of selecting from all possible edges, the MIP solver selects the edges from the set of edges corresponding to the set of candidate vertices. For this reason, the solution may be non-optimal and some of the added edges can be deleted from the anonymous graph. Therefore, the post-process stage is performed for the case of $l > 1$, by Algorithm 3 (line 10 of Algorithm 1). The *postProcess* function gets the original graph G, the anonymous graph G_2 (from *MIPsolver*'s output) and anonymization parameters k and l. The $\hat{G}$ is the improved anonymous graph that is generated as the output of the *postProcess* function. In line 1, the betweenness centrality

Algorithm 3: The pseudo-code of the postProcess function

Input: G: original graph, G_2: anonymized graph, k,l: anonymization parameters
Output: $\hat{G}$: improved anonymous graph

```
1 costs ← betweenness_centrality(G_2)
2 M ← AdjacencytMatrix(G)
3 M_2 ← AdjacencytMatrix(G_2)
4 AddedEdges ← sort((M_2 − M) * (costs + 1))
5 for i = 1 : length(AddedEdges) do
6     if removing AddedEdges(i) from G_2 keeps the graph (k,l)-anonymous then
7         remove AddedEdges(i) from G_2
8 Ĝ ← G_2
9 return Ĝ
```

of all edges in graph G_2 is computed as *costs*. The edge betweenness centrality is defined as the number of the shortest paths that go through an edge in the graph or network [37]. The adjacency matrices corresponding to graphs G and G_2 i.e. M and M_2 are obtained in lines 2 and 3, respectively. In line 4, all of the added edges in the anonymized graph G_2 are sorted in a descending order based on the edge betweenness centrality criterion. Then, in lines 5–7, from the beginning of the sorted *AddedEdges* list, the possibility of deleting the edges is checked and all unnecessary edges are removed from the anonymous graph G_2 in line 7. At the end of Algorithm 3, $\hat{G}$ is set by the improved anonymous graph G_2 and is returned as output in line 9. Each edge in the network can be associated with an edge betweenness centrality value. An edge with a high edge betweenness centrality score represents a bridge-like connector between two parts of the network, with their removal potentially affecting communication between many pairs of nodes, based on the shortest paths between them [38]. Therefore, deleting the edge with the greatest betweenness centrality value will bring the structure of the anonymized graph closer to the original one by increasing APL.

5. Empirical Evaluations

In this section, we conduct some experiments to validate the proposed method. The aim of these experiments is to show the strengths and weaknesses of model 1 and model 2, as an implementation method of the Feder *et al*. model [19] and the proposed method, respectively. In all experiments, a laptop with an Intel Core i7 2 GHz CPU, 16 GB of main memory and the Windows 8 64-bit operating system is used. The models have been solved using CPLEX/GAMS MIP [39] software.

First, the datasets that have been used in the experiments are introduced in Subsection 5.1. Then, the results are shown in Subsection 5.2. In all experiments, the degree of change in AVD, APL, and ACC is used to measure utility, as in-

Table 2
Structural properties of synthetic and real graphs

Dataset	Vertices	Edges	AVD	APL	ACC
SF50	50	96	3.8400	2.8433	0.0073
SF100	100	196	3.9200	2.9048	0.0035
SF200	200	396	3.4449	3.9600	0.0015
RA82	82	94	2.2927	5.7350	0.0022
RA129	129	178	2.7597	4.8307	0.0017
RA176	176	238	2.7045	5.4292	0.0011
karate	34	78	4.5882	2.4082	0.0129
dwt_72	72	75	3.0833	8.2254	0.0018
lesmis	77	254	6.5974	2.6411	0.0051
can_96	96	336	8.0000	4.4105	0.0024
polbooks	105	441	8.4000	3.0788	0.0032
football	115	613	10.6609	2.5082	0.0035

troduced in Subsection 3.2. The weaker the modification of these measures caused by the method concerned, the better the anonymized graph is suited for future investigation.

5.1. Datasets

The proposed method is applied to a number of synthetic and real graph datasets that are available online to see its performance in different topologies. Structural properties of these graphs are shown in Table 2. For each graph, the number of vertices, the number of edges and other structural properties described in Subsection 3.2, such as AVD, APL, and ACC, are reported. The synthetic datasets are summarized as follows:

- **SF** – a scale-free dataset based on the Barabasi's model. This graph is a connected graph, where vertex degrees are drawn from a power-law distribution similar to real-world social networks;
- **RA** – a random network based on the Erdos-Renyi model in which vertices are connected based on probability p.

