International Journal of Engineering

Journal Homepage: www.ije.ir

An Effective Method for Utility Preserving Social Network Graph Anonymization Based on Mathematical Modeling

R. Mortazavi*, S. H. Erfani

School of Engineering, Damghan University, Damghan, Iran.

P A P E R I N F O

Paper history:
Received 20 March 2018
Received in revised form 25 May 2018
Accepted 17 August 2018

Keywords:
Mathematical Modeling
Graph Anonymization
Graph Modification
Social Network
Privacy
Database Security

A B S T R A C T

In recent years, privacy concerns about social network graph data publishing has increased due to the widespread use of such data for research purposes. This paper addresses the problem of identity disclosure risk of a node assuming that the adversary identifies one of its immediate neighbors in the published data. The related anonymity level of a graph is formulated and a mathematical model is proposed to solve the problem. The application of the method on a number of synthetic and real-world datasets confirms that the method is general and can be used in different contexts to produce superior results in terms of the utility of the anonymized graph.

***doi**: 10.5829/ije.2018.31.10a.03*

1. INTRODUCTION

The vast dissemination of social networks data for research purposes in recent years has motivated the privacy concerns of individuals. Participants most often expressed privacy concerns about social network services, and the sharing of personal information online left them scared and feeling vulnerable [1]. In fact, in the case of online social networks, there are many reasons such as privacy that restrict datasets to be public which limits the application of interesting machine learning techniques [2]. For instance, in health information dissemination, the shift by health care providers towards more dynamic methods of information sharing can be the cause of threat to the privacy of social media platforms [3].

Usually, a social network is represented as a simple graph in which nodes are individuals and edges correspond to a relationship between individuals. Respecting the privacy issues of involving individuals, the owner of the social network has to anonymize the underlying graph before publishing it. Despite the fact that privacy concerns in releasing social network data have been pinpointed, there is no agreement on the definition of privacy or anonymity that should be used for such data [4].

This paper considers a scenario in which an adversary attempts to re-identify a real-world entity in the graph based on its neighborhood. The main contributions of the paper include the following items:

- A new anonymity measure is defined to quantify the privacy level of a given graph with respect to the anonymity model.
- A mathematical model is proposed to capture a solution that minimizes the information loss of the anonymization process.
- Four different variants of the base mathematical model are proposed by changing its objective function.
- A set of experimental evaluations of the method on multiple synthetic and real-world networks are conducted.

Notation: Let G=(V, E) be a simple graph, in which V denotes the set of vertices[2] and E the set of edges. The

*Corresponding Author Email: *r_mortazavi@du.ac.ir* (R. Mortazavi)

[2]The terms *vertex* and *node* are assumed with the same meaning in this study.

Please cite this article as: R. Mortazavi, S. H. Erfani, An Effective Method for Utility Preserving Social Network Graph Anonymization Based on Mathematical Modeling, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 31, No. 10, (October 2018) 1624-1632

vertices are referred to by $v_i \in V$ and {V_i, V_j} is used to show an undirected edge between vertices v_i and v_j. Assume $n = |V|$ denotes the number of vertices. The degree of a vertex $v_i \in V$ is denoted by $deg(v_i)$ and the shortest path length between v_i and v_j is denoted by $d(v_i, v_j)$. The average of node degrees is also shown as AVD. Additionally, $N(v) = \{u \in V : \{u, v\} \in E\}$ shows neighbors of v. All notations and abbreviations are defined in Table 1.

Roadmap: The remaining of the paper is organized as follows: Section 2 introduces a number of related works. Section 3 defines the anonymity measure that is used to quantify the privacy and proposes a model to satisfy the related requirement. Section 4 evaluates the proposed method on a number of synthetic and real-world datasets and finally, Section 5 concludes the paper.

2. RELATED WORK

Currently, large amounts of private information are gathered in social networks. It is shown that trivial procedure of removing names or other primary identifiers cannot produce a suitable data for publication [5]. To address the problem, similar to traditional relational data anonymization, some procedures reported in literature [6-9], are proposed. For instance, perturbation-based approaches that randomly insert or delete edges of the graph are considered by Ying and Wu [10]. Hay et al. investigate the potential of structural queries on graphs for the re-identification of vertices and propose an anonymization method based on k-anonymity. Liu et al. [11] introduced k-degree anonymity and Casas Roma et al. [5] presented an efficient algorithm to fulfill the requirement. Feder et al. [4] defined (k, l)-anonymity as a relaxed version of k-anonymity. Based on the definition, a graph G = (V, E) is called (k, l)- anonymous if for every vertex $v \in V$, there exists a set of vertices $U \subseteq V$ not containing v such that $|U| \geq k$ and for each $u \in U$, the two vertices u and v share at least l neighbors. The authors devised a random matching algorithm for the special case of (k,1)-anonymity that uses edge addition to produce an anonymous graph. In order to preserve the utility of the produced graph in terms of the number of added edges, the authors consider matching only between problemistic (called deficient) vertices in the original graph. However, Stokes and Torra showed that the original definition of (k, l)-anonymity suffers from some real-world problems related to privacy of involved vertices and refined the definition.

