k-Degree Anonymity Model for Social Network Data Publishing

Kamalkumar R. MACWAN, Sankita J. PATEL
Computer Engineering Department,
Sardar Vallabhbhai National Institute of Technology, Surat-395007, Gujarat, India
kamal.macwan@yahoo.com, sankitapatel@gmail.com

***Abstract*—Publicly accessible platform for social networking has gained special attraction because of its easy data sharing. Data generated on such social network is analyzed for various activities like marketing, social psychology, etc. This requires preservation of sensitive attributes before it becomes easily accessible. Simply removing the personal identities of the users before publishing data is not enough to maintain the privacy of the individuals. The structure of the social network data itself reveals much information regarding its users and their connections. To resolve this problem, k-degree anonymous method is adopted. It emphasizes on the modification of the graph to provide at least k number of nodes that contain the same degree. However, this approach is not efficient on a huge amount of social data and the modification of the original data fails to maintain data usefulness. In addition to this, the current anonymization approaches focus on a degree sequence-based graph model which leads to major modification of the graph topological properties. In this paper, we have proposed an improved k-degree anonymity model that retain the social network structural properties and also to provide privacy to the individuals. Utility measurement approach for community based graph model is used to verify the performance of the proposed technique.**

***Index Terms*—data privacy, data processing, publishing, social network services, utility programs.**

I. INTRODUCTION

Social networks have gained popularity among people nowadays. People are using these social network platforms to create their profiles, maintain their connections, browse news, etc. In order to use the features and the functionalities of the social networks, users provide a huge amount of data that is maintained by social network service providers. These collected data from social network services is called as social network data and it is much useful in the field of marketing and survey. Data owner often makes this social network data available to third parties for specific analysis or for advertisement purpose. But, publishing such information may cause some serious privacy threat.

As social network data contains private information of the individuals and also their sensitive relationship information, publishing social data have a risk of privacy disclosure [1]. A privacy breach occurs when the private and confidential information of the individual is disclosed to an adversary. The social network data should be anonymized to maintain user privacy before releasing it for analysis purpose.

There are different anonymization approaches: anonymization via clustering, graph modification approach, and a hybrid approach [2]. The resultant anonymized dataset have different graph structure properties. So, a proper balance should be made between the data usefulness and the anonymization level. In this paper, we address this problem and present an approach that provides privacy to user identity and it maintains structural properties too.

A. Privacy Breaches

An adversary may try to know sensitive information of some victims using the published dataset along with some background knowledge. Depending upon the knowledge that an adversary uses to disclose the sensitive information of the victim, social network attacks can be categorized into three types: identity disclosure, sensitive link disclosure, sensitive attribute disclosure [3, 4]. Identity disclosure occurs when an individual behind a record is exposed whereas sensitive link disclosure occurs when connections between two individuals are revealed. Sensitive attribute disclosure results when an adversary obtains the sensitive user attribute. It is not possible to resolve all the three privacy attacks by using only one privacy preservation technique.

Among all these privacy problems, identity disclosure attack [1] is of most concern. Number of relations of the target user can be easily known to the third party. Generally, social network is represented as an undirected graph and it reveals much useful information regarding vertices (users), such as vertex degree (number of connections), neighborhood structure, mutual relations, etc. Adversary can use information regarding number of connections as background knowledge and can try to de-anonymize the nodes.

B. Related Work

Published dataset should be properly anonymized to maintain user and data privacy. Anonymization approaches are mainly classified in two categories based on the generalization and perturbation. **1. Clustering based approach:** In this method [5, 6], the vertices and the edges are formed into groups and anonymize a subgraph into a super-vertex. Each super-vertex represents number of vertices and number of edges that it contains. Individual information can be hidden properly by this method but it fails to provide an accurate analysis of the user behavior. **2. Graph modification approach:** Graph anonymization can be done by modifying the edges and the vertices in a graph. There are three sub-categories of this anonymization method. First, the optimization approach estimates the optimal anonymized dataset and modifies the original

Digital Object Identifier 10.4316/AECE.2017.04014

dataset accordingly [1]. Second, the randomization approach modifies graph structure by randomly adding/deleting edges or switching edges [7, 8]. It protects against re-identification in a probabilistic manner. Last, the greedy graph modification approach tries to optimize data utility objectives to fulfill the privacy preservation requirement by greedily anonymization operations.

The method of just removing identifier attributes of the node is not enough to provide user privacy [9]. The uniqueness of some nodes in small embedded subgraph in a network can infer the privacy of users. Adversary can easily map targeted users with vertices of the published graph if they are unique in terms of their degree. A practical solution to defend against the identity disclosure attack is the k-anonymity. It states that the degree of each vertex is identical to at least k-1 other vertices [10]. Although an adversary has some background information of a user, it can be mapped to multiple identities of k-anonymized dataset. If an adversary has background knowledge of a target victim and also the relationship between its neighbors, the victim's identity may be revealed even though the vertex identity is preserved using the anonymization method. Zhou and Pei [2] proposed k-neighborhood anonymity approach to have at least k nodes that have same neighborhood structure. Adversary can use the subgraph information around a certain individual as the background knowledge to re-identify targeted user. Zou [11] proposed K-Automorphism method to provide security against the attacker having knowledge about the degree, subgraph and the neighbor of the target node. Moreover, the privacy preserving data mining approach using fuzzy logic, neural networks[12 - 15] can also be applied for anonymization in data publishing.