Additionally, six real datasets are tested in a similar manner to assess our method in different topologies:

- karate – a social network of friendships between 34 members of a karate club at a US university in the 1970s;
- dwt – this collection consists of thirty matrix patterns collected by Gordon Everstine of the David W. Taylor Naval Ship Research and Development Center, Bethesda, MD, USA. These patterns were collected from various US military and NASA users of NASA's structural engineering package NASTRAN, for use as a benchmark collection for variable bandwidth reordering heuristics;
- lesmis – co-appearance weighted network of characters in the novel "Les Miserables";
- can – a graph that has a symmetric pattern made of cans;

Table 3
The amount of AVD error for different values of k and l and for the proposed method and for the Feder *et al.* [19] model applied to synthetic graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
SF50	Proposed method	1	0.4000	1.8400	3.4800	6.2400	Feder *et al.*	1	0.4000	**1.7600**	**3.4400**	6.2400
		2	**6.5200**	**10.9200**	**14.4800**	18.2000		2	6.6800	11.0000	14.5200	**18.1600**
		3	8.0400	14.1200	17.8000	22.1600		3	**6.6800**	**13.1600**	**17.5200**	22.1600
SF100	Proposed method	1	0.5600	2.1600	3.8400	6.5200	Feder *et al.*	1	**0.5000**	**2.1000**	**3.8000**	**6.5000**
		2	10.9000	16.7000	21.2600	27.0200		2	**10.0600**	**16.3800**	**20.9400**	27.0800
		3	14.2800	22.5600	27.7800	34.2800		3	**12.5600**	**21.8600**	**12.4600**	**18.0800**
SF200	Proposed method	1	0.5300	2.0600	3.8300	6.6400	Feder *et al.*	1	**0.4500**	**1.9100**	**3.6600**	**6.4400**
		2	12.2000	**19.2000**	**25.6500**	**32.5400**		2	**11.1600**	19.2700	25.9100	34.2600
		3	4.2000	8.3100	12.3800	18.2000		3	4.2000	8.3100	12.3800	18.2000
RA82	Proposed method	1	1.0244	2.9268	4.8537	7.8537	Feder *et al.*	1	**0.9512**	**2.7317**	**4.7073**	**7.7073**
		2	3.8780	7.7317	**10.9268**	15.0000		2	**4.2683**	**7.3659**	10.9512	**14.8537**
		3	3.9268	**7.3171**	**11.1463**	16.1707		3	**3.5122**	7.3415	11.3171	**15.8780**
RA129	Proposed method	1	0.8217	2.4031	4.3721	7.3643	Feder *et al.*	1	**0.7287**	**2.3256**	**4.2636**	**7.2403**
		2	5.5814	9.7674	**14.2326**	19.9070		2	**5.4109**	9.7674	14.4031	**19.8140**
		3	5.5814	10.0310	14.0620	21.3333		3	**5.1783**	**9.9380**	**13.9690**	**21.1938**
RA176	Proposed method	1	0.8523	2.4318	4.3636	7.3636	Feder *et al.*	1	**0.7841**	**2.3977**	**4.3068**	**7.2955**
		2	**5.7386**	**9.7386**	**13.7614**	19.5341		2	5.8977	9.8750	13.9091	**19.0568**
		3	5.7159	10.2273	**13.7727**	20.5000		3	**5.6591**	**9.7386**	14.2386	**19.9091**

Table 4

The amount of AVD error for different values of k and l and for the proposed method and for the Feder *et al.* [19] model applied to real-world graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
karate	Proposed method	1	0.4706	1.7059	3.2941	5.8824	Feder *et al.*	1	**0.4118**	**1.6471**	3.2941	5.8824
		2	5.1765	9.3529	11.7647	14.5294		2	**4.4118**	**8.8235**	**11.5882**	14.5294
		3	6.5882	**11.0588**	14.0000	**17.5882**		3	**5.2353**	11.4118	**13.1176**	17.6471
dwt_72	Proposed method	1	1.0000	2.9722	4.9722	7.9722	Feder *et al.*	1	**0.9722**	**2.9167**	**4.9167**	**7.9167**
		2	2.7222	6.7222	9.3889	12.8056		2	2.7222	6.7222	**9.3611**	**12.7222**
		3	2.7500	**6.7222**	9.5278	13.1111		3	2.7500	6.7500	**9.47221**	**12.8889**
lesmis	Proposed method	1	0.6234	1.5584	2.7532	4.8312	Feder *et al.*	1	**0.5714**	**1.4805**	**2.4675**	**4.5195**
		2	6.1299	11.2987	14.9351	**19.7922**		2	**5.6883**	**10.6234**	14.9351	19.9740
		3	**4.2857**	**7.8961**	**18.1039**	**24.6753**		3	11.7922	15.3766	19.6883	24.9091
can_96	Proposed method	1	0.0000	0.0000	1.3333	3.3333	Feder *et al.*	1	0.0000	0.0000	**0.6667**	**3.0000**
		2	3.5417	**7.3750**	**10.4375**	**14.9792**		2	**3.4792**	7.8333	10.7708	15.0625
		3	**7.0000**	**10.6250**	**14.5417**	19.1458		3	7.9375	12.1667	15.0000	**18.5417**
polbooks	Proposed method	1	0.0190	0.4000	1.3333	3.3333	Feder *et al.*	1	0.0190	**0.2857**	**1.2000**	**3.2381**
		2	6.7238	11.6381	15.5429	21.5238		2	**5.1048**	**10.0952**	**15.4476**	**21.2571**
		3	3.8667	7.9619	20.4190	**27.2762**		3	3.8667	7.9619	**11.8286**	28.2667
football	Proposed method	1	0.0000	0.0000	0.0000	0.1913	Feder *et al.*	1	0.0000	0.0000	0.0000	**0.1217**
		2	8.2783	14.0522	**18.7652**	**25.4261**		2	**7.8783**	**13.8783**	18.9565	25.6696
		3	4.7826	8.8000	12.5043	17.7043		3	4.7826	8.8000	12.5043	17.7043