TABLE 1. Abbreviations and notations used in the paper

AVD	Average Vertex Degree
ABC	Average Betweenness Centrality
ACC	Average Clustering Coefficient
APL	Average Path Length
C	Cost matrix
$N(v)$	Neighbors of v
$d(v_i, v_j)$	The shortest path length between v_i and v_j
$deg(v)$	Degree of v
k	Privacy parameter of the model
n	Order of the graph$\|V\|$, its number of vertices

Definition 1 (Definition 21 of Liu and Terzi [12]). G=(V, E) is (k, l)-anonymous if for any vertex $v \in V$ and for all subset $S \subseteq N(v)$ of cardinality $|S| \leq l$ there are at least k distinct vertices $\{v_i\}_{i=1}^{k}$ such that $S \subseteq N(v_i)$ for i∈{1,k}.

According to Definition 1 and for the special case of $l = 1$ and $k \geq 2$, if an adversary identifies a (real) neighbor (v_j) of a vertex v_i in the graph, she would not be able to re-identify v_i using v_j with probability more than $1/k$. The adversary may use different background knowledge such as the degree of the vertex v_j [5] to identify it in G and then tries to cascade the knowledge to other nodes. Liu and Terzi [12] suggested cloning problemistic vertices, i.e., to insert a number of new vertices with the same neighborhood of the problemistic vertices. However, as stated by Liu and Terzi [12], these fake nodes cause the algorithm to lose its status as an anonymization method. Additionally, an adversary who knows about the anonymization algorithm can re-identify the victim with probability greater than $1/k$.

3. THE PROPOSED METHOD

In this paper, we propose a general framework for the (k, l)-anonymity problem. The solution uses only edge addition to anonymize the original graph and is formulated as a mixed integer program. According to definition 1, Section 3.1 defines a measure to quantify graph anonymity. Section 3.2 presents the general solution of the problem. Different variants of the model are then introduced in Section 3.3. Finally, Section 3.4 describes the algorithm of the proposed method.

3. 1. The Anonymity of The Original Graphs In this section, we define the anonymity of a graph G=(V, E) based on definition 1.

$$anonymity(G, k) = \min_i \frac{|\{v_j : v_j \in N(v_i), deg(v_j) \geq k\}|}{deg(v_i)} \quad (1)$$

Equation (1) calculates the anonymity level of G for different values of k. For each node v_i, the number of its neighbors with degrees greater than k is calculated and is normalized by $deg(v_i)$. The minimum of the values for all nodes is considered as the anonymity level for G. The larger the value, the better the graph is protected against such adversaries.

3. 2. The Mathematical Model The proposed solution attempts to minimize modifications to the original graph while preserving the privacy of nodes with respect to Definition 1. The solution as a mixed integer program contains the following parts:

- Definition of sets: The indexes of vertices $v_i \in V$ in the graph are saved in I, i.e., $I = \{1,2, \dots, n\}$.
- Fixed parameters and constants: Two parameters n and k represent the number of vertices, and k of the anonymization model, respectively. Additionally, the parameter c_{ij} denotes the cost of adding the edge $e_{ij} \notin E$ that connects v_i and v_j both in V. 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 G, and a positive value for disjoint ones. In order to reduce the size of passed data to the solver, only upper triangular part of C is passed to the solver.
- Independent problem variables: The solution consists of vertices to be connected. 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 the related vertices are to be connected. In order to decrease the space complexity of the final model we only consider x_{ij} for $j > i, i, j \in I$.
- Objective function: The objective is to minimize the aggregate cost of changes with respect to the original graph, i.e.,
$$\min_{x_{ij}} \sum_{i,j \in I, i<j} c_{ij} x_{ij}.$$
- Constraints: The constraints fall into two categories: edge preserving constraints, and anonymization constraints.
  (a) Edge preserving constraints: None of the existing edges in the original graph can be removed, i.e., $x_{ij} = 1, \forall e_{ij} \in E$. In order to speed up the calculation, the warm-start strategy is used in which $x_{ij} = 1$ for all of the connected vertices v_i and v_j.
  (b) Anonymization constraints: These constraints enforce that the minimum degree of each vertex has to be at least k, i.e.,
  $\sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} \geq k \ \ \forall i \in I, \deg(v_i) < k.$

3. 3. The Variants In this work, four different variants of the general mathematical model have been applied to the original graph. The difference between these variants is in the different cost functions that are supposed to be minimized in the objective function. Specifically, the following models are considered:

