Privacy Preservation Analysis in Social Network Graphs for avoiding Community Detection and Publishing Sensitive Information

Sharath Kumar J, Maheswari N

Abstract*: We have too much data online in this modern world that is available to the public for some functionality or the other. This information is not misused by the general user but an adversary how wants to extract sensitive information and uses it against the user is definitely a problem in this social era. This information present online is not represented is not simple form like a tabular data but present in complex form as network graphs. The main objective of this research is to understand the problem of large graphs and provide a modified anonymized version of the original graph that holds the same structure and still anonymize to prevent information leakage. This is a huge task due to the complexity of graphs in large networks. This paper finds the problem that even though many algorithms are there for anonymity, it is still a difficult task to keep the adversary away. This work focus on partitioning the original graph based on modularity and then the edges are analysed and reconstruction of the anonymized graph takes place by modifying the edges present with the partitions such that the graph structure remains the same. The algorithm uses network hops to identify connectivity and modify the edges without losing the path. The nodes are not altered and no fake nodes or edges are added to maintain the structure. The anonymized graph can now be released and utilized. Till now work is done in edge modification and vertex modification has moved on to differential privacy leaving wide scope for the study on the problems in detail. This paper will showcase the entire graph modification with preservation of the structural properties.

Index Terms*: Social Networks, Data anonymization, Edge modification, Un-weighted graphs, Privacy, Social Network Randomization, Privacy preservation Data Mining.

I. INTRODUCTION

Today, social networks are publicly available and host a great deal of data on the internet. Several works have been proposed to protect sensitive user information before publishing online on the web to handle such networks. Due to the complexity of the graph networks, very few methods and algorithms are available and cannot be focused on selective sensitive information. In this paper, we focus on edge modification when the graph structure is altered based on the detection of the community and with minimal modification in order to preserve the usefulness of a graph. There are a number of algorithms and techniques available today, which make data more secure on the social web.

Many social-web offer free services in exchange for a large source of user information and some services on the social web are paid and users demand privacy for their paid content for not leaking to an external source. However, paid or unpaid data on social web data has to be shared to be used for studies or for business sake. Sometimes sanitized data is required for further processing and analysis. Whatever may be the case data has to secure in today's world and identity preserved. Otherwise, it will be total chaos and plenty of damage caused by many people associated with the data compromised. The ultimate goal is to publish social graphs in some noisy form such that it is ready for any distribution on the social web.

There are two types of certain problems in anonymization of networks: one being the known semantics and the other the unknown semantics of sensitive information. Known semantics are often concurred by traditional privacy preservation techniques and there will be no surprises after deployment of the anonymized graph. However, the unknown semantics network holds a different viewpoint as any information can be important and there is no clause in hiding the information. For this reason, we cannot just like that add noise and destroy the utility of the graph. Though the Gaussian noise technique and random perturbation technique are present, the utility of the graph is lost as more noise is added to the network.

II. PREVIOUS WORK

Some of them are traditional methods based on user inputs to select sensitive information and these anonymization techniques are mainly applicable to static tabular data sets. They comprise of either generalization or suppression techniques. When large graphs are considered there is complexity involved in preserving the structural properties and also gives high utility on analysis when anonymized. Since this requires an algorithm which considers all the factors before releasing the anonymized data, more research areas are still open for investigation.

The author uses clustering techniques to a Graph G and then a number of methods and algorithms are present for the graph anonymization but they are very limited to social graph due to the complexity of the graph structures [11]. Most algorithms apply to small and medium graphs,

Revised Manuscript Received on April 07, 2019.

Sharath Kumar J, School of Computing Science and Engineering, VIT University Chennai, India.

Maheswari N, School of Computing Science and Engineering, VIT University Chennai, India.

while dynamic graphs vary in scope and complexity. This is still an open problem because privacy is required in various organizations for everyday use. The data that is anonymized must be minimally modified and be very useful in accessing the data. Any type of analysis carried out on anonymized data will produce exactly the same results as the original data, but still, hide all sensitive information from a user.