In this paper, we focus on k-anonymization method. It can be achieved by graph modification operations. Moreover, the k-anonymity model aims to sanitize the original graph, resulting into a compromise of the data utility. Therefore, in social network data publishing, the tradeoff between the individual's privacy and the data utility has become a major concern. Wang and Xie [16] proposed *k*-anonymization approach that performs edge shift and deletion operations. Anonymized dataset in this approach may lose some useful relationship information between the two vertices due to edge deletion operation. Moreover, cluster formation is same irrespective of the size of dataset and anonymization parameter *k*. Liu and Li implemented an anonymization algorithm that contains edge and vertex addition operations for graph modification [3]. Vertex addition operation adds too many vertices that deviates from the aggregate result and changes the graph properties.

C. Contribution and Organization

Existing anonymization algorithms work directly on the degree sequence and perform smallest degree change to achieve high utility [1, 3]. But, it fails to preserve the graph properties. It is proved that the change in community structure also reflects in graph topological properties [17] such as average betweenness (BW), clustering coefficient (CC) and average path length (APL). So, we propose an approach to build different communities (sub-graph of original graph) to perform edge addition operations within that community. Utility measurement approach based on community detection is used to find out usefulness of the data. The main contributions of this work are listed below:

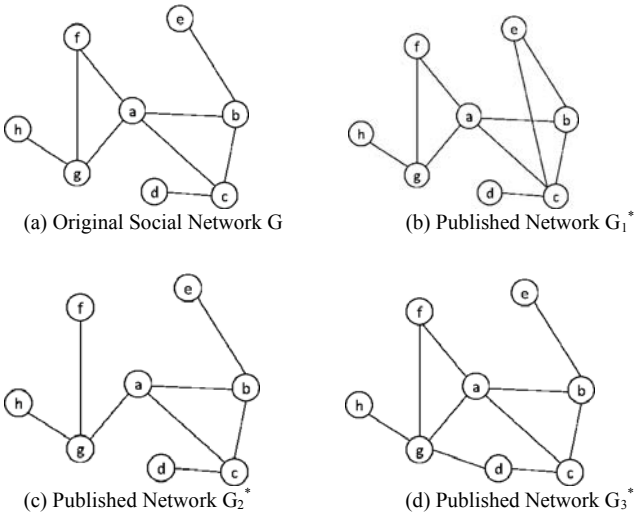
- We propose a new clustering approach that makes partition of the entire dataset into clusters based on their connectivity. The partition parameter used for this clustering technique depends on anonymization parameter *k*, resulting into compatible partition of the dataset. This clustering technique can be applied on continuous live social data too.
- We design a framework for k-anonymization method that preserves the graph properties too.
- We conduct experiments that show improvement on some graph properties compared to the existing anonymization.

The rest of the paper is organized as follows. Section 2 explains different possible options for anonymizing social network graph. Section 3 describes suitable clustering approach for social dataset. Anonymized graph construction operations are briefly discussed in section 4. Section 5 presents privacy analysis and utility measurement approach. Section 6 experimentally evaluates our proposed approach and section 7 concludes the paper with future directions.

II. PROBLEM DEFINITION

G is a finite set that represents individual or social entities, V represents user or entity and E represents binary relationship between them. For a social graph having n users (vertices), a tuple $d_G = (d_1, d_2, d_3, \ldots\ldots, d_n)$ where d_i represents the degree of vertex V_i. According to definition in [10], graph is k-anonymized if every value in tuple d_G repeats for at least k-times and such degree sequence is called as anonymized degree sequence. The goal of privacy preservation is to convert the original degree sequence into anonymized sequence with least number of graph modification operations.

There are multiple options to achieve the desired target degree. A social network G is shown in fig 1(a), three different published graphs satisfying 2-degree anonymity are formed as G_1^*(fig 1(b)), G_2^*(fig 1(c)), G_3^*(fig 1(d)). Based on the common measurement, all three anonymized graphs have same utility loss. However, they are very different in terms of graph structure properties such as APL, BW and CC as illustrated in table 1.



(a) Original Social Network G (b) Published Network G_1^*

(c) Published Network G_2^* (d) Published Network G_3^*

Figure 1. Example of 2-degree anonymization by edge insertion & deletion

Definition 1: (Average Path Length) It is calculated as the average number of nodes for the shortest path between

all possible pairs of network nodes.

Definition 2: (Average Betweenness) It is defined as the average of betweenness centrality of all nodes. It is calculated as number of shortest paths from all nodes to all other nodes that pass through that node.

Definition 3: (Clustering Coefficient) It is a measure of the degree to which nodes in a graph can form cluster.

Definition 4: (Degree Sequence (DS)) Degree Sequence is a vector that represents the degree of all vertices of a given graph. For a graph G (V, E), DS is a vector of size |V| and DS[i] represents the degree of vertex V_i.

TABLE I. PROPERTIES OF THE ORIGINAL AND PUBLISHED SOCIAL DATA

Graph	DS	ΔDS	APL	BW	CC
G	4 3 3 3 2 1 1 1	N.A.	2.07	3.75	0.47
G_1^*	4 4 3 3 2 2 1 1	2	2	3.50	0.60
G_2^*	4 4 3 3 2 2 1 1	2	1.89	3.12	0.35
G_3^*	3 3 3 3 1 1 1 1	2	2.25	4.37	0.25

Among all three 2-anonymized graph, G_1^* is the most similar to original social graph G compared to G_2^* and G_3^*. Different options are available for edge insertion for same DS requirement. Edge inserted between the two distant vertices brings more deviations in the graph structural properties. Selection of the vertices pair for edge insertion plays a crucial role here. So, in our approach, we choose the vertices having distance of just a few hops for edge insertion operation. Based on this assumption, the proposed anonymization process for social data is presented in fig 2.