- Model 1 (M1): In this model, it is assumed that all edges introduce the same level of destruction to the original graph G, i.e., $c_{ij} = 1, \forall i, j \in I$. Therefore, the model minimizes the total number of added edges and chooses a random edge when there are different candidates to be added to G. It is an implementation of the main objective reported in literature [4] that is used for comparison throughout the experiments in the paper.
- Model 2 (M2): This model tries to add edges that minimize the total length (in the original graph) between connected vertices. This model assumes that connecting two closer vertices is less harmful to the utility, therefore it uses $c_{ij} = d(v_i, v_j)$.
- Model 3 (M3): It is interesting to add new edges that minimally change the average path length (APL) of G as an important property of the graph. This model tries to add edges that decrease APL minimally. It is hard to calculate 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 a 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 by addition of e_{ij} to G, the total value of the change in the APL for a set of new edges $E' \subset V \times V \backslash E$ is approximated by $\sum_{e_{ij} \in E'} c_{ij}$.
- Model 4 (M4): Inspired from [13], added edges in this model are chosen to connect similar vertices in terms of the overlap of their neighbors. This model assumes that the cost of connecting two disjoint vertices with very different neighbors is more than the cost of connecting two similar vertices with some more overlap in the neighborhoods. For this model $c_{ij} = |N(v_i) \cup N(v_j)| / |N(v_i) \cap N(v_j)|$ is used.

3. 4. The Algorithms The algorithm of the proposed method is given in Algorithm 1. The function accepts the adjacency matrix M of the graph G, selected variant of the model Var, and the privacy parameter k as input and produces the anonymized graph G_k. In the first step, the Compute Cost function is called to compute C (Line 1). Next, the optimization problem is solved to obtain the modified adjacency matrix M' (Line 2). In Line 3, the anonymized graph G_k is generated based on M'. Finally, G_k is returned (Line 4). Algorithm 2 shows the ComputeCost function that takes the

adjacency matrix M and the selected model variant Var. Based on the selected model variant, a different cost matrix C has to be computed as output. If M1 is chosen, c_{ij} will be equal to 0 when v_i is connected to v_j in the original graph G, otherwise it is set to 1, i.e., $c_{ij} = 1 - M[i,j]$. This is computed in Line 1. For the case of M4 (Line 2), the cost matrix can be computed efficiently using bitwise operators **or** and **and** for the union and intersection functions, respectively. The **size** function also computes the number of 1st of its operand (Line 3). It is notable that a very small value ϵ is added to the denominator to prevent arithmetic errors. The other two options for Var, i.e., M2 and M3 are considered in Lines 4-16. In Line 5, the graph-theoretic shortest distance between all nodes are computed using the Floyd-Warshall algorithm [14]. For M2, the computed distances d_{ij} minus 1 are returned[3] as c_{ij}(Line 6). For M3, the function calculates the average value of shortest paths and saves it in APL (Line 8). Then, in the following steps for each i,j, the value of c_{ij} is calculated. If v_i is connected to $v_j (i \neq j)$, c_{ij} is set to 0 (Line 11). Otherwise, these vertices are connected temporarily (Line 14) and the new APL is computed (Line 15) and saved in APL' (Line 16). The difference between APL and APL' is stored in c_{ij} (Line 17). Finally, the cost function C is returned (Line 18).

ALGORITHM 1. Generating the anonymized graph G_k

Input: M (adjacency matrix of G), Var (model variant), k (privacy parameter)
Output: Anonymized graph G_k
Function G_k=AnonymizeGraph(M, Var, k)
1 C=ComputeCost(M, Var) // Algorithm 2
2 Solve the optimization problem introduced in Section 3.2 to obtain $M' = [x_{ij}]$
3 G_k = graph(M')
4 **Return** G_k
END Function

ALGORITHM 2. Computation of the cost matrix

Input: M (adjacency matrix of G), Var (model variant)
Output: The cost matrix C
Function C=ComputeCost(M,Var)
1 **If** Var ==M1 **Return** Ones(n,n)-M
2 **Elseif** Var ==M4
3 **Return** $C = [c_{ij}]$ where
c_{ij}=**Size**(**Or**($M[i,:], M[j,:]$)) /
(ϵ+**Size**(**And**(($M[i,:], M[j,:]$))) $\forall i,j \in I$
4 **Else** // M2 or M3
5 $D = [d_{ij}]$ =Floyd-Warshall(M)
6 **If** Var ==M2 **Return** $D - 1$
7 **Else** // M3
8 APL =average(D)
9 **Foreach** $i,j \in I$
10 **If** $M[i,j] == 1$
11 $c_{ij} = 0$
12 **Else**
13 $M' = M$
14 $M'[i,j] = M'[j,i] = 1$
15 D' = Floyd-Warshall(M')
16 APL'=average(D')
17 $c_{ij} = APL - APL'$
18 **Return** C
End Function

[3]If v_i is connected to v_j $(i \neq j)$ then $d_{ij} = 1$. So, $c_{ij} = d_{ij} - 1 = 0$ is considered.