The authors propose a UMGA (Uni-variant Micro-aggregation for Graph Anonymization) Algorithm. UMGA is a k-degree anonymization algorithm for massive networks [3], [5]. It uses the micro-aggregation concept to anonymize the network's node degree sequence. The edge swap is then applied to modify the graph structure to implement the anonymous k-degree sequence.

Only a few algorithms can be applied on large graphs due to the complexity involved in the anonymization of the graph. The paper introduces a k-grade anonymity algorithm on large networks. Given the G network, an anonymous G* k-degree network with the minimum number of edge changes. Our algorithm uses uni-variate micro-aggregation to anonymize the grade sequence and then modifies the graph structure to match the anonymous sequence of k-degrees. The algorithm is applied to various large networks and demonstrates its efficiency and practical use.

In this article the authors talk about RGO (Random Graph Obfuscation) Algorithm RGO [4], [2]. It uses various concepts to select borders that can be deleted or created on the anonymous network. This paper takes into account the problem of anonymization of large graphs and also the utility of the data that is released to the public. Random methods can work with medium or large graphs while achieving the desired privacy level. In this article, a simple and efficient algorithm for graph randomization is used to improve the data utility and reduce the loss of information on the anonymous graph; the algorithm also considers the relevance of the preservation the most important edges of the graph.

Louvain Method In 2008 [9] propose an algorithm for large networks based on communities to extract graph structure from large networks. This method uses heuristics and optimization of modularity results. The idea is to select a node and then based on modularity gain the other nodes are linked and a community is created. If a member node belongs to another community they are switched over to the other community. This process continues until no further addition or switches over of node take place between communities and thereby giving a positive gain in modularity in dividing the large network graph.

There is another study by [13], [6], [7] and [8] which also focuses on anonymization based on grouping with some parameters. All these study work on anonymization and have a certain level defined to prevent unwanted third party adversaries. This paper emphasizes the fact of how the edge is to be removed or altered by breaking the structural network of graphs. Consider if a new edge (x, y) has to be added, node s need to send an information message in the network then after any modification the new structure is propagated through the network. Whenever an edge is removed, the necessity is there to find if the link is intact to the parent node. If a new node is introduced it will not affect the structure unless it forms the main parent node in the network. Assign Cluster or Community uses a value in a table for each node, we can assign cluster and community labels.If we want to have crisp cluster assignments, we sum up all values for each; the h with the highest sum will be the cluster label. If we do not need crisp assignments, we normalize a sum and use them as a membership degree for a certain community.

III. Anonymized Graph Construction

In this section, we will discuss the process of how the newly constructed graph is going to be constructed by our proposed algorithm. The main technique to provide anonymity is achieved by reconstruction of the original graph such that the modified graph is totally anonymized by random edge modification. The key community structure and the vertex are preserved such that the modified graph cannot be inferred by unknown users. The anonymized graph provides strong privacy properties and cannot be inferred by even a strategic adversary having prior knowledge and full access to the static anonymized graph.

A. Partitioning of Original Graph

Graph Partitioning Algorithm as presented in Algorithm 1 discuss community best partitioning of graphs which uses the Louvain algorithm to identify the best community structures based on modularity. Once the partitions are defined by identifying the nodes and the edges for each partition, the intra-edges are also indentified which act as the potential connectivity between partitions. An edge set is created which holds all the edges present in the original graph and is called as term edges. The edges are refined by checks done to identify the partition they belong to and if required swapped between partitions. The edges present in each partition are removed from the term edges and that leaves us with the potential residual marginal edges.

B. Marginal Nodes Algorithm

Partition margin nodes take care of the nodes which are going to be common between the partitions. This is validated and identified using the residue of the term edges as shown in Algorithm 2. Further to this, the edges are compared with both partitions and with commonalities they are identified as marginal edges. This set is also required for anonymization. The partition marginal nodes act as the potential intra-partition edge set.

C. Partition Link Nodes Algorithm

Algorithm 3 computes which edges are going to connect between partitions. This takes into consideration the degree of nodes and also the anonymity factor reduction based on the probability value calculated using the length.