Table 5

The amount of APL error for different values of k and l for the proposed method and for the Feder *et al.* [19] model applied to synthetic graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
SF50	Proposed method	1	**0.0275**	**0.1472**	**0.3937**	**0.7244**	Feder *et al.*	1	0.0672	0.4035	0.6476	0.9088
		2	**0.8591**	**1.0615**	**1.1456**	**1.2346**		2	0.8999	1.0672	1.1595	1.2354
		3	**0.8950**	1.1382	1.2223	**1.3170**		3	0.9562	**1.1235**	**1.2191**	1.3170
SF100	Proposed method	1	**0.0185**	**0.0609**	**0.2154**	**0.4588**	Feder *et al.*	1	0.0324	0.2507	0.4322	0.6449
		2	0.8718	1.0522	**1.1225**	**1.1871**		2	**0.8441**	**1.0550**	1.1229	1.1889
		3	**0.9572**	1.1209	1.1910	1.2602		3	0.9742	**1.1188**	**1.0413**	**1.0980**
SF200	Proposed method	1	**0.0042**	**0.1035**	**0.3025**	**0.5887**	Feder *et al.*	1	0.0674	0.3588	0.6084	0.8624
		2	1.1604	**1.4144**	**1.5202**	**1.5894**		2	**1.1516**	1.4265	1.5410	1.6154
		3	1.4687	1.4894	1.5098	1.5391		3	1.4687	1.4894	1.5098	1.5391
RA82	Proposed method	1	**0.2143**	**1.4763**	**2.2697**	**2.9009**	Feder *et al.*	1	1.6521	2.7352	3.1845	3.4299
		2	**2.5621**	**3.1125**	**3.3643**	**3.6067**		2	2.6834	3.1574	3.4176	3.6518
		3	**2.4564**	**3.1098**	**3.4109**	**3.6729**		3	2.4871	3.1833	3.5082	3.7199
RA129	Proposed method	1	**0.1590**	**0.9227**	**1.5043**	**1.9891**	Feder *et al.*	1	0.6486	1.5853	2.0178	2.3767
		2	**2.0746**	**2.4457**	**2.6544**	**2.8300**		2	2.0926	2.4628	2.6765	2.8623
		3	2.0110	**2.4569**	**2.6622**	**2.8580**		3	**1.9873**	2.4640	2.6766	2.8830
RA176	Proposed method	1	**0.2011**	**0.9942**	**1.6827**	**2.2745**	Feder *et al.*	1	0.8921	1.9239	2.4338	2.8156
		2	**2.4295**	**2.8372**	**3.0393**	**3.2186**		2	2.5297	2.8714	3.0653	3.2452
		3	**2.3297**	2.8523	**3.0745**	**3.2801**		3	2.4284	**2.8485**	3.0863	3.2912

- polbooks – a network of books about US politics, sold by Amazon. Edges in the network show the frequent purchases made by buyers. Data compiled by V. Krebs (www.orgnet.com);
- football – a network of American football games between Division IA colleges during the regular Fall 2000 season, as compiled by M. Girvan and M. Newman.