4. EXPERIMENTAL EVALUATIONS

In this section, different experiments are conducted to evaluate the efficacy of the proposed method. In all experiments, a PC with Intel Core 2 Duo 3.6 GHz CPU, 16 GB of main memory and Windows 10 operating system is used. The CPLEX optimization engine is used to solve the mathematical problem. Section 4.1 introduces the datasets. Section 4.2 evaluates the risk of the original datasets based on Equation (1). Section 4.3 reports the usefulness of the anonymized graphs based on different utility measures. At the end of this section, execution times are reported in Section 4.4.

4. 1. The Graph Datasets The proposed method is applied to two synthetic and two real-world datasets to observe its performance in different situations. The synthetic datasets are summarized as follows:

- SF50 and SF500: Two scale-free datasets based on Barabasi's model. These graphs are connected graphs where vertex degrees are drawn from a power-law distribution similar to real-world social networks. The datasets are generated using the tool introduced in [15].
- RA82: A random network based on Erdos-Renyi model in which vertices are connected based on probability $p = 0.03$.

Additionally, three real-world datasets are tested to assess our method in different topologies. The first two datasets are also used in [13].

- PolBooks: A network of books about US politics sold by Amazon.com. Edges in the network show the frequent purchasing of buyers. The data is compiled by V. Krebs (www.orgnet.com).
- Football: the network of American football games between Division IA colleges during regular season Fall 2000, as compiled by Girvan and Newman [16].
- Dwt1005: An undirected graph from Everstine's collection that is included in the Harwell-Boeing database[4].

[4]https://math.nist.gov/MatrixMarket/data/Harwell-Boeing/dwt/dwt_1005.html

Structural properties of the graphs are given in Table 2. For each graph, the Average Path Length (APL), the Average Clustering Coefficient (ACC), and the Average Betweenness Centrality (ABC) are reported. The APL is a concept in the network topology that is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. It is a measure of the efficiency of information or mass transport on a network[5]. The ACC is the average of local clustering coefficients of graph nodes. The measure for a node quantifies how close its neighbors are to being a clique and determines whether a graph is a small-world network [17]. Moreover, ABC is the average of betweenness centrality of all nodes. The measure for each node captures the number of shortest paths in the graph passing through the node[6].

4. 2. The Anonymity Measure Table 3 shows the measure for the synthetic and real-world graphs introduced in section 3.1. The results confirm that the anonymity of graph diminishes very fast when k increases. For instance, for $k = 4$ in Polbooks, there is at least one node v for which the degree of half of its neighbors is lower than 4. Therefore, if an adversary identifies one of the neighbors randomly, the probability to *limit* $\boldsymbol{v}$ in a group with lower than 4 members is more than 0.5.

TABLE 2. Structural properties of Synthetic and Real-world graphs

Dataset	Vertices	Edges	APL	ACC	ABC
SF50	50	96	2.8433	0.1178	45.1600
SF500	500	996	3.8837	0.0331	719.4880
RA82	82	94	5.7350	0.0000	191.7683
Polbooks	105	441	3.0788	0.4875	108.0952
Football	115	613	2.5082	0.4032	85.9652
Dwt1005	1005	3808	13.2600	0.9098	6154.5174

TABLE 3. Anonymity of different datasets based on Equation (1)

	$k = 3$	4	5	6	7	8	9	10	11	12	13
SF50	1.00	0.40	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SF500	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RA82	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Polbooks	1.00	0.67	0.50	0.50	0.25	0.00	0.00	0.00	0.00	0.00	0.00
Football	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.67	0.60	0.36	0.00
Dwt1005	1.00	1.00	0.67	0.33	0.17	0.00	0.00	0.00	0.00	0.00	0.00

[5] https://en.wikipedia.org/wiki/Average_path_length
[6] https://en.wikipedia.org/wiki/Betweenness_centrality

Generally, the results show the importance of considering the anonymization process, especially for $\boldsymbol{k \geq 4}$ with respect to Definition 1 for different datasets.

