Structural Anonymity For Privacy Protection In Social Network

Rana AL-Asbahi

Computer science and technology, Zhejiang university

DOI: 10.29322/IJSRP.11.06.2021.p11414
http://dx.doi.org/10.29322/IJSRP.11.06.2021.p11414

Abstract—the release of social networks in digital media has verified many benefits in our daily life, but the user privacy breach accompanies this intensification in popularity. Various methods has been used for publishing the social network data, the privacy preservation of the individuals in the data published has become a significant concern. Several works in relational data showed that the degree of privacy preservation does not depend on the size of the equivalence classes on quasi identifier attributes; it is determined by the number and distribution of distinct sensitive values associated with each equivalence class. To overcome the weakness in *k*-anonymity, we will use *l*-diversity to achieve guarantees more substantial privacy preservation *along with structural anonymity*. The proposed algorithms use minimum edge addition techniques. The empirical study shows that our algorithm leads significantly to less number of edge modifications for anonymization of the social network data and has an extensively lower running time than the other algorithms previously proposed in the field. We will use more than one dataset in our experiments to declare differential datasets lead to a different result; Every dataset has its properties, which should be analyzed to select the best Anonymization method.

Index Terms— Privacy , anonymization , *k*-anonymity ,*l*-diversity

1 Introduction

The Social network is a social structure made up of individuals groups, which have become increasingly popular applications in web 2.0. recent research has begun to study social networks to understand their structure[1, 2], advertising, and marketing[3] [4]. However, if individuals can be uniquely recognized in the released data, then their private information would be disclosed. Releasing the data without revealing their sensitive information consider as a significant problem many researchers have developed. To avoid the identification of records in released data, exceptionally by isolating information like names and the community security numbers are removed from the table. However, this method still does not guarantee the privacy of individuals in the data. With some local knowledge about individual vertices in a social network, an adversary may attack the privacy of some targets. a lot of studied provides a relative guarantee to prove that the identity of individuals cannot be discovered With applying the term of Anonymization.

In the latest years, a novel explanation of privacy called k-anonymity has increased popularity. In Privacy Preserving of Data Publishing, the progress of data processing is called data anonymization. *K*-anonymity is one model of anonymization approaches proposed by L.Sweeney [5, 6], it's a practical and straightforward privacy-preserving approach. The study declares that each record in the dataset cannot be distinguished with at least another (*k-1*) records under the prediction of quasi-identifiers of the dataset after a series of anonymity operations. This study has drawn considerable interest from the research community.

Therefore, no privacy-related information can be inferred from the k-anonymity protected table with high confidence, but the k-anonymity algorithm is reluctant to background knowledge and homogeneity attacks.

K-anonymity has been extensively studied in recent years [7,8,9,16]. It's assumed that the probability of uniquely representing an individual in the released dataset will not be greater than *1/k*. Therefore By using the k-anonymity, The adversary can breach the sensitive attribute in this network even if all the vertices in the network had a similar structure and the same degree. This limitation is fixed by another concept called *l* -diversity[23]. For the attack of attribute linkage, the adversary could understand sensitive information from the released dataset based on the distribution of sensitive values in the group that the individual belongs to. The standard effective solution for the attribute linkage attack is to lessen the correlation of quasi-identifiers and sensitive attributes of

the original dataset. Definitely, others models also bloom recently for capturing this kind of attack like recursive ℓ-diversity[10], (X,Y)-Anonymity[11], (a, *k*)-Anonymity[12], (*k*,e)-Anonymity[13], t-closeness by Li et al.[14], personalized privacy by Xiao and Tao[15] and so on.

The rest of this paper is organized as follows. In section2, we introduce the related work . In section 3 we present our proposed method of l-diverse anonymity based on clustering techniques. In section 4 we analyze the performance of our method through extensive experiment results. Section 5 contains the conclusions and future work.