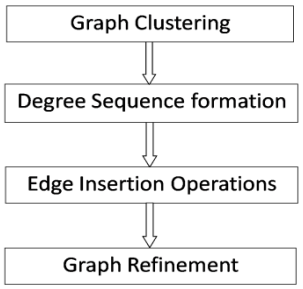


Figure 2. Flow of the work

III. GRAPH CLUSTERING

The goal of this step is to partition the vertices of a large graph into different clusters based on vertex connectivity. This results into different densely connected sub-graphs which has more number of edges within the cluster and fewer edges between the vertices from different clusters. Many existing graph clustering approaches like normalized cut [18], modularity [19], and structural density [5] focuses on topological structure of a graph to achieve a cohesive internal structure in each partition.

Existing clustering algorithms based on modularity function has a drawback that it may create weakly connected very dense communities [17]. Here, we want to make a cluster of vertices having high connectivity among them. Our aim for graph clustering operation is to partition the entire graph into different clusters that are to be used in the anonymization operation. This clustering approach is also useful to measure the data utility. In our proposed graph clustering approach, we start cluster formation with vertices having higher degree as cluster agents and add other vertices based on their connectivity with the cluster. The number of clusters agent (α) depends on the number of vertices and the anonymization parameter *k* and is defined as:

$$Number\ of\ cluster\ agent(\alpha) = \frac{Number\ of\ vertices}{2k-1} \quad (1)$$

Algorithm 1 Clustering Algorithm

```
Score each node based on its connectivity;
visited[] = false;
for each cluster C_i do
        V_i^1 = 1-Neighborhood of cluster C_i
        if score(v ε V_i^1) == 1 or 2 then
                add vertex v to C_i;
                visited(v) = true;
        end if
end for
for each cluster C_i do
   V_i^1= 1-Neighborhood of cluster C_i & visited(v) =false;
    if (No. of edges between C_i and v) > (score(v)/2)) then
        add vertex v to C_i;
        visited(v) = true;
   end if
end for
Repeat above step until no further changes in cluster C_i
for each unvisited node V_x do
        add V_x to cluster C_i
        where V_x ∩ C_i = maximum connectivity
end for
```

Algorithm 1 exemplifies the steps to partition the entire graph into different clusters. Here, the cluster formation plays a crucial role. Edge insertion operation performed within the same cluster does not lead to significant deviation in the graph properties. So, based on the anonymization parameter *k*, cluster size should be formed. First N numbers of nodes having higher degree are appointed as cluster-agent. Initially, cluster set is initialized with cluster-agent. The nodes directly connected with cluster-agent nodes and having degree 1 or 2 will be placed in that particular cluster-set. 1-Neighborhood of each nodes of cluster-set will be checked repeatedly and if the number of edges between that node and cluster exceeds the maximum limit of its connectivity to any cluster then that node can be directly placed into that cluster-set. At the end, based on connectivity, unvisited nodes will be covered for its proper placement in cluster-set.

Appointing vertices as a cluster agent can be processed in $O(\alpha)$ time. It takes $O(V+E)$ time to allocate all the vertices into their respective cluster-agent set. Figure 3 shows the result of algorithm 1 for a given social network G.

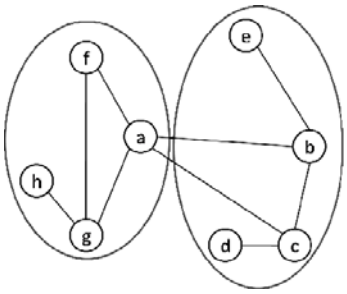


Figure 3. Clustering of original social network G

IV. Anonymized Graph Construction

In this section, we discuss a procedure to build the *k*-degree anonymous graph G'(V', E'). Here, we consider degree sequence is in the descending order (i.e. DS[1] ≥ DS[2] ≥ DS[3] ≥..... ≥ DS[|V|). The graph construction procedure has two steps. Initially, it divides the entire degree sequence into different groups that meet the k-anonymity requirement and construct anonymized degree sequence $d_{G'}$. Then, edge insertion operations are performed to construct anonymous graph.

A. Degree Sequence Anonymization

As there should be at least *k* number of nodes having same degree, it is required to change the degree of some nodes. We apply graph-modification operations to the edges only and keep the same number of vertices.

To perform effective partition of degree sequence for k-anonymized operation, maximum degree difference between neighbors is a crucial parameter for the degree sequence partition [3]. As per the observation in Liu and Terzi[1], any k-anonymous group have maximum size of 2*k*; for any group of size larger then 2*k*, it can be further divided into two subgroups. Divide-and-conquer approach [3] for degree sequence partition is used to implement the above concept.

For each degree sequence partition, the degree of the first vertex will be considered as target degree for the entire partition. The degree sequence difference DS^* (for the same partition of G^*) is calculated as the difference between the target degree (DS') and the original degree (DS) of vertices. Resulted DS' must be k-anonymized and the difference between DS and DS' should be minimized. For example, the social network described in fig 1(a), a degree sequence X = (4,3,3,3,2,1,1,1) contains 8 degree of vertices. For the anonymous parameter k = 2, according to the maximum neighbor difference the optimal anonymization partition is X' = (<4,3>, <3,3>, <2,1>, <1,1>) resulting into DS' = (4,4,3,3,2,2,1,1). However, there might be a need of modification in the degree sequence to meet anonymization requirement.

B. Generating candidate edge set operations

Candidate edge set operations that convert the original graph into anonymized graph satisfying the given anonymization parameter *k* should be performed based on the representation of DS^*. Here, we consider only the edge insertion operation. The goal of this edge modification operation is to match original degree sequence DS to anonymized degree sequence DS'. The edge insertion operation *insert (V[i], V[j])* is to insert a new edge that links vertex V[i] to vertex V[j]. Each element in DS^*[i] indicates necessary edge addition operation. A vertex in G (V, E) having degree DS[i] needs to increase up to DS'[i].