4. 3. The Utility of Anonymized Graphs This section reports the amount of error introduced in the anonymized graph after applying the proposed method. The performance measures are based on the $AVD, APL, ACC,$ and ABC that are defined in Section 4.1. For example, $\Delta APL = |APL_2 - APL_1|$ where APL_1 and APL_2 are the APL of the original and anonymized graphs, respectively. Other measures are defined similarly. The lower the value, the better the anonymization procedure and the anonymized graph is more similar to the original one. Figures 1-4 show these quantities for different datasets and models. The results confirm for all models that when k increases, all error measures increase. For instance, $\Delta APL = 0.1241$ in SF50 for $k = 3$ and M1, while the value increases to $\Delta APL = 0.4604$ for $k = 5$. Similarly, $\Delta AVD = 0.572$ in SF500 for $k = 3$ and M3, while the error reaches $\Delta AVD = 6.86$ for $k = 10$. Figure 1 shows that M1 is always successful to reach the best ΔAVD, since the measure is minimized when the number of added edges is minimized, which is the main task of M1 [4]. The results in Figure 2 indicate that in most cases, M3 achieves the best ΔAPL. This is rational since the weights in M3 represent the cost related to APL, therefore edges with smaller costs have more priority to be added to the solution of the model. For instance, in Dwt1005, $\Delta APL = 9.5045$ for M1 and $k = 10$, while the quantity is $\Delta APL = 1.1955$ for M3 which means a significant improvement in compare with literature [4]. Figure 3 shows that in all cases except for the Polbooks and Dwt1005, M1 produces a more useful graph in terms of ΔACC. This confirms that ACC is more sensitive to the number of added edges than the way they are added. Finally, Figure 4 validates again the claim, i.e., a model with the minimum number of added edges does not necessarily result in the best utility for the anonymized graph. For SF50, in $k \leq 6$, M3 usually produces the lowest errors, while for $k > 6$, M4 achieves the most useful datasets in terms of ΔABC. Interestingly, M3 is also the winner in Polbooks, Football and Dwt1005 graphs. However, for RA82, M2 has the best performancein all cases. This may be related to the structure of RA82, which is a very sparse graph (its initial density is lower than 0.03) while other graphs are at least twice denser than RA82. As the figure illustrates, RA82 is more sensitive than the other three graphs in terms of ΔABC. Similar comparison based on these measures confirm that RA82 is more sensitive to the anonymization procedure. For instance, ΔAPL values of RA82 are about three times more than the values for other datasets. Generally speaking, the denser the original graph, the lower utility will be lost

during anonymization for the proposed method which is based on edge addition. Therefore, it is suggested to social network owners to allow their network grows to a minimum threshold (for example in terms of its density), and then attempt to apply anonymization procedure on it.

4. 4. Execution Time In this section, the running time of experiments are reported. The main time-consuming tasks consist of solving the optimization problem and computing the cost matrix. Other computations in each separate experiment require less than 2 seconds and are ignored.