2. Related Works

The research in social network privacy is very recent, and many evolutionary models have been proposed. However, most of them mainly based on adding edges or nodes in relational data. Some methods such as l -anonymity, and l-diversity have been developed for privacy preservation. But these cannot be applied to social network data directly. Anonymization of social network data is a much more challenging task than

anonymizing relational data in many ways. In recent years, several attack models such as [17], [18], [19], [20], [21] have been proposed in the study of privacy protection in publishing static networks. Liu and Terzi [21], in their research, proposed the degree attack to achieve the graph anonymization technique, in which an enemy utilizes the number of connections to a target individual to re-identify the target from a published network. Wu et al. [22] proposed a model to measure a common incident, where an attacker knows not only the vertex degree of a target individual but also the vertex degrees of the neighbors. The vertex degree consider as a basic feature in an important object in the social network, several subsequent studies have also adopted this category of attacks with some extensions. . Zhou and Pei [23] has studied the structural matching of attacks in the graphs which considered the one-hop neighborhood connections around a target individual as the knowledge of an attacker. Machanavajjhala et al. [24] studied l-diversities in three varieties. While Tripathy et al. [25] developed an algorithm with three phases that tackled the distinct l-diversity.

3. Proposed Method

3.1 PROBLEM DEFINITION

In this paper, we model social networks as graphs that are unweighted graph G=<V,E> where v is the set of nodes, E a set of edges for a given edge $e_{i,j} = (i, j) \subset v \times v$

In the published data, the vertex which has the highest degree is consider an important vertex that has a maximum connection so the attacker can identify this vertex easily, therefore by knowing the degree of connection for one vertex, the attacker will be able to breach the privacy even if the vertex had a similar structure to others vertex, this limitation is fixed by using the l-diversity which we will combine it with the K-anonymity.

Difination1: graph has k-anonymity if each node in it consider indistinguishable from at least k-l from others node.

3.1.1 Clustering Phase

In our method, the basic idea in this work is to cluster the graph G into a group that has a similar structure that satisfies specific metrics, each group size at least equal to K, and each vertex in the clusters should have l distinct values of the sensitive attribute to satisfy the l-diversity condition. The vertex which has high similarity are clustered into one group :

- arrange the vertex in the graph in the social network into descending order depend on vertex degree
- create a cluster of the size k, then put the vertex inside the cluster by taking care of each cluster that should have an l-distinct value to satisfy the l-diversity.
- If the cluster is full, create a new cluster and put the other vertex in till it gets the size k to create another cluster, each time when we put the vertex inside the cluster, we try to satisfy the l-distinct value the fig.1 down here declare the clustering process.

Figure 1 represents the clustering phase diagram
if all the clusters already created and there is a vertex from the same degree, and the same l-distinct value don't have a cluster yet, and we can't put it inside the clusters, we compute the degree cost between this node and all the clusters, the cluster which has the minimum cost' we put it inside.

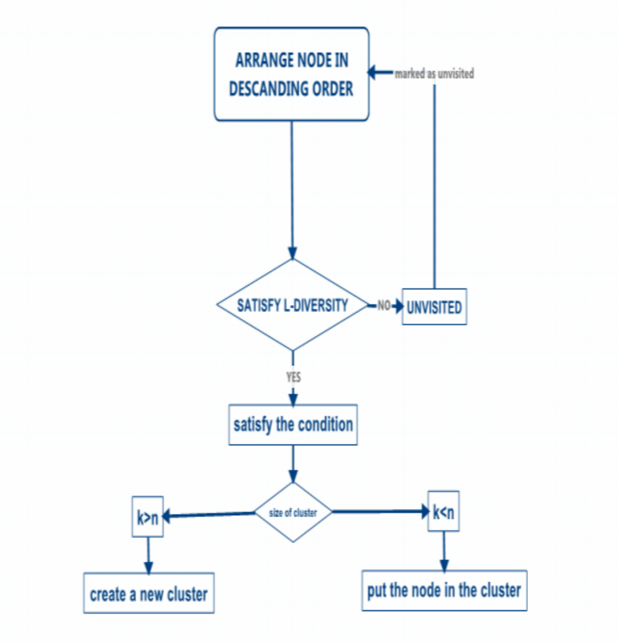


Fig.1 represent the clustering phase diagram

To measure the distance between the node and the clusters we use k-mean to compute the average degree for each cluster then compare all the clusters with the node .the cluster which have the minimum cost, and it is average degrees consider the nearest to the node degree we put this node inside the cluster, each time we add a node to the cluster, we should measure the average again for the cluster.

