An Improved K-anonymization Approach for Preserving Graph Structural Properties

A. Mohammed Hanafy, Sherif Barakat, Amira Rezk
Dept. of Information System, Faculty of Computers and Information
Mansoura University, Mansoura, Egypt

***Abstract*—Privacy risks are an important issue to consider during the release of network data to protect personal information from potential attacks. Network data anonymization is a successful procedure used by researchers to prevent an adversary from revealing the user's identity. Such an attack is called a re-identification attack. However, this is a tricky task where the primary graph structure should be maintained as much as feasible within the anonymization process. Most existing solutions used edge-perturbation methods directly without any concern regarding the structural information of the graph. While that preserving graph structure during the anonymization process requires keeping the most important knowledge/edges in the graph without any modifications. This paper introduces a high utility *K*-degree anonymization method that could utilize edge betweenness centrality (EBC) as a measure to map the edges that have a central role in the graph. Experimental results showed that preserving these edges during the modification process will lead the anonymization algorithm to better preservation for the most important structural properties of the graph. This method also proved its efficiency for preserving community structure as a trade-off between graph utility and privacy.**

***Keywords*—Privacy; social networks; anonymization; edge-perturbation methods**

I. Introduction

Social network sites have become one of the largest sources of personal information. Daily, millions of users can use social applications like Twitter, Facebook, and LinkedIn to communicate with others. The increase in data being collected from different social network sites has attracted many researchers and social network analysts for extracting knowledge from data [1]. Hence, social network data publishing for analysis purposes becomes Inevitable, as the structural data analysis and studying the relations between individuals can serve many fields including marketing and also business. Social data includes a large amount of sensitive information about individuals, so releasing data of social networks in its primary form without anonymizing it could expose data to many attacks [2], [3], which harms the user's privacy. Many types of data privacy-related attacks had been discussed in previous literature [4], [5], which were summarized as follows: identity disclosure, sensitive attribute disclosure, and link disclosure risk. That's why privacy preservation methods must be implemented by specialists before the release of network data to the public.

The re-identification attack causes dangerous violations of social networks which harm user's privacy. An adversary can violate the user's privacy in two ways: (1) either by reaching the target's personal information such as name, edge, and salary, known as profile data, or (2) by utilizing the graph structural information. Recognition of the topological structure of graphs and relations between individuals enables an adversary to utilize his background knowledge to re-identify individuals. Once an adversary recognizes a specific person in the social network, all sensitive information related to him becomes identified. Also, confidential information regarding the belonging of individuals to a particular community becomes disclosed. For example, in the healthcare domain, PatientsLikeMe is a social network site that consists of several communities of patients. Each community represents the patients that suffer from the same illness. To keep track of their health and benefit from patient-reported concerns, members of this site are allowed to exchange private information such as health status and treatments [6]. In such a case, the disclosure of a patient's existence in a particular group will result in revealing all secret information that they share with others and violating their privacy.

The primitive way that people follow to prevent re-identification attacks, for the publishing data of social networks, is to delete a user's identifier attributes and replace them with symbols or synthetic identifiers. This method is known as simple and naïve anonymization. The authors in [7] presented two types of attacks of the naïve-anonymized graph: passive attack and active attack, which means that this simple method of anonymizing graphs is not enough to prevent the re-identification attack. The attacker can exploit his background knowledge concerning only the graph structure to reach the target and breach privacy.

For example, in the above-displayed graph shown in Fig. 1, each vertex/node represents an individual, and the edge connecting between two individuals represents the relation between them. After performing the naïve anonymization on the original graph G, we can get an anonymous version G^* as shown in Fig. 1(B). If an attacker has some background knowledge about Carl and knows that Carl has five friends. Hence, he can re-identify Carl in the anonymously published graph G^* and reach all sensitive information about Carl. Once an attacker got to the information about Carl, this will also increase the probability that this attacker will reach all of Carl's friends. So, such a method can't preserve the user's privacy. Therefore, researchers extended the well-known K-anonymity [8] model, introduced to protect statistical data from the disclosure risk, to develop different privacy models of the graph according to various assumptions of an attacker's

background knowledge. Such as K-Degree [4], K-Neighborhood [9], [10], and K-Automorphism [11] model to prevent different types of structure-based re-identification attacks.