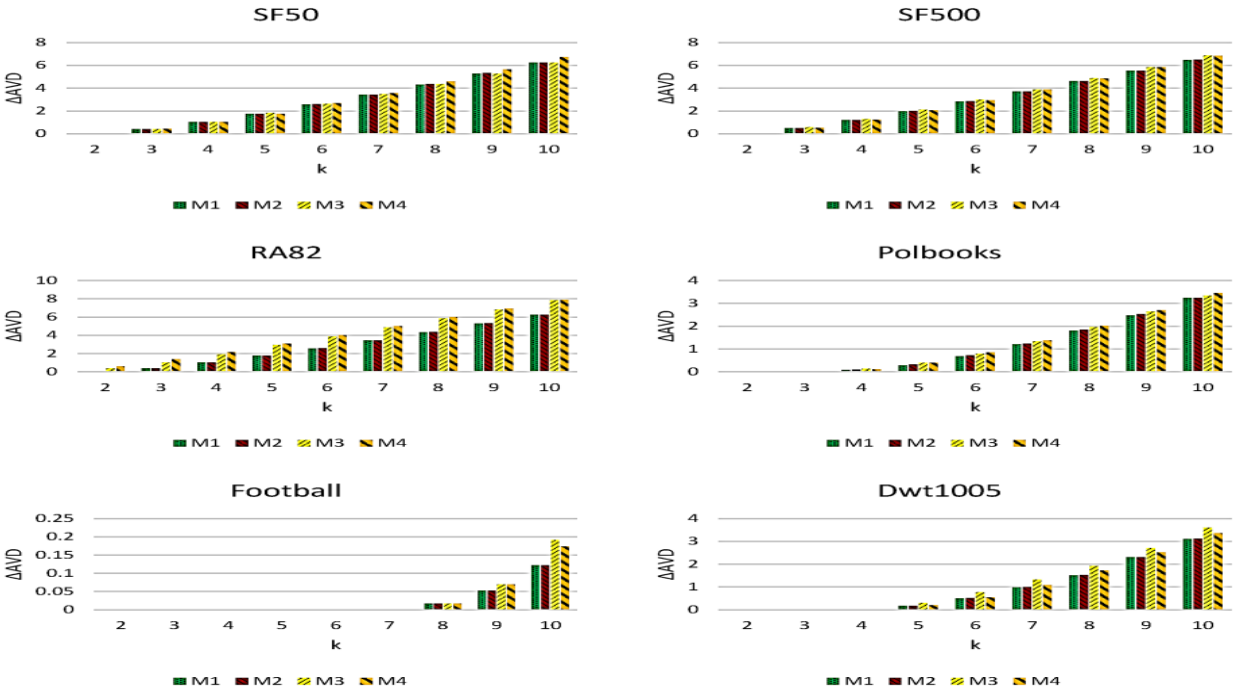

Figure 1. The value of ΔAVD for different datasets and models in $k = 2\ldots10$

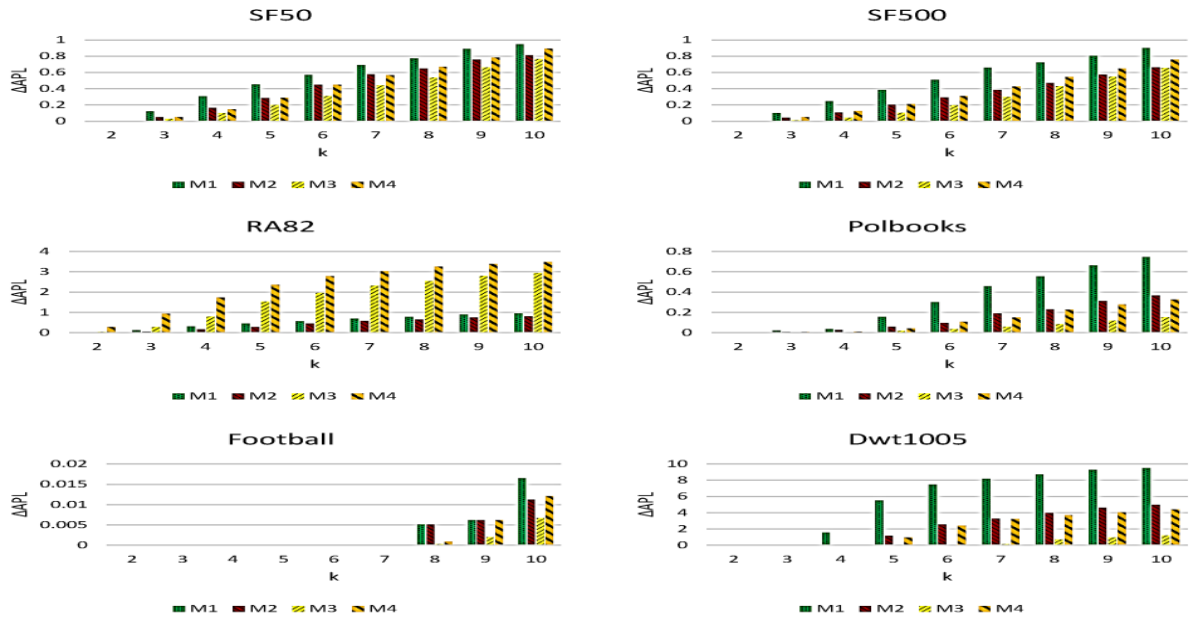

Figure 2. The value of ΔAPL for different datasets and models in $k = 2\ldots10$

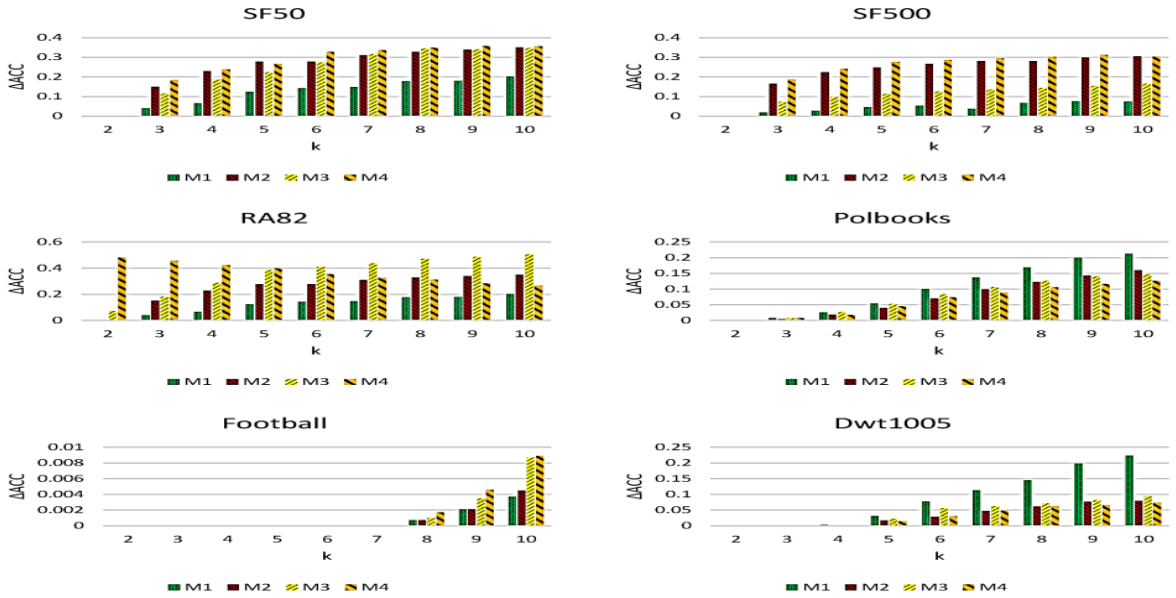

Figure 3. The value of ΔACC for different datasets and models in $k = 2\ldots10$

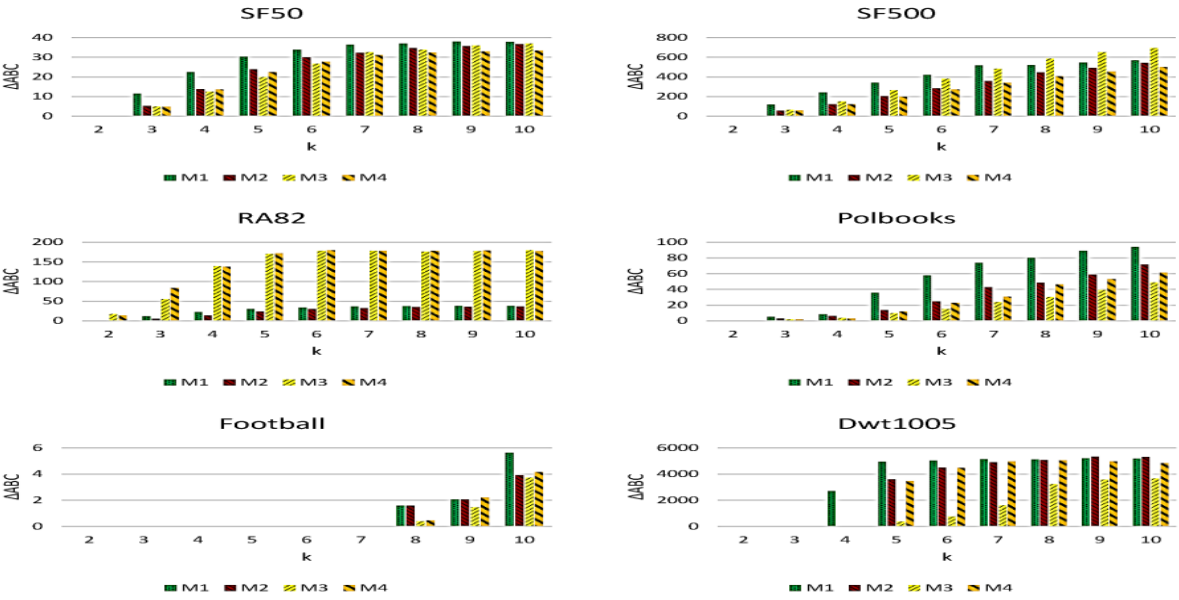

Figure 4. The value of ΔABC for different datasets and models in $k = 2\ldots10$

Table 4 shows the time required to solve the problem for each dataset and model. The results confirm that in most cases, the problem can be solved in a reasonable time. There may be some cases in which the engine requires a considerable time to produce the optimal solution. It is notable that graph anonymization task is usually considered as an offline procedure. If solving time matters, the optimization engine can be tuned to stop if a near to optimal solution is achieved using the *gap*>0[7].

The other time-consuming task is the Compute Cost function in which the cost matrix C is computed. This task is independent of the privacy parameter k. Table 5 shows the execution time of the function for different datasets and model variants.

[7]The parameter controls the relative termination tolerance. All experiments in this paper use gap=0 to find the exact optimal solution of the optimization problem.

TABLE 4. The average solve time (in second) for each dataset and model variant

Dataset	M1	M2	M3	M4
SF50	0.23	0.23	0.28	0.22
SF500	3.25	2.85	3.09	339.89
RA82	0.27	0.26	0.26	161.55
Polbooks	0.24	0.25	0.25	0.24
Football	0.21	0.20	0.20	0.20