Algorithm 1 :

```
Data: A set T of n-nodes, the value k for k-anonymity, the value l for l-diversity, and the value d for d-neighborhood.
Let Ti be the ith node in the input graph.
Result: Clusters {C1,C2, . . . ,Cm};
Order {Ti} according to degrees in descending order. Store this in array Q;
Let cid (cluster id) = 0;
Mark visited [Ti] = 0 ∀ i = 1,2, ...,n;      Let t,j=0;
for t=1:n
   if  visited[t]==1
      continue;
   endif
   Add node t to cid
   Mark t as visited
   for j=1:n
      if visited[j]==1
         continue;
      endif
  let sv[ ] = sensitive values for all node in cid;
 if sensitive value of j in sv[ ]
         continue;
     else
         Add node j to cid;
         visited[j]==1;
         break;
     endif
   endfor
   cid=cid+1;
   if n/cid > k
      break;
   endif
endfor

numClusters=Total number of clusters;

for t=1:n
   if  visited[t]==1
      continue;
   endif
Costclusters= cost between t and ALL clusters ;
Order Costclusters in a descend order.
```

http://dx.doi.org/10.29322/IJSRP.11.06.2021.p11414

```
for c=1: numClusters
        let sv[ ] = sensitive values for all node in cluster c;
if sensitive value of t in NOT in sv[ ]
                Add node j to c;
      visited[t]==1;
       break;
endif
endfor
  if visited[t]==0
        Add node t to c WHERE c=1;
endif
endfor
```

3.1.2 Anonymization phase

There are many approaches to satisfy the degree requirements of nodes that are to be anonymized. Within this strategy, during the anonymization period, to satisfy degree requirements of nodes that are to be anonymized is through the addition of edge among them. In this manner, we're fulfilling their prerequisites with the addition of one edge between them is dependent on two nodes simultaneously they are not connected. For instance, consider two nodes X and Y so that they belong to various clusters and so are not connected previously. A benefit is added between them to satisfy the degree requirements of both nodes, and therefore, nodes are anonymized.

If a node's degree requirement hasn't been satisfied and there's no other suitable node to which it could be connected, then join that node to some node of the nearest Suitable group with odd variety of factors if the amount required is weird.

If there is a degree of node left unanonymized, and there is no cluster with a number of nodes need a degree to satisfy the requirement, create a fake node and Connect the remaining elements to this fake node

Algorithm 2 :

```
Input: Clusters {C1,C2, . . . ,Cm};
Output: Anonym Clusters {C1,C2, . . . ,Cm};

Let n =total number of nodes in the graph
neededDegreeCalculation:
Let degreeNeeded[ Ni] = 0 ∀ i = 1,2, ...,n
for j=1:n
        degreeNeeded[j]= the needed number of edges for node j in order to satisfy k anonymity;
endfor
      sort degreeNeeded[ ] in descending order;
EndOf neededDegreeCalculation

      for j=1:n
            NewNodes=Nodes in degreeNeeded WHERE there is no connection between them and j
            add edges between j and NewNodes;
            Mark j as visited
            Goto neededDegreeCalculation and re-calculate degreeNeeded[ ]
endfor
m=maxDegreeNeededAfterAddingEdges
Define F[Mf] be the fake nodes need to be added to the graph , where f=1:m
x= number of nodes still need edges
for j=1:x
      add edges between j and   F(1:degreeNeeded[ j]);
      Mark j as visited
endfor
```

4. EMPIRICAL EVALUATION

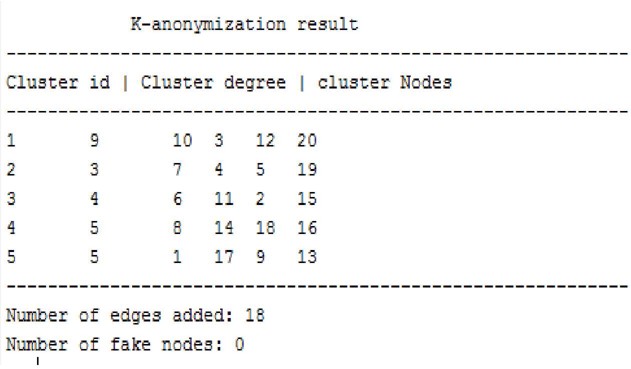
We report a systematic empirical study to evaluate our anonymization method using a small graph contain 20 nodes and real data sets. All experiments were conducted on a Personal computer running Microsoft Windows 7 operating system, with 3.40 GHz i3-3240 CPU intel® core(TM), 8.0 GB main memory and a 600 GB hard disk. The program was written in Matlab.

