Preservation of People's Privacy in Social Networks Based on Clustering-Based Honey Bee Optimization Algorithm

Mehdi Sadeghzadeh
Department of Computer Engineering,
Science and Research Branch, Islamic
Azad University,
Tehran, IRAN
e-mail: sadeghzadeh1999@gmail.com

Mohammad Reza Pourfasih
Arvandan University of Khorramshahr
Khorramshahr, IRAN
e-mail: pourfasieh@yahoo.com

***Abstract*— Nowadays, the use of social networks has developed widely. When people publish too much private information about themselves in these networks, their information may be attacked by an adversary, so there is a need to protect the privacy of people on these networks. One of the methods of preserving private information is k-anonymization. Anonymization is encountered with the challenge of data loss. A method is needed that ensures data anonymization while the utility is also well preserved. In this research, we try to create a proper model for data privacy and utility preservation by combining a graph cut clustering method and an artificial bee colony optimization algorithm. Two datasets are used in this research, which are Ca-GrQc including 5242 nodes and 14496 edges, and Polbooks containing 105 nodes and 441 edges. Three measures are used to evaluate this model, including, Transitivity, APL, and ACC. Finally, based results it can be declared proposed method, is a proper method for preserving privacy in social networks.**

***Keywords*— Social networks, Graph-cut clustering, Artificial Bee Colony Optimization, K-Anonymization, privacy**

I. INTRODUCTION

The ever-increasing expansion of the Internet has led to the emergence of a new subject of study called social networks on the Internet, which is also called virtual or online communities. Nowadays, people use many online social networks such as Facebook, Twitter, Instagram, Telegram, etc. Part of the service of these networks is that they allow users to publish many details about themselves that are related to the nature of that social network and connect with their friends. Some of the information disclosed in these networks is private.

Considering that there is a lot of information about users' privacy in the data of social networks, it is necessary to make corrections on the real data in order to ensure the security of users' privacy. If the correction is too much, the usefulness of the social network data will decrease. On the other hand, if the modification is not sufficient, the privacy information is not well protected. For this reason, a balance must be established between privacy protection and utility.

II. RELATED WORKS

Liu et al. use an entropy probability distribution to quantify the level of anonymity. As discussed, the random disturbance mechanism does not reach a high degree of anonymity without reducing the characteristics of the graph [1].

The first innovative k-degree anonymization was proposed by Liu and Terzi. The assumption of the researchers was that the enemy, in relation to specific vertices from the previous information network, has and using this knowledge tries to identify the identity of network vertices. At this degree, sequence method related to the original graph is anonymized by the dynamic programming method. First, the algorithm generates the degree sequence of the graph G and then converts it into an equal minimum degree sequence k [2].

Roma et al. suggested a simple and efficient algorithm of k-degree anonymity in networks. Their algorithm is k-degree network anonymity with a minimum number of edge. They compare the present algorithm with other well-known k-degree anonymized algorithms. They compared and showed that information loss is reduced in real networks [3].

Makwan et al. suggest a method with the aim of anonymizing while preserving the structure of the method by examining. They proposed k nodes with a common communication to increase security. The proposed method is considered to get k common connection of each node at the time of edge addition and removal, select the lower cost node with the first candidate (with a less common connection) and then maintain the number of common connections. It improves anonymization, but the drawback of the mentioned method is the long execution time due to multiple search phase surface, and the lack of optimization is the way to change the graph [4].

Ni et al. proposed a method with the aim of preserving privacy through anonymization and enhancement speed,

clustering along with parallel anonymization of clusters. The advantage of the present method is its higher speed compared to sequencing methods, as well as the use of centrality in selecting candidates for change. In order to minimize the loss of information and the lack of optimization in the anonymization phase, the optimal graph is to use clustering. [5]

Tian et al. suggested anonymization using the k-means method for clustering, and they applied the greedy method in order to change the degree by increasing or decreasing the edge of each subgraph. Advantage is the present method uses the degree of centrality to identify valuable nodes and prevent them from changing in order to preserve the nature of the graph, and its problem is the lack of optimization for the anonymization phase and the lack of optimization considering the addition of nodes is unrealistic [6].