For given social network in fig 1(a), DS=(4,3,3,3,2,1,1,1) and DS'=(4,4,3,3,2,2,1,1) is 2-anoymized estimated degree sequence, we derive degree difference sequence as DS^*=(0,1,0,0,0,1,0,0). Vertex having degree 3 and 1 are supposed to be increased. So, all the vertices having degree 3 and 1 are inserted into VS^+ set. VS^+={{g, b, c},{h, e, d}} contains those vertices that need insertion operation to perform among them to increase their degree. The vertices having same degree are placed together as a mutual exclusive set. Algorithm 2 shows steps for edge insertion among candidate nodes found in DS^*. To keep the structural properties of the original social network dataset, first we perform edge insertion operation within the cluster and then in between the different clusters. As edge insertion algorithm checks for suitable vertices pair selection, in worst case scenario it takes $O(V^2)$ time.

Algorithm 2 Edge Insertion Algorithm

```
DS* = DS' - DS
for each i ε DS* and DS*[i] > 0 do
    j = i+1;
    while DS*[i] > 0 and j < |V|
      if C[V[i]]==C[V[j]] & edge(V[i],V[j])=false} then
            add edge(V[i],V[j])
            decrease DS[i] and DS[j]
      end if
      end while
end for
```

C. Graph Refinement

Edge insertion operations are performed to modify the existing graph towards DS'. In some cases, it is not possible to achieve the estimated degree sequence DS' for k-anonymized graph by merely performing the above operations. If this happens, it is required to make some changes in DS' and to have one more attempt for k-anonymized graph. The additive adjustment on DS' is considered to achieve updated degree sequence that is close to the old DS'. Vertices from DS' that require further modification will be put in the set DS".

Algorithm 3 Graph Refinement Algorithm

```
while DS" ≠ empty do
      V[i] = smallest degree vertex from V
      if partition_count(V'[i]) > k) then
          for each vertex j ε DS" do
            if edge(V[i], V[j]) = false then
              count++;
              flag[] = j
            end if
          end for
      end if
      index = nearest k-partition value less than count
      for j=0 to index do
        add edge(V[i], V[flag[j]])
        remove flag[j] from DS"
      end for
end while
```

In order to achieve k-anonymized degree sequence, an adjustment in DS' is performed. The procedure to modify the degree of all vertices will start with the smallest degree vertex and will go to the higher ones until k-anonymization requirement is satisfied. For each anonymized vertex, if that vertex partition already has more than k vertices, then it can be shifted to another partition. Algorithm 3 shows the steps for the graph refinement procedure. It takes $O(\eta)$ times, where $\eta = \Sigma$ DS"[i].

D. Anonymization Cost Analysis

Anonymized graph G' is constructed from G by adding minimal edges. Several cost anonymization models are mentioned by Liu and Li [3]. Anonymization Graph Cost

(AGC) depends on the number of added edges and vertices, is defined as:

$$AGC = EA + VA \tag{2}$$

where *EA* represent the number of edges added to the original dataset to construct anonymized graph G'. As we increase the numbers of edges, the degree of vertices will also increase. Partition cost (PC) parameter calculates the increment in the vertex degree. It is the sum of difference of degree for all vertices from original dataset and anonymized dataset as defined as:

$$PC = \sum_{i=1}^{n} diff(x_i) \tag{3}$$

Using these two parameters, Graph Construction Ratio (CR) is defined as:

$$CR = \frac{PC}{AGC} \tag{4}$$

Lemma 1: For any value of *k*, Graph Construction Ratio for anonymized dataset, CR=2.

Proof: Graph modification method performs edge insertion and/or deletion operation and vertex addition operation. In our proposed anonymization approach, we have considered edge insertion operations only. Anonymized graph construction operation ensures to achieve the anonymization requirement for each vertex without adding new vertex into the anonymized dataset. So, for our approach, AGC depends on the number of inserted edges only. Single edge insertion operation increases the degree of two vertices. If the value of total number of inserted edges is *x*, then the total degree difference between vertices of original and anonymized dataset will be 2*x*. Thus, irrespective of anonymization parameter and data size, the value of CR is calculated as:

$$CR = \frac{PC}{AGC} = \frac{PC}{EA + 0} = \frac{2x}{x} = 2 \tag{5}$$

E. Summary

The entire anonymization process is divided into different operations as shown in fig 2. As a preprocessing step, algorithm 1 divides the entire graph into different clusters. Edge insertion operation shown in algorithm 2 is performed within the cluster and between the clusters in order to achieve the decided target degree for each vertex. Finally, graph refinement step shown in algorithm 3 is performed to increase the vertices degree which failed to reach their target degree by edge insertion operation.

V. Privacy and Utility Measurement

Data owner's privacy is inferred when an adversary can successfully map target victim to any vertex from published dataset with the use of some background knowledge. Here, we evaluate the privacy of proposed k-degree anonymity scheme against the minimality attack and community based utility loss measurement method to calculate usefulness of anonymized social data.

A. Privacy Analysis

To preserve the data utility, k-anonymity model performs anonymization operation that should minimize the distortion to the original data. The minimality attack is very familiar attack on the k-anonymized social dataset and was first discussed in [20].