The data set was collected from the University of Oregon Route Views Project - Online data and reports. The dataset contains 733 daily instances which span an interval of 785 days from November 8 1997 to January 2 2000 ， The version used contained all components of the network, for a total of 6474 Nodes and 13895 Edges and an average

http://dx.doi.org/10.29322/IJSRP.11.06.2021.p11414

degree from 1 to 1500. we generated a list of 30 disease names and equally distributed them among the nodes. Including fewer edges is appealing in anonymizing a social network since the network structures can be protected better.

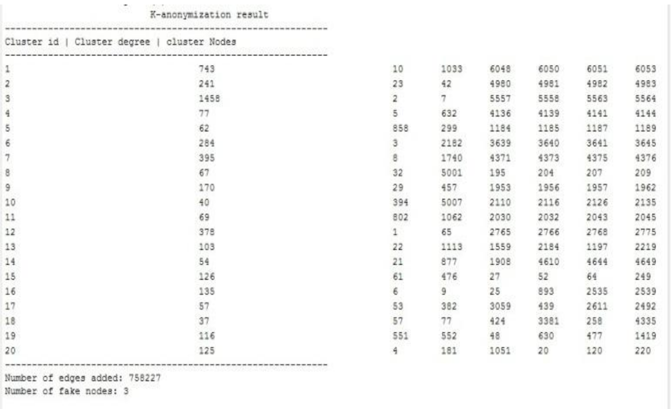
For the graph which contains 20 nodes, we get a good result illustrated in fig.2



```
            K-anonymization result
-----------------------------------------------------------
Cluster id | Cluster degree | cluster Nodes
-----------------------------------------------------------
1       9       10   3   12  20
2       3       7    4   5   19
3       4       6    11  2   15
4       5       8    14  18  16
5       5       1    17  9   13
-----------------------------------------------------------
Number of edges added: 18
Number of fake nodes: 0
```

Fig2: illustrate the approached algorithm

The results for the dataset are demonstrated in Fig. 3. The project for #nodes = 6474,k=20 and ℓ=1 executed in 5.34 sec. Obviously, the number of edges included increments with increment in estimations of k and ℓ and because of the huge average degree between the nodes.



```
                    K-anonymization result
-------------------------------------------------------------
Cluster id | Cluster degree | cluster Nodes
-------------------------------------------------------------
1                          743                 10    1033  6048  6050  6051  6053
2                          241                 23    42    4980  4981  4982  4983
3                          1458                2     7     5557  5558  5563  5564
4                          77                  5     632   4136  4139  4141  4144
5                          62                  858   299   1184  1185  1187  1189
6                          284                 3     2182  3639  3640  3641  3645
7                          395                 8     1740  4371  4373  4375  4376
8                          67                  32    5001  195   204   207   209
9                          170                 29    457   1953  1956  1957  1962
10                         40                  394   5007  2110  2116  2126  2135
11                         69                  802   1062  2030  2032  2043  2045
12                         378                 1     65    2765  2766  2768  2775
13                         103                 22    1113  1559  2184  1197  2219
14                         54                  21    877   1908  4610  4644  4649
15                         126                 61    476   27    52    64    249
16                         135                 6     9     25    893   2535  2539
17                         57                  53    382   3059  439   2611  2492
18                         37                  57    77    424   3381  258   4335
19                         116                 551   552   48    630   477   1419
20                         125                 4     181   1051  20    120   220
-------------------------------------------------------------
Number of edges added: 758227
Number of fake nodes: 3
```

Fig3: illustrate our approached algorithm with k=20

5. Conclusion and Future Work

Within this article, we discussed the privacy problems while publishing the social networks and methods to preserve them, and the challenges in Anonymization social networks. We introduced two algorithms for reaching Structural vulnerable and anonymity attribute-value safety in published social network data. one algorithm described the clustering phase the other algorithm describe the Anonymization phase . In our algorithm we tried to achieve the privacy preserving to solve the social network breach problems . We evaluated first our algorithm using a small graph denoted by an array contain 20 nodes represent disease names and two real datasets and we found that it required addition of significantly less number of edges. The maximum addition of the edge depends on which dataset we are using and the difference between the node degree. In future work, we need to compute the utility loss incurred due to addition of edges/fake nodes in the social network graph. We similarly expect to actualize our proposed idea of incomplete anonymity for d > 1 and test its data assurance versus the utility of information loss.