Dwt1005	6.38	6.25	5.74	5.47

TABLE 5. Execution time of the Compute Cost function (in second) for each dataset and model variant

Dataset	M1	M2	M3	M4
SF50	<0.01	<0.01	0.05	0.01
SF500	<0.01	0.01	368.85	1.31
RA82	<0.01	<0.01	0.26	0.02
Polbooks	<0.01	0.01	0.71	0.02
Football	<0.01	0.01	0.89	0.03
Dwt1005	<0.01	0.04	6641.12	13.45

These running times are almost negligible with respect to the solving time of the optimization problem except for the case of M3. In this case, it is required to call Floyd-Warshall multiple times, so the execution time gets considerable especially for large graphs.

5. CONCLUSIONS

This paper realizes a procedure for graph anonymization. The adversary is assumed to be able to identify a neighbor of a victim node. Regarding this attack model, an anonymity measure is defined. It is shown that the measure decreases sharply when the anonymity parameter increases slightly. A general mathematical model is proposed to increase the anonymity measure using edge addition. Four different variants of the general model are proposed. The procedures are evaluated on different synthetic and real-world graphs. The results show that trying to minimize the number of added edges is not usually a good objective to produce a useful anonymized graph. Additionally, the results show that highly sparse graphs are very sensitive to different utility measures. Evaluating the proposed method for other general graph utility measures is considered to be accomplished as a future work. Moreover, it is interesting to devise some heuristics to anonymize larger social networks [18] without using mixed integer programming approach.