III. PROPOSED METHOD

The input of this method is a standard directed or undirected graph. The proposed method is introduced with the approach of maintaining the state and the main structure of the input graph. Also, as complementary parameters, an attempt was made to perform the best k-anonymity in this graph. The proposed method includes several stages of graph processing, optimization and, finally, k-anonymization.

A. Description of Proposed Method

The proposed method in this research, which is shown in Fig.1, includes pre-processing classes, initial k-anonymization and its supplementary stage. In each section, an operation has been performed, which is briefly mentioned (indicates Steps 1 and 2 of preprocessing, Steps 3 and 4 of k-initial anonymization, and Steps 5 to 8 of the supplementary step):

Step 1: The adjacency matrix is built from the pair of connections between the input nodes of the dataset.

Step 2: In this step, the degree and degree centrality of each node of the graph are calculated. These two values are calculated as the characteristic and ID of each node.

Step 3: Clustering is done with the help of graph cutting method; at this stage, the graph is divided into sub-graphs. The goal of dividing large graph data into optimal sub-graphs is to reduce the computational cost and also to provide the possibility of solving large graph data through the existing methods.

Step 4: The k-anonymity of each cluster is checked. In this way, each cluster is checked and if it is not k-anonymous, it goes to the next step.

Step 5: In the fifth step, the best k-anonymity situation for each cluster (under the graph) is calculated with the help of honey bee optimization; optimality means k-anonymization with the least cost of changing the number of edges and vertices of each cluster and the maximum amount of anonymity of that cluster according to the default parameter k. At the present stage, the best situation for anonymization is obtained with the help of ABC optimization method. The use of this method compared to the previous methods that had simple optimizations or did not use the optimization phase at all, can finally get the best result with the least change in the graph along with the most k-uncertainty in each subgraph.

Step 6: According to Step 5, edges and vertices in each cluster are added and removed.

Step 7: This step is the reconstruction of the original graph according to the clusters. At this stage, the variables are connected to each other, the graph is reconstructed and it is prepared to calculate the k-anonymous parameters.

Step 8: The current step is to calculate the anonymity check parameters and the parameters related to the amount of information loss of the original graph after the changes.

B. Preprocessing

The first step of constructing the mathematical representation of the graph is the adjacency matrix. In this step, the input data, which were initially in the form of ordered pairs of the connections of two nodes, are considered in the form of an n by n matrix of connections. The output matrix has rows and columns for the number of vertices, and due to the unweighted consideration of the graph, if there is an edge between two vertices, the number 1 is placed in it, and if there is not, the number 0 is placed in it (zero-one matrix).

A graph G is assumed with a set of vertices $V(G) = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $E(G) = \{e_1, e_2, \ldots, e_m\}$. We call the matrix $(a_{ij})n \times n$ A(G) = the adjacency matrix of the graph G with zero and one entries, where a_{ij} = 1 if and only if two vertices v_i and v_j are adjacent in G, otherwise, a_{ij} = 0, where a_{ij} are the only edges that connect two vertices v_i and v_j, (each ring counts as two edges). From this definition it follows that every adjacency matrix is symmetric, that is, $A = A^T$.

In this step, the closeness centrality degree is calculated for each vertex. The degree of centrality, along with the degree of each vertex obtained from the adjacency matrix, is used for clustering in the next step.

In short, in this section, the input is ready for clustering. In this step, first the graph is built, then the degree of each node and the centrality of its degree of closeness are calculated.

C. Initial Graph Clustering

The clustering operation is performed using graph-cut clustering, considering the neighborhood and degrees of each vertex. The graph-cut clustering operation aims to transform the graph into clusters that have the lowest edge cutting cost and the highest profit in terms of the balance of the total degrees of the vertices in the two parts that are created after cutting. In general, the resulting clusters will have the following characteristics:

- They are direct neighbors.
- Created with the lowest cutting cost.
- There is the most balance in the total degree of vertices and the degree of closeness centrality.