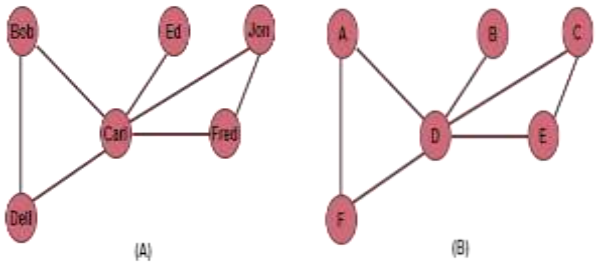

Fig. 1. An Example to show Naïve-anonymization of Graph Data, which (A) is the Primary Graph G and (B) is the Perturbed Version G* of G.

This paper assumes that structural information is the only information available to an attacker that can exploit it to carry out a re-identification attack. Assuming that the attacker is aware of the vertex degree of the target vertex. Researchers introduced many attempts to tackle such a case in previous studies by applying the *K*-Degree anonymity model. This model can distort networks structure by adding or deleting edges so that each vertex of the adjusted version is identical with at least $(K-1)$ vertex concerning vertex degree. However, this approach may cause a large distortion to the local structure of the primary graph. Thus, this distortion will harm the utility of data especially, when the anonymized data is used to meet analytical needs. The main reason behind this large distortion is that most existing anonymization algorithms, which are based on the edge modification approaches, don't take into consideration the concept of edge's relevance proposed in [12], which aim at maximizing data utility through keeping the important edges in the graph without any modifications.

In this paper, we introduce edge betweenness centrality measure [13] to highlight the most valuable edges in the graph and apply the *K*-Degree anonymity model only to edges with no or fewer betweenness values to preserve privacy and maximize the data utility, especially for clustering processes. Since edges with high betweenness values consider the most important knowledge for some popular community detection algorithms to discover community [14].

The remnant of this paper will be structured as follows, Section II discusses the literature review, Section III introduces the proposed Method, Section IV declares the results and evaluation, lastly, Section V highlights conclusions and the directions of the future works.

II. Literature Review

According to the previous works of literature [15] [16] on the anonymization of social networks, different anonymization approaches are categorized into three main groups: Edge modification-based anonymization approaches, clustering-based generalization, and differential privacy approaches [17].

Edge modification-based anonymization [4], [9]: these methods can anonymize the graph structure through modifying edges (adding and/or deleting) until reaching the desired value (*K*-anonymity). While some other methods suggest modifying the edges of the graph randomly.

Clustering-based generalization approaches [18]: these methods cluster nodes that are similar together (groups). Then each group will be generalized into an obscure cluster without any information about a specific individual. Although such methods succeeded in hiding the details about individuals, they fail in preserving the local graph structure of the social network. Because the graph structure is shrunk during the anonymization process. Consequently, these methods will not be eligible for analyzing the graph structure [19].

Differential privacy approaches: such methods seek for preserving user's privacy through imposing restrictions on the data release mechanisms; whereas the differentially private-based algorithms aim at providing statistical information about data without allowing direct access to the whole database. Consequently, such methods prevent a malicious attacker who can query the database from disclosing the target's identity.

In this paper, we focus on previous studies that addressed the anonymization problem through Edge modification-based approaches. Some authors concentrate on preserving the general structural properties of the anonymized network [20]–[22], While others are interested in preserving the community structure in the anonymized version [23], [24].

The authors in [25] compared the results of four algorithms, used for implementing *K*-degree anonymity, in terms of the information loss furthermore the data utility. These algorithms were introduced by different authors. The first one introduced the concept of *K*-degree anonymity in [4]. The second and third algorithms are EAGA and UMGA presented [26], [27] respectively. The last one, introduced in [28], which are based on the vertex addition method. They tested all algorithms using the same configurations. Each one follows its method for minimizing the changes performed on the graph structure. Their results showed that the UMGA scored the best results with all tested networks because it succeeds in minimizing the number of edges modified within the anonymization phase.

The authors in [12] propounded an efficient anonymization approach for creating a *K*-degree anonymized graph. They utilized the neighborhood centrality as a measure for assigning the most significant edges in the graph. They proved that preserving these edges during the anonymization process decreases the amount of information loss. At the same time, their method proved its efficiency in increasing the usefulness of the anonymized graph for evaluating the clustering process. Also, their algorithm achieves the highest results with less information loss compared to other popular anonymization algorithms.

The authors in [29] presented a new method to satisfy *K*-degree anonymity through node addition and edge set modification. Instead of adding nodes randomly, they gave the priority to the nodes with low betweenness centrality values to be modified. Their results proved that their approach could preserve APL, Closeness centrality, as well as nodes degree. But they didn't clarify how their proposed method achieves utility about the preservation of the anonymous graph's community structure.