In order to apply the minimality attack, the adversary uses information from the published dataset to reveal the sensitive information about the user. In worst case situation, adversary may know the identity and the number of connections for every user of the published dataset. Table II represents the adversary's worst background knowledge of the social network shown in fig 1(a). With background knowledge of the degree of vertices, adversary tries to map the real identities to the vertex IDs of the published graph shown in fig 1(b). Adversary predicts that there must be one user (vertex) that have degree 3 and one user with degree 1 whose degrees are increased. Therefore, the adversary infers that user Jack with degree 4 must be mapped to all vertices having degree 4 in anonymized published dataset. Since there are two vertices with degree 4, the probability of Jack mapped to any of them is $1/2$. Now, user Bob having degree 3 can be mapped to either a vertex with degree 4 or 3. Three users with degree 3 are present in original social network dataset and two users with degree 4, the probability of Bob mapped to vertex 'c' or 'a' is $\frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$. Similarly, the mapping of every individual is shown in table III. For 2-degree anonymized social dataset, the mapping probabilities derived by the adversary will not exceed $1/2$.

TABLE II. ATTACKER'S WORST-CASE BACKGROUND KNOWLEDGE

User Identity	Jack	Bob	Jim	Joel	Anne	Alice	Tom	Harry
Degree	4	3	3	3	2	1	1	1

TABLE III. ATTACKER'S INFERENCE PROBABILITY

User Identity	**Mapped Vertex ID**	**Probability**
Jack	a , b	$1 \times \frac{1}{2} = \frac{1}{2}$
Bob, Jim, Joel	a , b	$\frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$
	c , g	$\frac{2}{3} \times \frac{1}{2} = \frac{1}{3}$
Alice, Tom, Grace	e , h	$\frac{2}{3} \times \frac{1}{2} = \frac{1}{3}$

B. Utility Measurement

Anonymization algorithm that has less distortion in degree sequence contains much useful data. In this work, community based graph model is used for utility loss measurement [16]. Here, we consider fixed association of edges in whole graph. The impact of an edge insertion process is highlighted by the change in edge distribution within or between clusters. Here, we consider community based model to measure utility loss based on edge distribution. We already have different clusters of graph as a result of Graph clustering operation. This cluster set will be used for utility measurement too. We assume that the given graph G (*V, E*) is divided into m disjoint clusters, denoted as $C_G = \{C_1, C_2, ..., C_m\}$, such that $\forall v \in V$, there is only one partition containing *v*. Given social network G in fig 1(a) is divided into 2 communities as shown in fig 3. Cluster C_1 contains vertices {f, a, g, h} and C_2 contains vertices {b, c, d, e}.

Let ES_G and ES_{G^*} are corresponding edge distribution

sequences of the original social graph G and its anonymized result G^*. Utility loss induced by G^* compared to G, $UL(G,G^*)$ is defined as:

$$UL(G,G^*) = \| ES_G - ES_{G^*} \|_1 = \sum_{i=1}^{m} |ES_G[i] - ES_{G^*}[i]| \quad (6)$$

This utility measurement is based on the distance between ES_G and ES_{G^*}. Edge distribution ES_G represent the number of edges within the same cluster as well as between the clusters. Utility loss for three different 2-anonymized published dataset (depicted in fig 1) is shown in Table IV. Anonymized graph G_1^* causes the smallest utility loss compared to G_2^* and G_3^*.

TABLE IV. EDGE DISTRIBUTION AND UTILITY LOSS

Anonymized Graph	$ES_G = \left\langle \frac{n_{11}}{\|E\|} + \frac{n_{12}}{\|E\|} + \frac{n_{22}}{\|E\|} \right\rangle$	UL (G, G^*)
G	$\left\langle \frac{4}{9} + \frac{2}{9} + \frac{3}{9} \right\rangle$	N.A.
G_1^*	$\left\langle \frac{4}{10} + \frac{2}{10} + \frac{4}{10} \right\rangle$	0.1333
G_2^*	$\left\langle \frac{4}{10} + \frac{3}{10} + \frac{3}{10} \right\rangle$	0.1555
G_3^*	$\left\langle \frac{3}{8} + \frac{2}{8} + \frac{3}{8} \right\rangle$	0.1388

VI. EXPERIMENTS

The objective of our experiments is to demonstrate the effectiveness of our proposed approach in terms of different graph properties. The proposed anonymization algorithms presented in section IV are implemented in Java programming language to evaluate the performance on different datasets. Utility loss metrics explained in section V is used to calculate data utility of anonymized dataset. The experiments are conducted on an Intel Core, 2 Quad CPU, 3.20 GHz machine with 4GB RAM running Windows 7 OS to. We have used *Networkx* package to calculate graph topological properties in python.

A. Dataset

We have used four different real world datasets.

- **Dolphin's network:** It was constructed from observations of a bottleness dolphin community [4]. It contains 62 vertices and 159 edges in the network. An edge between vertices (dolphins) represents associations between dolphin pairs occurring more times.
- **DBLP:** The DBLP (Digital Bibliography and Library Project) computer science bibliography provides a list of research papers in computer science. It is constructed based on a co-authorship relationship. It contains 317080 nodes and 1049866 edges. This dataset is available at http://dblp.uni-trier.de/xml.
- **Powergrid Graph:** In this dataset, generators, transformers and substations in a powergrid network are represented by nodes. The high voltage transmission lines between them are represented by edge. It is available at http://www.cs.helsinki.fi/u/tsaparas/MACN2006/
- **GRQC:** The GRQC dataset (General Relativity and Quantum Cosmology) is a network that represents collaborations between authors papers submitted under this category. This dataset is available at http://snap/stanford.edy/data/ca-GrQc.html.