Table 6
The amount of APL error for different values of k and l for the proposed method and for the Feder *et al.* [19] model applied to real-world graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
karate	Proposed method	1	0.0566	**0.0084**	**0.0682**	**0.3178**	Feder *et al.*	1	**0.0522**	0.2892	0.4711	0.6333
		2	**0.4871**	0.7384	0.8293	**0.9096**		2	0.5816	**0.7295**	**0.8276**	0.9167
		3	**0.5210**	**0.7937**	0.9007	**1.0094**		3	0.6279	0.8097	**0.8739**	1.0112
dwt_72	Proposed method	1	**1.7117**	**3.8509**	**4.6897**	**5.2770**	Feder *et al.*	1	3.2516	4.2930	5.5074	5.9069
		2	4.3224	**5.1851**	5.5912	**5.8576**		2	4.3224	5.3146	**5.5575**	5.8580
		3	4.2660	**5.2406**	5.6358	**5.9002**		3	**4.4167**	5.3955	**5.6260**	5.9061
lesmis	Proposed method	1	**0.0223**	**0.0112**	**0.0669**	**0.1923**	Feder *et al.*	1	0.0361	0.1229	0.2418	0.3946
		2	**0.4954**	0.7254	**0.8293**	0.9390		2	0.6058	**0.7189**	0.8416	**0.9387**
		3	**0.7500**	**0.7976**	**0.8724**	**0.9965**		3	0.7965	0.8792	0.9127	1.0122
can_96	Proposed method	1	0.0459	0.0459	**0.0935**	**0.1984**	Feder *et al.*	1	0.0459	0.0459	1.0453	1.8569
		2	**0.5907**	**1.7659**	**2.0946**	**2.3190**		2	1.5547	2.0905	2.2089	2.3313
		3	**1.3738**	**1.9106**	**2.2207**	2.3995		3	1.9970	2.1569	2.2788	**2.3786**
polbooks	Proposed method	1	0.0264	**0.0094**	**0.0304**	**0.1262**	Feder *et al.*	1	**0.0077**	0.1170	0.4212	0.7243
		2	**0.6234**	**0.9291**	**1.1033**	**1.2580**		2	0.7908	1.0057	1.1390	1.2681
		3	1.1674	1.2068	**1.1273**	**1.3097**		3	1.1674	1.2068	1.2439	1.3441
football	Proposed method	1	0.0218	0.0218	0.0218	0.0151	Feder *et al.*	1	0.0218	0.0218	0.0218	**0.0052**
		2	**0.4952**	**0.6642**	**0.7349**	**0.8026**		2	0.5431	0.6897	0.7443	0.8050
		3	0.6218	0.6571	0.6896	0.7352		3	0.6218	0.6571	0.6896	0.7352

Table 7
The amount of ACC error for different values of k and l for the proposed method and for the Feder *et al.* [19] model applied to synthetic graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
SF50	Proposed method	1	**0.0001**	**0.0005**	**0.0013**	**0.0026**	Feder *et al.*	1	0.0003	0.0013	0.0022	0.0036
		2	**0.0033**	0.0046	**0.0052**	**0.0059**		2	0.0036	0.0046	0.0053	0.0059
		3	**0.0036**	0.0051	0.0058	0.0067		3	0.0040	**0.0050**	0.0058	0.0067
SF100	Proposed method	1	**0.0000**	**0.0001**	**0.0003**	**0.0007**	Feder *et al.*	1	0.0001	0.0004	0.0006	0.0010
		2	0.0015	0.0020	0.0022	0.0025		2	0.0015	0.0020	0.0022	0.0025
		3	0.0018	0.0023	0.0025	0.0028		3	0.0018	**0.0022**	**0.0021**	**0.0023**
SF200	Proposed method	1	0.0000	**0.0000**	**0.0001**	**0.0003**	Feder *et al.*	1	0.0000	0.0002	0.0003	0.0005
		2	0.0007	0.0010	0.0012	0.0013		2	0.0007	0.0010	0.0012	0.0013
		3	0.0011	0.0011	0.0012	0.0012		3	0.0011	0.0011	0.0012	0.0012
RA82	Proposed method	1	**0.0001**	**0.0008**	**0.0015**	**0.0023**	Feder *et al.*	1	0.0009	0.0020	0.0027	0.0033
		2	**0.0018**	**0.0027**	**0.0032**	**0.0038**		2	0.0020	0.0028	0.0033	0.0040
		3	0.0017	**0.0027**	**0.0033**	**0.0040**		3	0.0017	0.0028	0.0036	0.0042
RA129	Proposed method	1	**0.0001**	**0.0004**	**0.0007**	**0.0011**	Feder *et al.*	1	0.0002	0.0008	0.0012	0.0016
		2	**0.0012**	0.0017	**0.0020**	0.0024		2	0.0013	0.0017	0.0021	0.0024
		3	0.0012	0.0017	**0.0020**	**0.0024**		3	0.0012	0.0017	0.0021	0.0025
RA176	Proposed method	1	**0.0000**	**0.0002**	**0.0005**	**0.0008**	Feder *et al.*	1	0.0002	0.0006	0.0009	0.0011
		2	0.0009	0.0012	0.0014	0.0016		2	0.0009	0.0012	0.0014	0.0016
		3	**0.0008**	**0.0012**	**0.0014**	**0.0016**		3	0.0009	0.0012	0.0014	0.0017