D. Selective Anonymization Perturbation (SVP)

Algorithm 4 determines when an edge is to be added or removed. The algorithm works to anonymize both inter and the intra-partition before adding them to the newly created graph. The original values are validated based on degree and length such that the original graph structure is maintained in the modified graph with anonymity.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Here we use hops to anonymize the links, hops are based on random walks on the graph and identifying the link which can be removed and the links which can be added to the perturbed graph.

Algorithm 1 Graph Partition Algorithm (GPA)

Input: Original Graph (G)

Output: Partitions of Original Graph (G) stored in Graph (G_i^1)

```
1. Partition the graph G into 'N' partitions
2. For each partition Pi (N).
    2.1. Identify the list of nodes(v) for Partition Pi
    2.2. Create a Sub Graph ($G_i^1$) for every Partition
         Pi identified
    2.3. Remove the sub graph edges ($G_i^1$) from the
         original graph (G) edges
    2.4. Selectively perturbate the sub graph ($G_i^1$)
         using SVP technique with 'k' trials and 't'
         hops
    2.5. Append the resulted perturbed graph I with
         $G^1$ to construct the complete perturbed Graph
         (PG).
Create the partition edges nodes set using the balance
term edges from step 2.
```

Algorithm 2: Partition Margin nodes (PMN) Algorithm

Input: Partitions of Original Graph (G) stored in Graph (G_i^1)

Output: VASET and VBSET groups

```
1. Perturbate the marginal nodes as follows:
    1.1. For each partition $P_i$
    1.2. For each partition $P_j$
    1.3. If $p_i$ != $P_j$
        1.3.1. Create set $V_a$ which consists of all term
               nodes and all nodes of $P_i$
        1.3.2. Create set $V_b$ which consists of all term
               nodes and all nodes of $P_j$
        1.3.3. For each node in $V_a$ Check if n is a
               neighbour in G and belongs to $V_b$. then
        1.3.4. Create set VASET that includes nodes
               from $V_a$ which are the neighbours of G
               and the members of $V_b$
2. Create set VBSET that includes nodes from $V_b$ which
    are the neighbors of G and the members of $V_a$
```

Algorithm 3: Partition to Partition Link Algorithm

Input: Perturbed Graph (G_i^1)

Output: Connected edges between Partitions in Perturbed Graph (G_i^1)

```
1. Using the PMN algorithm, Create the edge set for
    marginal nodes E(s,d) where s belongs to VASET
    and d belongs to VBSET and combine the term
    edges.
2. Calculate the degree for VASET AND VBSET.
3. Find the probability value by dividing the value of
    Length VASET and degree for edges with the sum
    of length of VASET and VBSET, the product of
    length of edges.
4. Comparing the probability value and random value
    from 0 to 1, find the instance when an edge has to be
    added.
5. Add the edge to the perturbed graph (PG).
```

Algorithm 4: SVP- Selective perturbed Algorithm

Input: Perturbed Graph (G_i^1)

Output: Perturbed Graph (G_i^1) similar to Original Graph

```
1. Create a empty Graph (M)
2. For each node (u) from the partition graph or the
    sub graph received:
        a. Find if edges are available and hops are
           possible in the network
        b. If yes select the node by checking
           neighbours and find the z value which is
           the potential edge
        c. Based on hops and z value found connect
           the two nodes to form an edge and
           maintain the utility.
        d. Else the degree of Original graph G is
           copied to the empty graph (M) node
           under consideration.
        e. Use random to compare the probability
           value and degree of the node(u)
               i. Edge is added to the Graph (M)
        f. By this selective edges are added based on
           connectivity and the new sub graph is
           created as Graph (M)
```

IV. EXPERIMENTAL ANALYSIS

Our algorithm SVP - selective perturbation algorithm is implemented in python language using network X package in Jupyter notebooks. All experiments are run on a Windows 10 system with Intel ® Core— i5-7200 CPU with 2.50GHz and 8 GB of memory.

A. Networks used for testing

We ran experiments in three networks Polbooks-US Politics book data, Polblogs-Political blogosphere data, and GrQc- Collaboration network.