In simpler terms, the task of this section is to divide the graph into optimal sub-sections and ensure that each sub-

section has at least k nodes. These k nodes will be a connected subgraph.

The first step is to determine the number of nodes in each slice. Considering that the method seeks k anonymity, then each cluster must have at least k nodes. To do this, the number of graph nodes is divided by k and the number of clusters is obtained. Of course, it is possible that the result of the division is incorrect, in which case the correct part of the floor is added to the clustering algorithm as the number of clusters under a new condition. For example, if the graph has 11 nodes and needs to be bi-anonymized, the number of clusters will be five, and as a result, there will be at least one cluster with the size of three vertices in the graph.

D. Specifying Non-K-anonymos Clusters

k-anonymization requires that the clusters have k vertices with the same degree. According to the results of the clustering stage, there are three different situations for the clusters. If the cluster has at least k members and all vertices have the same degree, the flag is set to zero. If the cluster has at least k members, but not all vertices have the same degree, it gets a flag of one, and finally, if the cluster has less than k members, this cluster is merged with the closest cluster (in terms of neighborhood) that has a flag of one (k is not anonymous).

E. Completing K-anonymization Using The Bee Optimzation Algorithm

The output of the previous step is a number of k-unknown clusters and a number of non-k-unknown clusters. Completing k-anonymization requires that non-k-anonymous clusters be anonymized by removing and adding k-vertices and edges. For this task and to reach optimal conditions, the optimization algorithm of the honey bee has been used.

In the previous section, a method for clustering vertices was proposed to determine the same degrees and to make the social network graph as k-anonymous as possible. A number of clusters became anonymous, despite having the same degree of k. The anonymized clusters are discarded and the rest of the clusters, which are not anonymized, are k-anonymized by removing and adding vertices and edges. When adding and removing vertices and edges, a balance must be established between the anonymity of the graph and the usefulness of the data. In other words, for anonymization, too many vertices and edges should not be added or removed because the real data will be lost, and the amount of anonymization should not be so low that the privacy of individuals is compromised by the adversary. For this purpose, the bee optimization algorithm is used here for optimization.

The main idea of using the present algorithm is that by increasing or decreasing the degree and vertex in the cluster, all the vertices in the cluster will reach the same degree value and also minimize the value of the fitness function, which will be discussed later. It should be noted that the following explanations are described for one unit of algorithms and the architecture works in this way.

The steps of this algorithm to optimize and create an anonymous graph are as follows:

The bee optimization algorithm starts with an initial population that is randomly created in the determined interval (dimensions of the problem).

Two parameters, the number of vertices in the cluster and the degree of the vertex, are considered as the representation of the answer. These values change for the number of vertices between m/2 (half the size of the initial cluster) and 2m (twice the number of the initial cluster), as well as the degree of the vertex between zero and the largest degree of the vertex in the studied cluster.

The value of the fitness function for each response string (bee) is calculated and checked. The value of the fitness function is defined as follows:

$$\Delta = \left| \sum_{i=1}^{n} d_c(v_i) - \sum_{i=1}^{n} d_{\tilde{c}}(v_i) \right| \qquad (1)$$

Here, the fitness value is equal to the difference between the sum of the degrees of the cluster vertices in the original graph and the sum of the degrees of the cluster vertices in the anonymous graph.

Apply final changes to the original graph. Based on the result obtained in the previous step, the degree of each vertex is determined.

According to the vertex and degree of the original graph and the cluster size and degree calculated from the optimization algorithm, changes are made in the graph as follows:

If the degree calculated for the vertex is lower than the degree of the original graph, it must be removed from its edges to reach the desired degree of anonymity.

If the degree calculated for a vertex is greater than the degree of the original graph, an edge must be added to it to reach the desired degree of anonymity.

If it is necessary to add vertices and edges, first a vertex is added to the number after the last number of vertices in the graph with zero edge, and based on which vertex needs an edge, it is connected to the additional vertex with an edge.