5.2. Comparing the Proposed Method with the Feder Model

The following sections show the degree of utility distortion resulting from the modification of graph required to produce a (k,l)-anonymous graph. Tables 3, 5, and 7 show the amount of change in AVD, APL, and ACC of anonymous graphs in comparison with original ones for synthetic graph datasets, respectively. Table 3 shows the AVD error for anonymous graphs and compares it with correspond-

Table 8
The amount of ACC error for different values of k and l for the proposed method and for the Feder *et al.* [19] model applied to real-world graphs. In each experiment, the best value is highlighted

Graph	Method	l	k				Method	l	k			
			3	5	7	10			3	5	7	10
karate	Proposed method	1	**0.0001**	**0.0003**	**0.0007**	**0.0022**	Feder *et al.*	1	0.0006	0.0020	0.0034	0.0049
		2	**0.0037**	0.0062	0.0073	**0.0084**		2	0.0047	0.0062	0.0073	0.0086
		3	**0.0041**	**0.0069**	0.0085	**0.0101**		3	0.0052	0.0073	**0.0082**	0.0102
dwt_72	Proposed method	1	**0.0005**	**0.0016**	**0.0024**	**0.0032**	Feder *et al.*	1	0.0012	0.0020	0.0037	0.0046
		2	0.0020	**0.0031**	0.0039	**0.0046**		2	0.0020	0.0033	**0.0038**	0.0046
		3	**0.0020**	**0.0032**	0.0040	**0.0047**		3	0.0021	0.0035	0.0040	0.0047
lesmis	Proposed method	1	**0.0000**	**0.0001**	**0.0002**	**0.0004**	Feder *et al.*	1	0.0001	0.0003	0.0005	0.0009
		2	**0.0012**	0.0019	**0.0023**	0.0028		2	0.0015	0.0019	0.0024	0.0028
		3	**0.0021**	**0.0023**	**0.0026**	0.0032		3	0.0022	0.0026	0.0028	0.0032
can_96	Proposed method	1	0.0000	0.0000	**0.0001**	**0.0001**	Feder *et al.*	1	0.0000	0.0000	0.0008	0.0018
		2	**0.0004**	**0.0017**	**0.0022**	**0.0028**		2	0.0014	0.0022	0.0025	0.0028
		3	**0.0011**	**0.0019**	**0.0025**	0.0030		3	0.0021	0.0024	: 0.0027	**0.0029**
polbooks	Proposed method	1	0.0000	**0.0000**	**0.0001**	**0.0002**	Feder *et al.*	1	0.0000	0.0001	0.0005	0.0010
		2	**0.0008**	**0.0014**	**0.0018**	**0.0022**		2	0.0011	0.0016	0.0019	0.0023
		3	0.0020	0.0021	**0.0019**	**0.0024**		3	0.0020	0.0021	0.0023	0.0025
football	Proposed method	1	0.0000	0.0000	0.0000	0.0000	Feder *et al.*	1	0.0000	0.0000	0.0000	0.0000
		2	**0.0009**	**0.0013**	0.0015	0.0017		2	0.0010	0.0014	0.0015	0.0017
		3	**0.0012**	0.0014	0.0015	0.0017		3	0.0012	0.0014	0.0015	0.0017

Table 9
The average execution time (in seconds) for the proposed method and for the Feder *et al.* model

Graph	Execution time [s] for Feder *et al.* model				Execution time [s] for proposed method			
	Cost time	Solve time	Post-process time	Whole time	Cost time	Solve time	Post-process time	Whole time
karate	**0.0007**	25.0539	**0.0722**	30.5104	0.0362	**4.9168**	0.0760	**11.9279**
dwt_72	**0.0000**	**3.5578**	**0.0152**	**7.3205**	0.3596	3.6221	0.0162	7.5697
lesmis	**0.0012**	5137.9768	0.7115	5174.1992	0.3597	**4491.9676**	**0.7061**	**4519.3890**
can_96	**0.0002**	5047.8342	0.2352	5064.9843	0.5451	**1833.5432**	**0.1329**	**1849.7912**
polbooks	**0.0001**	5241.9255	**0.8628**	5292.6429	2.8282	**5067.6278**	1.8566	**5146.3476**
football	**0.0036**	6829.3870	**1.6720**	6840.1676	1.2615	**6729.2509**	1.9699	**6740.8451**
SF50	**0.0003**	995.9613	0.2244	1002.8781	0.1574	**61.7497**	**0.1952**	**67.0347**
SF100	**0.0003**	5613.8762	**0.7388**	5622.0445	2.3944	**5130.4746**	1.3076	**5140.5596**
SF200	**0.0004**	6936.9346	2.5641	6999.5473	8.6035	**6820.4279**	**1.3090**	**6915.9520**
RA82	**0.0002**	7.0836	0.0628	13.5097	0.4928	**6.3735**	**0.03290**	**11.9961**
RA129	**0.0019**	1727.7701	**0.0915**	1741.1626	4.3179	**32.7018**	0.2013	**51.8183**
RA176	**0.0002**	1795.2124	**0.11818**	1824.7315	8.9662	**52.5578**	0.1280	**82.9223**