Table 1: Network properties of few data-sets

Network	Nodes	Edges	Degree	K
Polbooks	105	441	8.40	1
Polblogs	1222	16.714	27.31	1
GrQc	5242	14,484	5.53	1

In the above Table 1, it shows a summary of the networks' main features including the number of vertices, edges, average degree, and default k-anonymity value. US politics book data (Polbooks) [8] is a graph network US politics books published in 2004 presidential election and sold by Amazon.

Political data on the blogosphere (Polblogs) [1] compile data on links between political blogs in the United States. Finally, the GrQc collaboration network [10] is scientific collaborations between authors of papers submitted to the category of general relativity and quantum cosmology.

Some of the Graph measures considered is:

B. Modularity

Modularity is one of the measures to check the structure of networks or graphs [12]. The measure is done by analyzing the strength of division of a network into different modules like groups, clusters or communities. It is defined by the fraction of all edges that are present in the communities minus the values required from the original graph in which the nodes have similar degrees. However, the edges are in without considering the communities.

C. Transitivity

This measure is one kind of clustering coefficient, which considers the local loops of a vertex. It also measures and characterizes such loops

D. Empirical results

Three different networks as shown in Table 1 are considered for checking the structural properties of the network. The existing algorithms measure values are taken from [2], [14], the authors compare the algorithms of UMANG -Random algorithm, UMANG-NC algorithm, kDA - K degree of anonymization and VA -Vector addition algorithm. All these modify the structure of the graph based on k values. The k values determine the category for performing the anonymity on the network. The measure is required to find the structural properties of a graph are preserved. We use two measures namely the modularity and the transitivity measure for k values 2, 5 and 10. Here in our algorithm we indicate the k value as degree to which the edges can be added in the network and also determines the level to which anonymity can take place.

The first network under consideration is shown in Table 2, here we see that the modularity value (Q) for SVP algorithm is better than other algorithms. However, in transitivity(T), Umang NC algorithm performs better. Here the transitivity value is slightly more due to edge modification to improve the privacy. We see that the structure is maintained for various values of k based on modularity.

Table 2: Polblogs Network

Modularity (Q)	K=2	K=5	K=10	DEV
SVPA	0.489	0.491	0.491	0.0012
R	0.403	0.405	0.402	0.0015
NC	0.404	0.403	0.402	0.001
kDA	0.402	0.396	0.385	0.0086
Transitivity (T)	K=2	K=5	K=10	DEV
SVPA	0.252	0.224	0.227	0.0154
R	0.224	0.224	0.223	0.0006
NC	0.224	0.224	0.224	0
kDA	0.225	0.221	0.221	0.0023
VA	0.219	0.205	0.183	0.0181

The second network under consideration is shown in Table 3, here we see that the modularity value for SVP algorithm matches the algorithm of kDA and also has closer values to show that the structural properties are intact.

Table 3: Polbooks Network

(Q)	K=2	K=5	K=10	DEV
SVPA	0.570	0.609	0.576	0.021
R	0.400	0.398	0.394	0.003
NC	0.400	0.401	0.385	0.009
kDA	0.390	0.360	0.350	0.021
(T)	K=2	K=5	K=10	DEV
SVPA	0.330	0.310	0.298	0.016
R	0.350	0.330	0.313	0.019
NC	0.350	0.339	0.324	0.013
kDA	0.330	0.330	0.300	0.017

The third network shown in Table 4 is partially considered in other algorithms due to the nature of the input graph which does not have labels. However, this does not create an issue in our algorithm due to the complete graph restructure. We also see that the values presented in modularity and transitivity is better when compared to other algorithms.

We compare the Degree distribution as shown in Figure 1 in the full network; x-axis is degree value and y-axis is the number of nodes with that degree for the original graph and the anonymized graph with
k value as 2.