B. Experimental Evaluation

We have used snowball algorithm [21] to take portion of the graph as experimental dataset that is consistent with original graph in terms of general structural properties. As privacy is related to individual user, probability of inferring user privacy cannot be expressed as aggregate result for the entire dataset. It varies from user to user, but its value will not exceed $1/k$ irrespective of anonymization parameter k and the size of dataset.

1. Utility Loss versus graph size: In our first experiment, we calculate the utility loss for various size of graph for k=10. First we observe the change in different structural properties of graphs. Change in APL, CC and BW graph properties for our proposed work is shown in fig 4.

We observe that, anonymized results for dataset containing vertices more than 400 has minor deviation in graph properties(fig 4.1(a), 4.1(b), 4.1(c)). Here, anonymization parameter k is 10. So, more number of edge insertion operations is required for dataset containing 100 or 200 vertices than the others. The change in the ratio generated for them by our method is in the range of (1.3%, 7.3%) for CC, (9%, 35%) for APL and (10%, 41%) for BW. As the value of k is 5% to 10% of total number of vertices in the dataset, it provides higher privacy at the cost of major deviation in structural properties. Utility loss is also shown in fig 4(d) for various sizes of datasets.

2. Utility Loss versus graph size: The effect of privacy requirement k on the graph properties is evaluated in this section with experimental results. The graph-size is fixed to 500 vertices for DBLP dataset. For different values of k, performance of proposed algorithm is shown in fig 5. More edge modification operations are required for larger k value that results into more utility loss (fig 5(d)) and more deviation in graph properties (fig 5(a), 5(b), 5(c)).

C. Discussion

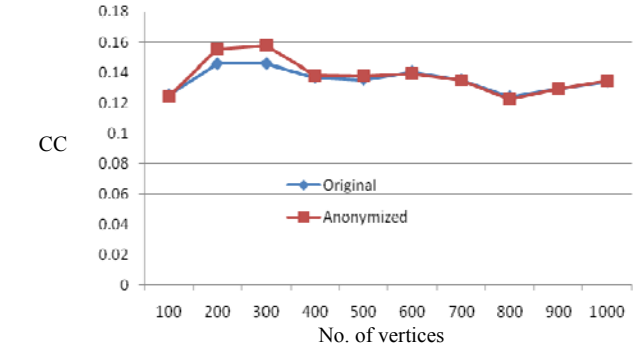
Experiments are performed on different real world datasets to verify the effectiveness of the proposed work. Our experiments prove that k-anonymized dataset also maintains the structural properties of the social network. Edge modification operation has a huge impact on data utility.

Kun and Terzi [1] proposed anonymization methods and presented results for various graph properties for powergrid dataset. Experimental results shown in fig 6 represent that our proposed approach has better performance compared to greedy-swap and priority algorithm [1]. Our approach achieves better results as compared to Peng and Li [3] for GRQC dataset as depicted in Fig 7.

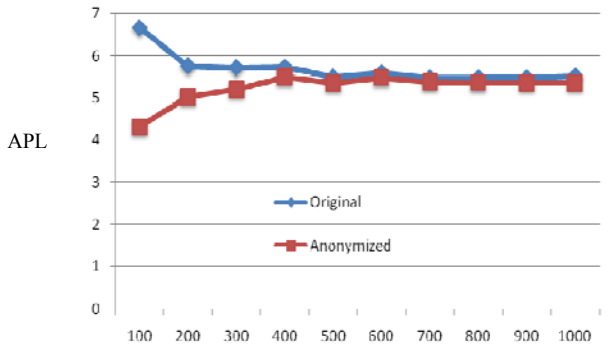
For the anonymized degree sequence, target degree can be selected as average degree of all nodes or it can be the degree of first node. First option contains edge insertion/ deletion/ shift operation while second option contains only edge insertion operation. Here, we have compared both the options for real-world dataset Dolphin. Fig 8 shows that, utility loss for only edge insertion operation is less compared to the edge shift/deletion. So, our approach meets the requirement of less utility loss by performing only edge insertion operation.

Edge deletion operation generates anonymized dataset

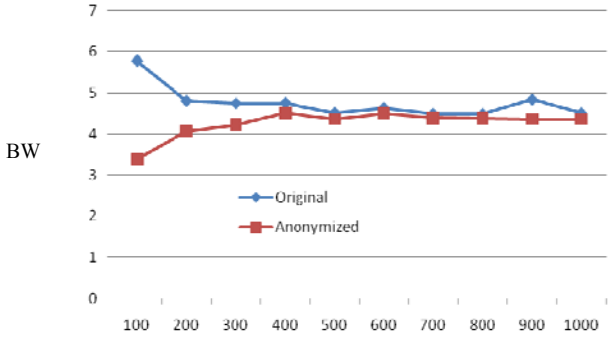
that may lose some important data [16]. As our approach does not contain any edge deletion operation, original data remains in anonymized dataset. In contrast to Peng and Li's work [3], our approach does not contain vertex addition operation. Numbers of vertices (users) in anonymized dataset are same as original dataset. It is very useful for aggregate queries in statistical analysis.