ing original graphs for different k, l ($k \in \{3,5,7,10\}, l \in \{1,2,3\}$)and for proposed method and for the Feder *et al.* model are shown. Similarly, Tables 5 and 7 are concerned with APL and ACC errors, while Tables 4, 6, and 8 are about AVD error, APL error, and ACC error of real-world graphs, respectively.

The results presented in Tables 3 and 4 show that, in most cases, the Feder *et al.* model achieves the best AVD, especially in all experiments for $l = 1$. This is rational, since the main objective of the model is to add a minimum number of edges to the original graph. Therefore, the (average) degree of vertices are changes minimally. However, for the cases of $l > 1$, our implementation suggests an approximate approach that uses CV, which means that in some cases the proposed method may be the winner with respect to AVD error. Anyway, the relative differences are negligible. For example, AVD error of the proposed method for SF200, $k = 5$ and $l = 2$ equals to 19.2 while the value is 19.27 for the [19]. As expected, Tables 5–8 confirm the superiority of the proposed method in almost all cases in terms of APL and ACC errors. These values confirm that using a more precise cost function for adding new edges to

produce anonymous graphs, even for an approximated approach in the cases of $l > 1$, yields more useful datasets than the naïve approach used by [19]. These degree of superiority is more evident for APL errors, as the proposed method is optimized for this scenario. For instance, APL error for the proposed method is 0.1262 in polbooks for $k = 10$ and $l = 1$, while the value is 0.7243 for [19]. A similar discussion is true for ACC error, i.e. the proposed method achieves, usually, a result that is better than or equivalent to that shown in [19], since the proposed method attempts to introduce new edges that do not change the structural properties of the underlying original graph. In brief, the results confirm that the trivial idea of trying to only add a minimum number of edges does not necessarily achieve the best results, and that using a more elegant way to introduce new edges may produce more useful datasets.

5.3. Proposed Method Execution Time

In this section, the execution time of the proposed method is reported. The entire execution time consist of cost time, mathematical problem solution time, and post-process time. Table 9 shows these components of the proposed method's execution time. The results of the entire time columns confirm that the anonymization problem can be solved in a reasonable time frame, because it is usually considered as an offline problem. The entire time and solution time of the proposed method are better, in most cases, in comparison with the Feder *et al*. model. But, as far as the cost time columns are concerned, the Feder *et al*. model is quicker, in most cases, than the proposed method.

References

[1] V. V. H. Pham, S. Yu, K. Sood, and L. Cui, "Privacy issues in social networks and analysis: a comprehensive survey", *IET Networks*, vol. 7, no. 2, pp. 74–84, 2018 (doi: 10.1049/iet-net.2017.0137).

[2] B. Palanisamy, L. Liu, Y. Zhou, and Q. Wang, "Privacy-preserving publishing of multilevel utility-controlled graph datasets", *ACM Trans. on Internet Technol.*, vol. 18, no. 2, pp. 1–21, 2018 (doi: 10.1145/3125622).

[3] P. Joshi and C. Kuo, "Security and privacy in online social networks: A survey", in *Proc. IEEE Int. Conf. on Multim. and Expo ICME 2011*, Barcelona, Spain, 2011 (doi: 10.1109/ICME.2011.6012166).

[4] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks", in *WPES '05: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, Aleksandria, VA, USA, 2005, pp. 71–80 (ISBN: 1-59593-228-3).

[5] B. Krishnamurthy and C. Wills, "Characterizing privacy in online social networks", in *Proc. of the 1st Worksh. on Online Soc. Netw.*, Seattle, WA, USA, 2008, pp. 37–42 (doi: 10.1145/1397735.1397744).

[6] T. Truta, M. Tsikerdekis, and S. Zeadally, "Privacy in social networks", in P*rivacy in a Digital, Networked World*, S. Zeadally and M. Badra, Eds. Springer, 2015, pp. 263–289 (ISBN: 978-3-319-08470-1).

[7] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data", *ACM SIGKDD Explor. Newslett.*, vol. 10l no. 2, pp. 12–22, 2008 (doi: 10.1145/1540276.1540279).

[8] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data", in *Privacy, Security, and Trust in KDD. First ACM SIGKDD International Workshop, PinKDD 2007, San Jose, CA, USA, August 12, 2007, Revised Selected Papers*. Springer, 2008, pp. 153–171 (doi: 10.1007/978-3-540-78478-4_9).