The authors in [30] introduced a genetic *K*-degree anonymity method in two steps to enhance the preservation of the structural information in anonymized graphs. In the first step, they partitioned vertices of the graph and assigned a label for each vertex to show how many edges needed to be added to achieve the required *K*-degree anonymized sequence. Then, they identified the set of vertices that should be existed in each community. In the second step, within each community, a few edges were added between the vertices to modify the graph using a meta-heuristic algorithm [31].

III. The Proposed Method

In our proposed approach we seek to preserve the most impactful edges during the modification phase which in turn help us to limit the number of the modified edges. We present edge betweenness Centrality (EBC) measure to determine the most essential edges in the graph. Also, keeping these edges in the anonymized network will lead the suggested approach to optimize data utility for clustering analysis.

A. Overview

For undirected and unlabeled graph $G(V,E)$, where V describes the set of vertices, and E defines the edges set in the graph. Let DS defines the degree sequence of graph G, where DS is a term to describe the vector of elements, i.e. $DS = \{ d_{V1}, d_{V2}, \dots, d_{Vn} \}$. each element $d_{vi} \in Ds$ is an integer, whereas d_{vi} is the degree value of vertex vi and n is the number of elements (vertices).

Regarding the graph anonymization, Liu and Terzi introduced two essential definitions in [4] for satisfying the *K*-degree anonymity concept:

1) A degree sequence DS is described as *K*-anonymous when each distinct value $d_{vi} \in Ds$ appears not less than *K* times.

2) A graph $G(V,E)$ is known as a *K*-degree anonymous graph when the degree sequence of the graph G is *K*-anonymized. As shown in Fig. 2.

By considering the previous definitions, we introduce our enhancing approach to anonymize the graph as described in Fig. 3. Our approach goes through two main stages. The first one accepts the original graph and anonymized the degree sequence. After executing this stage and getting the anonymized degree sequence, the second stage starts to realize the anonymized graph G^*. Finally, the utility estimation of the anonymized graph version will be evaluated in the experimental results section by extracting the community structure for both the initial and anonymized version of the graph.

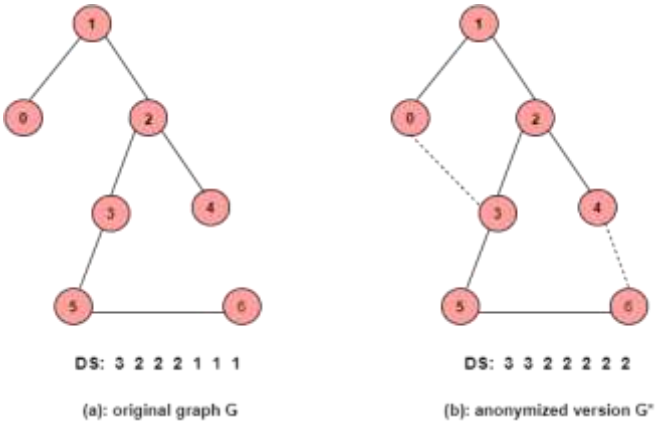

Fig. 2. Show an Example of Achieving 2-Degree Anonymity through Inserting some of Edges. (a): Original Graph G. (b): Anonymized Version G*.

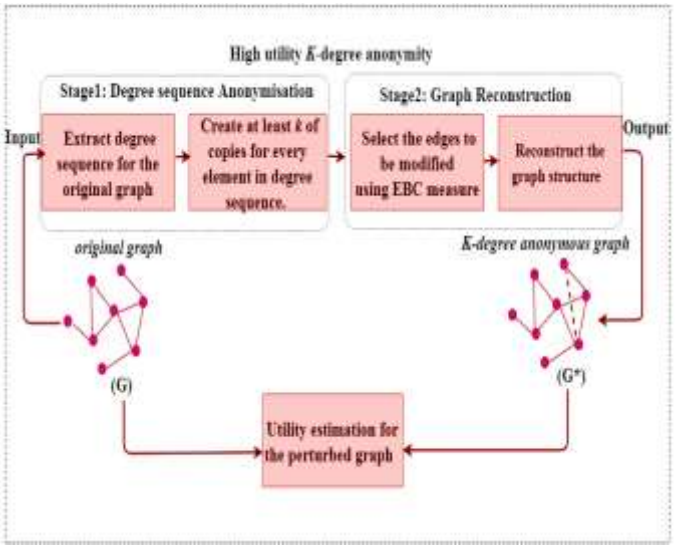

Fig. 3. Overview of Scheme.