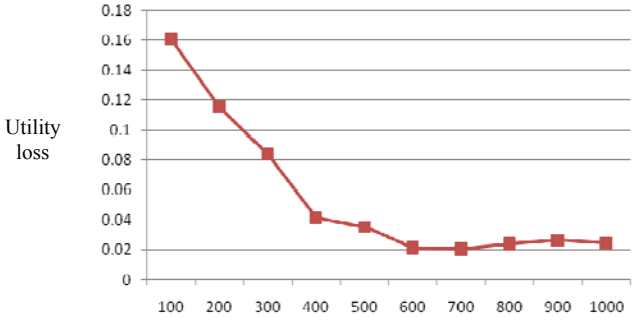
(a) CC



(b) APL

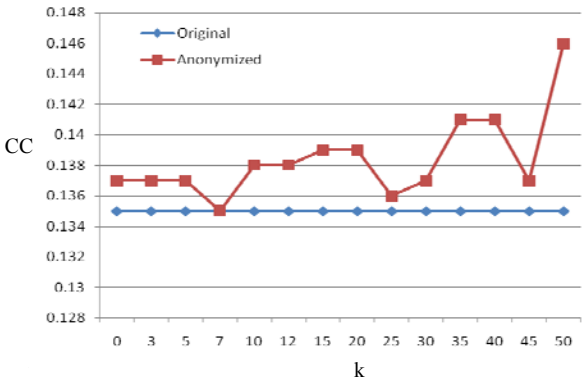


(c) BW

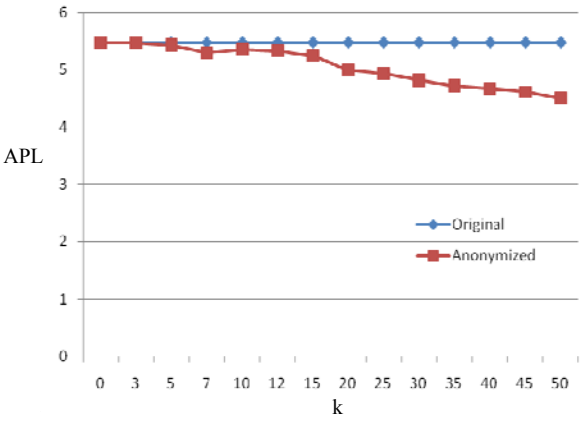


(d) Utility Loss

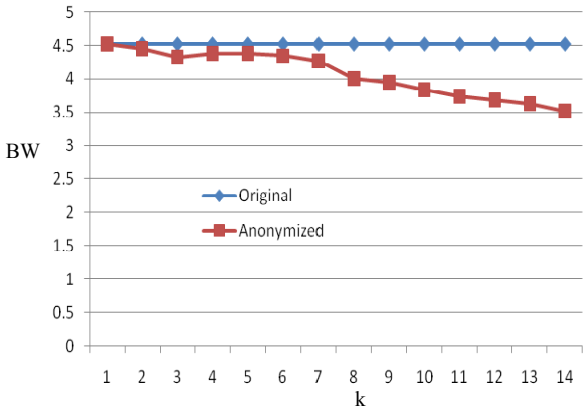
Figure 4. Properties of anonymized graph versus graph size on dblp graph (k=10)



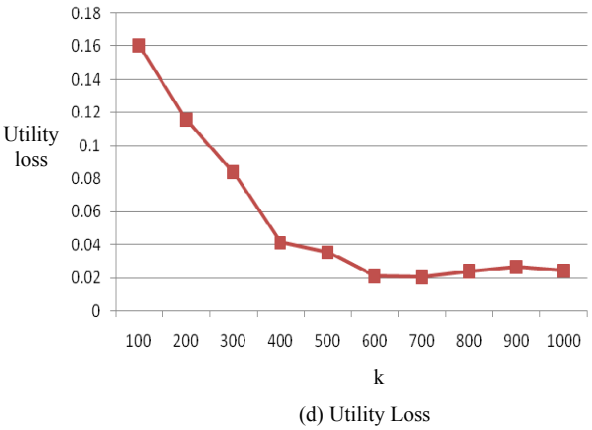
(a) CC



(b) APL



(c) BW



(d) Utility Loss

Figure 5. Properties of anonymized graph versus k on dblp graph (|v|=500)

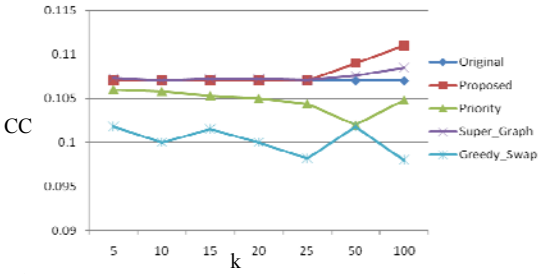
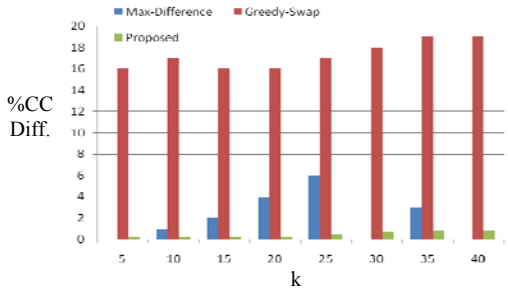
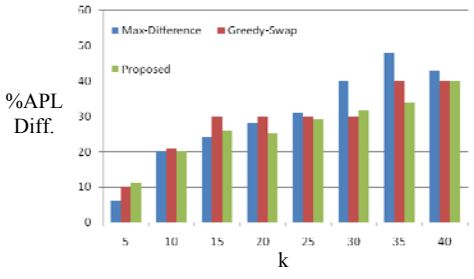


Figure 6. Comparison of CC result for various anonymization methods



(a) CC



(b) APL

Figure 7. Comparison of graph properties with Greedy-swap and Max-Difference methods

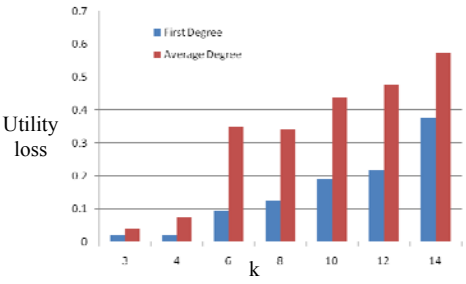


Figure 8. Comparison of utility loss for different target degree

VII. Conclusion

The privacy of users and utility are the main concerns for a good anonymization algorithm. Here, we propose an improved k-degree anonymity model that provides privacy with low utility loss. Edge insertion operation within same cluster has smallest deviation in graph properties. Resultant anonymized dataset have just approximately 4% change in CC. Resultant dataset does not contain additional vertices. Partitions of the entire network generated by the proposed algorithm give appropriate problem space for edge selection operation. The experimental results show that our approach works for different value of *k* and has less utility loss. Our approach can be extended for k-neighborhood privacy model and for k-automorphism model too.