[9] R. Trujillo-Rasua and I. G. Yero, "k-metric antidimension: A privacy measure for social graphs", *Inform. Sciences*, vol. 328, pp. 403–417, 2016 (doi: 10.1016/j.ins.2015.08.048).

[10] C.-H. Tai, P. S. Yu, D.-N. Yang, and M.-S. Chen, "Privacy-preserving social network publication against friendship attacks", in *Proc. 17th ACM SIGKDD Int. Conf. on Knowl. Discov. and Data Mining KDD'11*, San Diego, CA, USA, 2011, pp. 1262–1270 (doi: 10.1145/2020408.2020599).

[11] W. Wentao, X. Yanghua, W. Wei, H. Zhenying, and W. Zhihui, "k-symmetry model for identity anonymization in social networks", in *Proc. 13th Int. Conf. on Ext. Database Technol. EDBT'10*, Lausanne, Switzerland, 2010, pp. 111–122 (doi: 10.1145/1739041.1739058).

[12] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "An algorithm for k-degree anonymity on large networks", in *Proc. IEEE/ACM Int. Conf. on Adv. in Soc. Netw. Anal. and Mining ASONAM 2013*, Niagara Falls, ON, Canada, 2013, pp. 671–675 (doi: 10.1145/249251712492643).

[13] T. Tassa and D. Cohen, "Anonymization of centralized and distributed social networks by sequential clustering", *IEEE Trans. on Knowl. and Data Engin.*, vol. 25, no. 2, pp. 311–324, 2013 (doi: 10.1109/TKDE.2011.232).

[14] B. Fung, Y. Jin, J. Li, and J. Liu, "Anonymizing social network data for maximal frequent-sharing pattern mining", in *Recommendation and Search in Social Networks*, Ö. Ulusoy, A. Uz Tansel, and E. Arkun, Eds. Springer, 2015, pp. 77–100 (doi: 10.1007/978-3-319-14379-8_5).

[15] H. Jiang, "A novel clustering-based anonymization approach for graph to achieve privacy preservation in social network", in *Proc. Int. Conf. on Adv. in Mechan. Engin. and Indust. Inform. AMEII 2015*, Zhengzhou, China, 2015, pp. 545–549 (doi: 10.2991/ameii-15.2015.102).

[16] Z. Shiwen, L. Qin, and L. Yaping, "Anonymizing popularity in online social networks with full utility", *Future Gener. Comp. Syst.*, vol. 72, pp. 227–238, 2017 (doi: 10.1016/j.future.2016.05.007).

[17] W. Yazhe, X. Long, B. Zheng, and K. C. Lee, "Utility-oriented k-anonymization on social networks", in *Database Systems for advanced Applications. 16th International Conference, DASFAA 2011, Hong Kong, China, April 22-25, 2011, Proceedings, Part I*, J. X. Yu, M. H. Kim, and R. Unland, Eds. Springer, 2011, pp. 78–92 (doi: 10.1007/978-3-642-20149-3_8).

[18] C. Watanabe, T. Amagasa, and L. Liu, "Privacy risks and countermeasures in publishing and mining social network data", in *Proc. 7th Int. Conf. on Collab. Comput.: Network., Appl. and Worksharing CollaborateCom 2011*, Orlando, FL, USA, 2011 (doi: 10.4108/icst.collaboratecom.2011.247177).

[19] T. Feder, S. U. Nabar, and E. Terzi, "Anonymizing graphs", arXiv:0810.5578 [cs.DB].

[20] K. Stokes and V. Torra, "Reidentification and k-anonymity: a model for disclosure risk in graphs", *Soft Comput.*, vol. 16, no. 10, pp. 1657–1670, 2012 (doi: 10.1007/s00500-012-0850-4).

[21] J. Cheng, A. W.-C. Fu, and J. Liu, "K-isomorphism: privacy preserving network publication against structural attacks", in *Proc. of the ACM SIGMOD Int. Conf. on Manag. of Data SIGMOND'10*, Indianapolis, Indiana, USA, 2010, 459–470 (doi: 10.1145/1807167.1807218).

[22] K. Liu and E. Terzi, "Towards identity anonymization on graphs", in *Proc. ACM SIGMOD Int. Conf. on Manag. of Data SIGMOND'08*, Vancouver, Canada, 2008, pp. 93–106 (doi: 10.1145/1376616.3776629).

[23] B. Zhou and J. Pei, "The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks", Knowl. and Inform. Syst., vol. 28, no. 1, p. 47–77, 2011 (doi: 10.1007/s10115-010-0311-2).

[24] L. Zou, L. Chen, and M. Özsu, "K-automorphism: a general framework for privacy preserving network publication", in *Proc. of the VLDB Endowment*, vol. 2, no. 1, pp. 946–957, 2009 (doi: 10.14778/1687627.1687734).