If there is a need to reduce the vertex and edge, first the edges related to the vertex are removed and then the corresponding vertex is removed and, finally, the number of the other vertices in the shift graph is given.

The computational complexity of the proposed method is equal to O (L×N×M). L is the number of clusters to be anonymized, N is the number of the initial population, and M is the number of iterations of the algorithm.

IV. SIMULATION RESULTS

To evaluate the effectiveness of the proposed method, three important analytical criteria in social networks, including average path length, average clustering coefficient, and transferability were investigated [7]. These metrics are calculated in the anonymous graph output.

• Average path length: The distance between two vertices u, v is the shortest path between u and v in the original graph. The social distance between all connected pairs in a graph is measured by average path length (APL) and calculated through the equation.

$$APL = \frac{\sum_{(u,v)\in RP} SPL(u,v)}{|RP|} \qquad (2)$$

The RP parameter determines all pairs of available vertices and SPL(u,v) is the shortest path length between the vertices u and v. This measure is related to the information efficiency or transmission volume in the network.

• Average clustering coefficient: The clustering coefficient of a vertex is a ratio of possible triangles that exist through the vertex and is calculated through the following relationship.

$$C_u = \frac{2T(u)}{\deg(u)\ \ (\deg(u)-1)} \qquad (3)$$

The parameter T(u) is the number of triangles through node u and deg(u) is the degree of u. Then, the average clustering coefficient of a graph is defined as the average clustering coefficient of all vertices.

• Transferability: It is one of the transferability clustering coefficients that defines and measures the local loops near a vertex. In other words, in the graph, the transferability is calculated by the number of triangles and triple connections by the relation.

$$Transitivity = \frac{3 \times NumberOfTriangles}{NumberOfConnectedTripleOfVertices} \qquad (4)$$

Considering that the main graph under investigation is a big data problem and trial and error on this data is very time-consuming, for this reason, a real small data was used as Polbook data. The obtained parameters are the result of trial and error from this data set. Considering that the problem could be solved with the default parameters of the proposed evolutionary method, the ratio of the population to the average error loss was used as the efficiency criterion. As a result, the smallest population that produced the appropriate output was selected as the optimization population size. The figure shows how to choose.

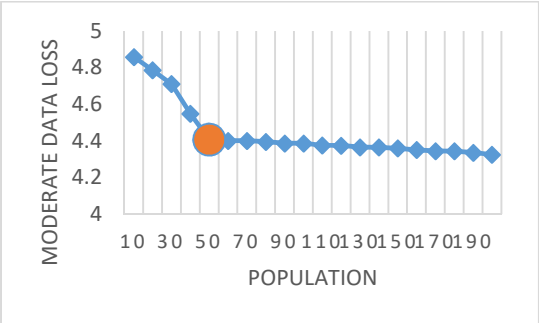

Fig. 1. Moderate data loss vs. Population

According to the shape of the population, 50 has a worse response than most populations, but this amount of difference is bearable in relation to the overhead that enters the system. The following graphs show the results of comparing the proposed model with the KDVEM and VertexAdd algorithms in [7] and the proposed method:

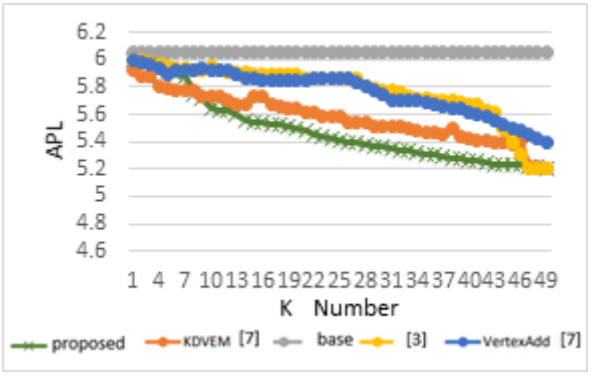

Fig. 2. APL vs k number

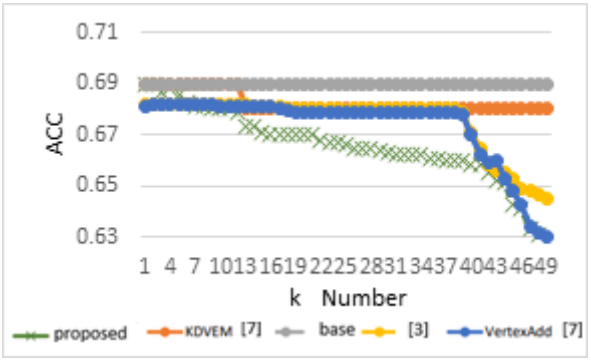

Fig. 3. ACC vs k number

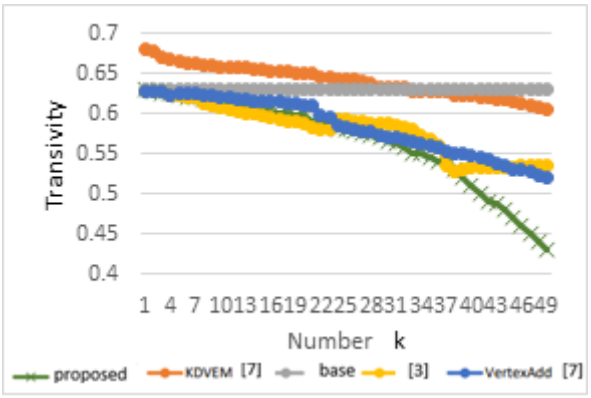

Fig 4. Transitivity vs k number

The results of the average path length of the proposed method are worse than KDVEM in smaller k, and with increasing k, the proposed method has better results, but in general, they are less than the original graph and very close to each other. But the proposed method has better results than the VertexAdd method.

The proposed method has created more anonymity than the KDVEM method in the average clustering coefficient and has had a better result than the VertexAdd method except for the first few small k.

For the transferability criterion, the k-anonymization result of KDVEM and VertexAdd methods is weaker than the proposed method. Only [3] has worked close to the proposed method.

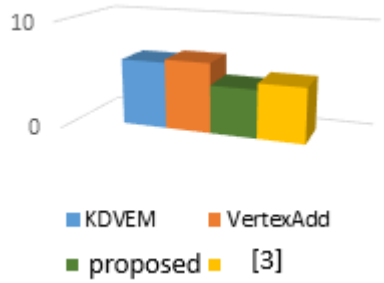

Fig. 5. Results of information loss of the proposed model compared to other methods

REFERENCES

[1] X. Liu, Q. Xie, L. Wang, "Personalized extended (α, k)-anonymity model for privacy-preserving data publishing", *Concurrency and Computation: Practice and Experience*. vol. 29, no. 6, pp. 25-29, 2017.

[2] K. Liu, E. Tersi, "Towards identity anonymization on graphs", *SIGMOD Conference,* pp. 93-106, 2008.

[3] J. Casas-Roma, J. Herrera-Joancomarti, V. Torra, "Evolutionary algorithm for graph anonymization", 2014, http://arxiv.org/abs/1310.0229v2

[4] K.R. Macwan, S.J. Patel, "k-NMF anonymization in social network", vol. 1061, no. 4, pp.601-613, 2018.

[5] S. Ni, M. Xie, Q. Qian, "Clustering Based k-anonymity algorithm for privacy preservation", *IJ Network Security*, vol. 19, no. 6, pp. 1062-1071, 2017.

[6] H. Tian, Y. Lu, J. Liu, J. Yu, "Between centrality based k-anonymity for privacy preserving in social networks*", in Proceedings of 16th international conference on advances in mobile computing and multi-media*, pp. 3-7, 2018.

[7] T. Ma, Y. Zhang, J. Cao, J. Shen, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhann, "KDEVM: a k-degree anonymity with vertex and edge modification algorithm", *Computing*, vol. 97, no. 12, pp. 1165-1184, 2015.