6. REFERENCES

1. Asadi Saeed Abad, F. and Hamidi, H., "An Architecture for Security and Protection of Big Data", ***International Journal of Engineering***, Vol. 30, No. 10, (2017), 1479-1486.
2. Hemati, H., Ghasemzadeh, M. and Meinel, C., "A Hybrid Machine Learning Method for Intrusion Detection", ***International Journal of Engineering, Transactions C: Aspects***, Vol. 29, No. 9, (2016), 1242-1246.
3. Sharma, P. and Kaur, P. D., "Effectiveness of web-based social sensing in health information dissemination-A review", **Telematics and Informatics**, Vol. 34, No. 1, (2017), 194–219.
4. Feder, T., Nabar, S. U. and Terzi, E., "Anonymizing graphs",CoRR, abs/0810.5578, (2008).
5. Casas-Roma, J., Herrera-Joancomartí, J. and Torra, V., "k-Degree anonymity and edge selection: improving data utility in large networks", ***Knowledge and Information Systems***, Vol. 50, No. 2, (2017), 447–474.
6. Sweeney, L., "k-anonymity: A model for protecting privacy", ***International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems***, Vol. 10, No. 5, (2002), 557–570.
7. Mortazavi, R. and Jalili, S., "Enhancing aggregation phase of microaggregation methods for interval disclosure risk minimization", ***Data Mining and Knowledge Discovery***, Vol. 30, No. 3, (2016), 605–639.
8. Salari, M., Jalili, S. and Mortazavi, R., "TBM, a transformation based method for microaggregation of large volume mixed data", ***Data Mining and Knowledge Discovery***, Vol. 31, No. 1, (2017), 65–91.
9. Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M., "L-diversity: Privacy beyond k-anonymity", ***ACM Transaction on Knowledge Discovery from Data (TKDD)***, Vol. 1, No. 1, (2007), 1–52.
10. Ying, X. and Wu, X., "Randomizing social networks: a spectrum preserving approach", in Proceedings of the 2008 SIAM International Conference on Data Mining, SIAM, (2008), 739–750.
11. Liu, K. and Terzi, E., "Towards identity anonymization on graphs", in Proceedings of the 2008 ACM SIGMOD international conference on Management of data, ACM, (2008), 93–106.
12. Stokes, K. and Torra, V., "Reidentification and k-anonymity: A model for disclosure risk in graphs", ***Soft Computing***, Vol. 16, No. 10, (2012), 1657–1670.
13. Ninggal, M. I. H. and Abawajy, J. H., "Utility-aware social network graph anonymization", ***Journal of Network and Computer Applications***, Vol. 56, (2015), 137–148.
14. Floyd, R. W., "Algorithm 97: Shortest Path", **Communications of the ACM**, Vol. 5, No. 6, (June 1962), 345. doi>10.1145/367766.368168
15. Hadian, A. and Nobari, S., Minaei-Bidgoli, B., Qu, Q., "Roll: Fast in-memory generation of gigantic scale-free networks", in Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16, ACM, New York, NY, USA, (2016), 1829–1842.
16. Girvan, M. and Newman, M. E., "Community structure in social and biological networks", in Proceedings of the national academy of sciences, Vol. 99, No. 12, (2002), 7821–7826.
17. Watts, D. J. and Strogatz, S. H., "Collective dynamics of 'small-world' networks", ***Nature***, Vol. 393, No. 6684, (1998), 440-442.
18. Mohammadi, A. and Hamidi, H., "Analysis and evaluation of privacy protection behavior and information disclosure concerns in online social networks", ***International Journal of Engineering***, Vol. 31, No. 8, (2018), 1234-1239.

An Effective Method for Utility Preserving Social Network Graph Anonymization Based on Mathematical Modeling

R. Mortazavi, S. H. Erfani

School of Engineering, Damghan University, Damghan, Iran.

P A P E R I N F O

Paper history:
Received 20 March 2018
Received in revised form 25 May 2018
Accepted 17 August 2018

Keywords:
Mathematical Modeling
Graph Anonymization
Graph Modification
Social Network
Privacy
Database Security

چکیده

در سال‌های اخیر، نگرانی کاربران از انتشار داده‌های شبکه‌های اجتماعی با توجه به استفاده گسترده از این اطلاعات برای اهداف پژوهشی افزایش یافته است. در این مقاله به خطر افشای شناسه یک گره در شبکه با فرض اینکه مهاجم اطلاعات یکی از همسایه‌های بلافصل آن گره را در اختیار دارد، پرداخته می‌شود. ابتدا سطح بی‌نامی یک گراف شبکه بر اساس این خطر فرمول‌بندی شده و در ادامه یک مدل ریاضی برای حل این خطر ارائه می‌شود. استفاده از روش فوق بر روی تعدادی از مجموعه داده‌های مصنوعی و واقعی نشان‌دهنده عمومیت روش فوق است که می‌تواند در کاربردهای مختلف نتایج مناسبی را بر حسب سودمندی داده‌های بی‌نام‌سازی شده به‌دست آورد.

***doi**: 10.5829/ije.2018.31.10a.03*