[25] M. I. H. Ningga and J. H. Abawajy, "Utility-aware social network graph anonymization", J. of Netw. and Comp. Appl., vol. 56, pp. 137–148, 2015 (doi: 10.1016/j.jnca.2015.05.013).

[26] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on network", *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 341–366, 2017 (doi: 10.1007/s10462-016-9484-8)

[27] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks", Tech. Rep. no. 07-19, Computer Science Department, University of Massachusetts Amherst, 2007 [Online]. Available: https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1175;context=cs_faculty_pubs

[28] K. Stokes and V. Torra, "On some clustering approaches for graphs", in *Proc. IEEE Int. Conf. on Fuzzy Syst. FUZZ-IEEE 2011*, Taipei, Taiwan, 2011, pp. 409–415 (doi: 10.1109/FUZZY.2011.6007447).

[29] J. Casas-Roma, "Privacy-preserving on graphs using randomization and edge-relevance", in *Modeling Decisions for Artificial Intelligence 11th International Conference, MDAI 2014, Tokyo, Japan, October 29-31, 2014. Proceedings*, V. Torra, Y. Narukawa, Y. Endo, Eds. *LNCS*, vol. 8825. Springer, 2014, pp. 204–216 (doi: 10.1007/978-3-319-12054-6_18).

[30] P. Samarati, "Protecting respondents' identities in microdata release", *IEEE Trans. Knowl. Data Engin.* (*TKDE*), vol. 13, no. 6, pp. 1010–1027, 2001 (doi: 10.1109/69.971193).

[31] L. Sweeney, "k-anonymity: a model for protecting privacy", *Int. J. of Uncert., Fuzziness Knowl.-Based Syst.* (*IJUFKS*), vol. 10, no. 5, pp. 557–570, 2002 (doi: 10.1142/S0218488502001648).

[32] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks", in *Proc. IEEE 24th Int. Conf. on Data Engin.*, Cancun, Mexico, 2008, pp. 506–515 (doi: 10.1109/ICDE.2008.4497459).

[33] Y. Wang, L. Xie, B. Zheng, and K. C. K. Lee, "High utility k-anonymization for social network publishing", *Knowl. and Inform. Syst.*, vol. 41, no. 3, pp. 697–725, 2014 (doi: 10.1007/s10115-013-0674-2).

[34] X. He, J. Vaidya, B. Shafiq, N. Adam, and V. Atluri, "Preserving privacy in social networks: a structure-aware approach", in *Proc. IEEE/WIC/ACM Int. Joint Conf. on Web Intell. and Intell. Agent Technol. WI-IAT'09*, Milan, Italy, 2009, pp. 47–54 (doi: 10.1109/WI-IAT.2009.108).

[35] G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network", *Science*, vol. 311, no. 5757, pp. 88–90, 2006 (doi: 10.1126/science.1116869).

[36] R. Mortazavi and S. H. Erfani, "An effective method for utility preserving social network graph anonymization based on mathematical modeling", *Int. J. of Engin.*, vol. 31, no. 10, pp. 1624–1632, 2018 [Online]. Available: http://www.ijeir.info/article_81694_e3849caf4384bbe53aed25f3afd8d93e.pdf

[37] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", *Proc. of the Nat. Acad. of Sci. USA*, vol. 99, no. 12, pp. 7821–7826, 2002 (doi: 10.1073/pnas.122653799).

[38] L. J. Lu and M. Zhang, "Edge betweenness centrality", in *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, K.-H. Cho, H. Yokota, Eds. New York, NY: Springer, 2013, pp. 647–648 (doi: 10.1007/978-1-4419-9863-7_874).

[39] CPLEX, GAMS. The solver manuals, GAME/CPLEX; 1996 [Online]. Available: https://www.gams.com/latest/docs/S_CPLEX.html

Seyedeh Hamideh Erfani received her B.Sc. degree and the M.Sc. degree from the Department of Computer Engineering, Ferdowsi University of Mashhad, Iran, in 2010 and 2013, respectively, from the Department of Computer Engineering, Ferdowsi University of Mashhad. She is a lecturer at Damghan University. Her main research interests include computer vision, machine learning, and information systems.
E-mail: sh.erfani@du.ac.ir
School of Engineering
Damghan University
36716-41167, Damghan, Iran

Reza Mortazavi received his Ph.D. degree in Software Engineering in 2015 from Tarbiat Modares University, Iran. Currently he is an Assistant Professor at the School of Engineering, Damghan University, Iran. His current research interests include data privacy and machine learning.

E-mail: r_mortazavi@du.ac.ir
School of Engineering
Damghan University
36716-41167, Damghan, Iran