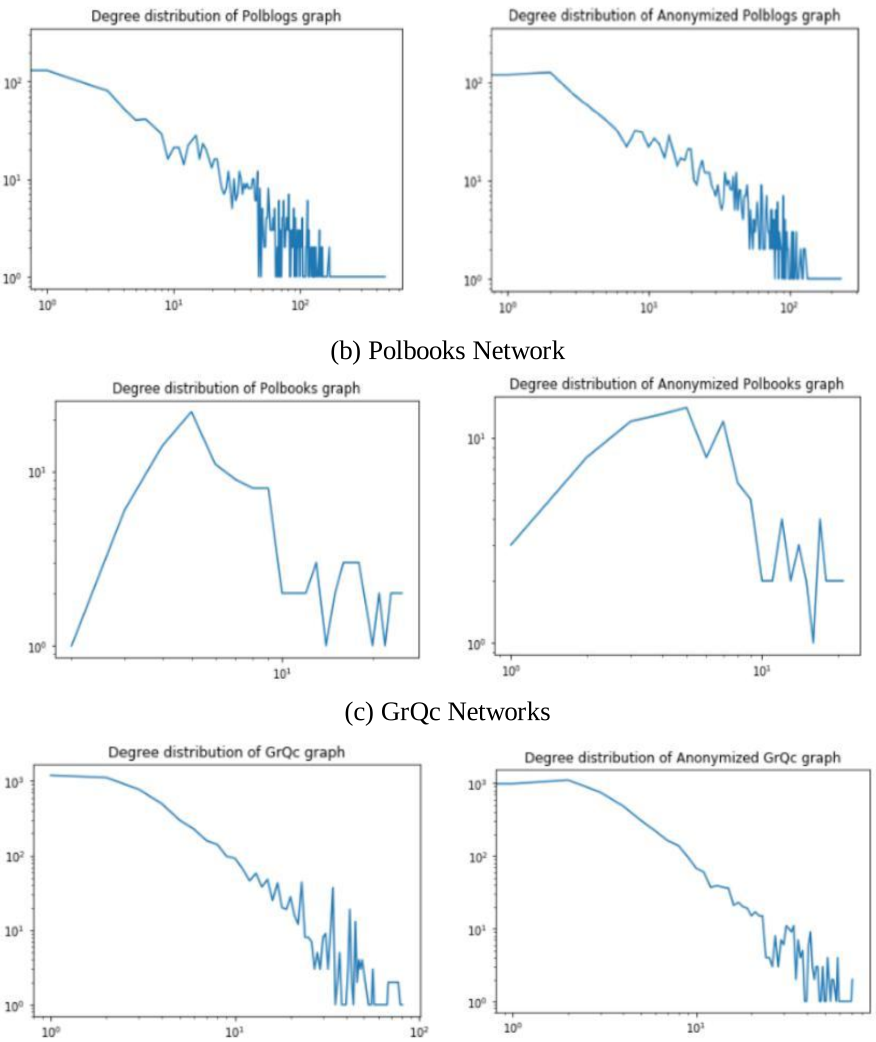
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The graph clearly shows that the structural properties are intact even after anonymization.

Table 4: GrQc Network

Modularity (Q)	K=2	K=5	K=10	DEV
SVPA	0.921	0.926	0.927	0.003
Transitivity (T)	K=2	K=5	K=10	DEV
SVPA	0.462	0.450	0.421	0.021
R	0.622	0.588	0.548	0.037
NC	0.625	0.588	0.584	0.022

Figure 1: Degree Distribution Comparison



(a) Polblogs Network

(b) Polbooks Network

(c) GrQc Networks

Figure 1a, 1b, and 1c show the degree distribution of the original and the perturbed graph. x-axis is degree value and y-axis is the number of nodes. When the two graphs are compared, the graph degree distribution follows just like the original graph but still different. Therefore an unauthorized user will not be able to tell the difference to infer if it is modified. The graph structural properties are intact, considering the utility of the graph, SVP algorithm keep the number of nodes intact. This means no changes are done to nodes, however, based on prior knowledge a third party hacker cannot gain information in the anonymized graph because the entire graph is modified to the structural properties and changes of how the edges are connected. These provide an enhanced utility which can be presented in further study.