References:

[1] M Girvan, M.E.J.N., *Community structure in social and biological networks.* Proc. Natl. Acad. Sci. USA 99, 2002

[2] Y.-Y. Ahn, S.H., H. Kwak, S. Moon, and H. Jeong,, *Analysis of Topological Characteristics of Huge Online Social Networking Services.* ACM, 2007,: p. 835–844

[3] S. Hill, F.P., and C. Volinsky, , *Network-Based Marketing: Identifying Likely Adopters via Consumer Networks.* Statistical Science, 2006: p. 256–275.

[4] L. Getoor and C. Diehl, *Link mining: A survey,"* ACM SIGKDD Explorations Newsletter, 2005.

[5] L. Sweeney, *k-Anonymity*: *A model for protecting privacy, Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, **10**, 557-570 (2002).

[6] L. Sweeney, *Achieving k-anonymity privacy protection using generalization and suppression, Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, **10**, 571-588 (2002).

[7] Dewri R., k-anonymization in the presence of publisher preferences, Knowledge and Data Engineering, IEEE Transactions, **23**, 1678-1690 (2011).

[8] Nergiz M. E., Multirelational k-anonymity, Knowledge and Data Engineering, IEEE Transactions, **21**, 1104-1117 (2009).

[9] Jiuyong Li, Wong, R. C. W. Wai-chee Fu, A., Jian Pei, Transaction anonymization by local recoding in data with
attribute hierarchical taxonomies, Knowledge and Data Engineering, IEEE Transactions, **20**, 1181-1194 (2008).

[10]Ahmed Abdalaal, Mehmet Ercan Nergiz, Yucel Saygin, Privacy-preserving publishing of opinion polls, Computers & Security, 143-154 (2013).

[11] Ke Wang, Benjamin C. M. Fung, Anonymizing sequential releases, In Proceedings of the 12th ACM SIGKDD Conference, 414-423 (2006).

[12] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, KeWang, (a, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing, In Proceedings of the 12th ACM SIGKDD, 754-759 (2006).

[13] Qing Zhang, Koudas N., Srivastava D., Ting Yu, Aggregate query answering on anonymized tables, In Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE), 116-125 (2007).

[14] Ninghui Li, Tiancheng Li, Venkatasubramanian S., tcloseness: privacy beyond k-anonymity and l-diversity, In Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE), 106-115 (2007).

[15] Xiaokui Xiao, Yufei Tao, Personalized privacy preservation, In Proceedings of the ACM SIGMOD Conference, 229-240 (2006).

[16] Tamir Tassa, Arnon Mazza, k-Concealment: An Alternative Model of k-Type Anonymity, Transactions on Data Privacy, 189-222 (2013).

[17] J. Cheng, A.W.-C. Fu, and J. Liu, "K-Isomorphism: Privacy Preserving Network Publication against Structural Attacks," Proc.ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), 2010

[18] B. Zhou and J. Pei, "Preserving Privacy in Networks Against Neighborhood Attacks," Proc. IEEE 24th Int’l Conf. Data Eng. (ICDE), 2008.

[19] L. Zou, L. Chen, and M.T. Ozsu, "K-Automorphism: A General Framework for Privacy Preserving Network Publication," Proc. VLDB Endowment, vol. 2, pp. 946-957, 2009.

[20] C.-H. Tai, P.S. Yu, D.-N. Yang, and M.-S. Chen, "Privacy- Preserving Social Network Publication against Friendship Attacks," Proc. 17th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), 2011.

[21] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), 2008.

[22] C.-H. Tai, P.S. Yu, D.-N. Yang, and M.-S. Chen, "Privacy- Preserving Social Network Publication against Friendship Attacks," Proc. 17th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), 2011.

[23] B. Zhou and J. Pei, "Preserving Privacy in Networks Against Neighborhood Attacks," Proc. IEEE 24th Int’l Conf. Data Eng. (ICDE), 2008

[24] Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M.: l-diversity: Privacy beyond kanonymity, In Proc. 22nd Intl. Conf. Data Engg.. (ICDE), (2006),24.

[25] Tripathy, B.K.; Mitra, A., "An algorithm to achieve kanonymity and l-diversity anonymisation in social networks," Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on , vol., no., pp.126,131, 21-23 Nov. 2012.

http://dx.doi.org/10.29322/IJSRP.11.06.2021.p11414