References

[1] K. Liu and E. Terzi, "Towards identity anonymization on graphs." In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, vol. 10, no. 2, pp. 93-106. ACM, 2008. doi: 10.1145/1376616.1376629

[2] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," ACM Sigkdd Explorations Newsletter, vol. 10, no. 2, pp. 12- 22, 2008. doi: 10.1145/ 1540276.1540279.

[3] P. Liu and X. Li, "An improved privacy preserving algorithm for publishing social network data," in Proc. 10th Int. Conf. High Perform. Comput. Commun., pp. 888–895, 2013. doi : 10.1109/HPCC.and.EUC.2013.127

[4] D. Lusseau, "The emergent properties of a dolphin social network." Proceedings of the Royal Society of London B: Biological Sciences, vol. 270, no. 2, 2003. doi:10.1098/rsbl.2003.0057

[5] Tian, Yuanyuan, Richard A. Hankins, and Jignesh M. Patel. "Efficient aggregation for graph summarization." In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, vol. 37, no. 2, pp. 567-580. ACM, 2008. doi : 10.1145/1376616.1376675

[6] Campan A., Truta T.M., "Data and Structural k-Anonymity in Social Networks." In: Bonchi F., Ferrari E., Jiang W., Malin B. (eds) Privacy, Security, and Trust in KDD. Lecture Notes in Computer Science, vol. 545, pp 33-54, . Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-01718-6_4

[7] Bonchi, F., Gionis, A. and Tassa, T. "Identity obfuscation in graphs through the information theoretic lens." Information Sciences, vol. 275, pp.232-256, 2014. doi : 10.1109/ICDE.2011.5767905

[8] Ying, Xiaowei, and Xintao Wu. "Randomizing social networks: a spectrum preserving approach." In Proceedings of the 2008 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, pp. 739-750, 2008. doi : 10.1137/1.9781611972788.67

[9] L. Backstrom, C. Dwork, and J. M. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," Commun. ACM, vol. 54, no. 12, pp. 133–141, 2011. doi : 10.1145/1242572.1242598

[10] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," Int. J. Uncertainty Fuzziness Knowl. Based Syst., vol. 10, no. 5, pp. 571–588, 2002. doi : 10.1142/S0218488502001648

[11] L. Zou, L. Chan, and M. T. Ozsu, "K-automorphism: A general framework for privacy preserving network publication," in Proc. VLDB Endowment, vol. 2, pp. 946–957, 2009, doi : 0.14778/1687627.1687734

[12] J. Medina and M. Ojeda-Aciego, "Multi-adjoint t-concept lattices." Information Sciences, vol. 180, no. 5, pp. 712–725, 2010, doi : 10.1016/j.ins.2009.11.018

[13] Tabales N, Rey J, Carmona F, Caridad Y, "Commercial properties prices appraisal: alternative approach based on neural networks.", Int. Journal of Artificial Intelligence, vol. 14, no. 1, pp. 53-70, 2016.

[14] O. Geman, H. Costin, "Automatic Assessing of Tremor Severity Using Nonlinear Dynamics, Artificial Neural Networks and Neuro-Fuzzy Classifier," Advances in Electrical and Computer Engineering, vol.14, no.1, pp.133-138, 2014, doi:10.4316/AECE.2014.01020

[15] C. Pozna, N. Minculete, R.-E. Precup, L. T. Kóczy, Á. Ballagi: "Signatures: Definitions, Operators and Applications to Fuzzy Modeling", Fuzzy Sets and Systems, Vol. 201, pp. 86-104, 2012.

[16] Y. Wang, L.Xie, B. Zheng, and K. C. Lee, "High utility k-anonymization for social network publishing", Knowledge and Information Systems, vol. 41, no. 3, pp. 697-725, 2014. doi: 10.1007/s10115-013-0674-2.

[17] Fortunato, S. "Community detection in graphs." Physics reports, vol. 486, no.3-5, pp.75-174, 2010. doi: 10.1016/j.physrep.2009.11.002

[18] Shi, J. and Malik, J. "Normalized cuts and image segmentation." IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 8, pp. 888-905, 2000. doi : 10.1109/34.868688

[19] Newman, M.E. and Girvan, M. "Finding and evaluating community structure in networks." Physical review E, vol. 69, no. 2, pp. 026113, 2004. doi : 10.1103/PhysRevE.69.026113

[20] Wong, Raymond Chi-Wing, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. "Minimality attack in privacy preserving data publishing." In Proceedings of the 33rd Int. conference on Very large data bases, vol. 16, no. 4, pp. 543-554, 2007. doi : 10.1007/s10115-006-0035-5

[21] Maiya, Arun S., and Tanya Y. Berger-Wolf. "Sampling community structure." In Proceedings of the 19th international conference on World wide web, pp. 701-710. ACM, 2010. doi : 10.1145/1772690.1772762