V. CONCLUSION

We study the problem of complexity in large graphs and also check our algorithm to analyze the anonymized graph with structural measure property of graphs. There are various algorithms which offer edge modification and up to a level when an adversary can have background knowledge, try to break the network and finds sensitive information. However, this adversary with background information will not be able to get information on sensitive data due to complete graph modification. This paper covers large networks anonymity of clusters and also node identification. The entire graph is modified and restructured, still the graph properties and utility is highly preserved. This modified graph will have the same number of nodes but differs by the number of edges.

Any third party person cannot deduce from existing links. Also the partitions with best communities is also preserved and anonymized. Utility is not compromised, Work will continue to study with scalability with very large networks and also look out other graph reconstruction methods based on edge modifications.

REFERENCES

1. L. A. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In proceedings of the 3rd international workshop on Link discovery, pages 36_43. ACM, 2005.
2. J. Casas-Roma. Privacy-preserving on graphs using randomization and edge-relevance. In Modeling Decisions for Artificial Intelligence, pages 204_216. Springer, 2014.
3. J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra. An algorithm for k-degree anonymity on large networks. In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pages 671_675. ACM, 2013.
4. J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra. Analyzing the impact of edge modifications on networks. In International Conference on Modeling Decisions for Artificial Intelligence, pages 296_307. Springer, 2013.
5. J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra. K-degree anonymity and edge selection: improving data utility in large networks. Knowledge and Information Systems, 50(2):447-474, 2017.
6. S. Chakraborty and B. Tripathy. Privacy preservation in social networks through alpha anonymization techniques. In Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on, pages 1602_1603. IEEE, 2015.
7. S. Chakraborty and B. Tripathy. Alpha anonymization techniques for privacy preservation in social networks. Social Network Analysis and Mining, 6(1):29, 2016.
8. R. Ghosh and K. Lerman. Community detection using a measure of global influence. In Advances in Social Network Mining and Analysis, pages 20_35. Springer, 2010.
9. P. Held and R. Kruse. Online community detection by using nearest hubs. arXiv preprint arXiv:1601.06527,2016.
10. J. Leskovec, J. Kleinberg, and C. Faloutsos Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1):2, 2007.
11. R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. Discrete Applied Mathematics, 144(1-2):173_182, 2004.
12. Y. Song and S. Bressan. Fast community detection. In International Conference on Database and Expert Systems Applications, pages 404_418. Springer, 2013.
13. G.-M. Yang, J. Yang, and J.-P. Zhang. Achieving(alpha, k) anonymity via clustering in data publishing. Dianzi Xuebao(Acta Electronica Sinica), 39(8):1941_1946, 2011.
14. X. Ying, K. Pan, X. Wu, and L. Guo. Comparisons of randomization and k-degree anonymization schemes for privacy-preserving social network publishing. In Proceedings of the 3rd Workshop on Social Network Mining and Analysis, page 10. ACM, 2009.

AUTHORS PROFILE

Sharath Kumar J is pursuing his PhD. Under the guidance of Dr. N Maheswari and also a Faculty of School of Computing Science and Engineering at Vellore Institute of Technology (VIT), Chennai. He has published research articles on Data privacy in social networks as a part of his research. His interests are in data Mining, Privacy preservation and data analysis.

Maheswari. N is a Professor in the School of Computing Science and Engineering at Vellore Institute of Technology (VIT), Chennai. She has 18 years of academic experience currently her research interests include Machine Learning and Analytics. She has authored and co-authored a large number of research articles in the journals. She had written a book chapter on Large Scale Data Analytics Tools in the Springer Book Series. She is a member of ACM. She serves as a Doctoral Committee member of the research meeting held at various universities. She has examined several doctoral theses and served as an examiner in many universities. She also takes part in various Board of Studies meeting for curriculum designing in various universities. She serves as a reviewer in national and international conferences. She also serves as a reviewer for Social Network Analysis and Mining, Journal of Computer and Information systems etc... She has received the award for Excellence in Education from the Lions Club of India and the best faculty award from the VIT Chennai.