B. Stage1: Degree Sequence Anonymization

Taking description (A) into consideration, we must adjust the values of DS to construct groups of at least K copies for each element. To satisfy this definition, we adopt the well-known univariate microaggregation technique proposed in [32] to perturb the degree sequence of the primary graph. The main objective is to get the optimal solution that decreases the distance between the primary degree sequence (DS) and the resulting K-anonymous sequence (DS^*), using the distance function:

$$dist\ (DS, DS^*) = \sum_{i=1}^{n} \left| d_{vi}^* - d_{vi} \right| \quad (1)$$

Our method starts by getting an optimal partitioning of graph vertices which is an order degree sequence that has been divided into several groups, then modifying the values of each group to achieve the required degree sequence that minimizes distance calculated by Eq.1. As stated in [32], given a directed graph $G_{k,n,}$ the optimal partition is defined as a set of groups which match the arcs of the shortest-paths that follows from the source vertex 0 to a vertex n of the graph. Each group that belongs to the optimal partition represents an arc that exists on the shortest path in the graph. The group size is in the range of K and $(2K-1)$ items. Then, to modify the values of each group of the optimal partition, we calculated the differences matrix as computed by [12]. Using this matrix, several solutions existed to satisfy *K*-degree anonymity. Finally, the Greedy method being selected to find the optimal solution among all possible ones using a probability distribution matrix.

C. Stage2: Graph Reconstruction

In this stage, we start to adjust the original graph according to the anonymous degree sequence DS^* resulting from the first stage. In our approach, the scope of modifications made is limited only to the set of edges, while the vertices set don't get any changes. We deploy three types of operations to modify the set of edges of the original version:

- Edge insertion operation is to link and include a new edge between two vertices vi, vj and it is denoted as $edge^{ins}(vi, vj)$.
- Edge deletion operation is to eliminate an existing edge between two vertices, denoted as $edge^{del}(vi, vj)$.
- Edge swap operation is used to switch between two edges, i.e., $e1(vi, vj)$ with $e2(vi, vf)$. It is referred as edge $^{swap}((vi, vj), (vi, vf))$.

Instead of specifying edges to be modified randomly, as with most previously used anonymization methods, we prefer to select the set of auxiliary edges that help to preserve the graph structure for the community analysis purpose. So, we utilize edge betweenness centrality measure for quantifying the most significant edges in the graph. The betweenness centrality of an edge e being estimated by computing the number of times that this edge exists on the shortest paths in each pair of graph vertices. It is computed as follows:

$$BC(e) = \sum_{s \neq t \in V} \sigma_{st}(e)/\sigma_{st,} \quad (2)$$

Where σ_{st} indicates the number of shortest-paths from a vertex $\boldsymbol{s}$ to a vertex $\boldsymbol{t}$ while $\sigma_{st}(e)$ is the number of the paths that go across $\boldsymbol{e}$.

Among all available edges to be adjusted, we choose edges with high betweenness values to be preserved during the modification process. These edges have more importance than others. Only edges with no or low betweenness values are allowed to be modified during the modification process.

D. Summary

Algorithm: High utility $\boldsymbol{k}$-degree anonymity algorithm

Input: A graph $G(V, E)$, and anonymity parameter $\boldsymbol{k}$.
Output: k-degree anonymous graph G^*.

Begin:
// elements sorted in a descending order.
1: DS ← construct degree sequence (G);
2: $G^*(V, E) = G(V, E)$.
3: DS^* ← anonymize degree sequence (G);
// show vertex set that needs to change its degree.
4: *while* G^* is not feasible *do*:
5: $DS_{diff} = (DS^* - DS)$;
6: S_{ops} ← Identify the operation type needed to satisfy the required degree: (edge ^{ins}, edge ^{del}, edge ^{swap});
7: *while* $S_{ops} \neq \{\}$ *do*:
8: EdgeList ← find the candidate edges to be modified;
9: EBC_List ← calculate betweenness centrality value for each edge (EdgeList, $G^*(V, E)$);
10: S_{aux_edges} ← EBC_List. *min_value* ();
11: run (ops, aux_edges, G^*);
// define new set of operations.
12: S_{ops} ← Identify the operation type needed to satisfy the required degree: (edge ^{ins}, edge ^{del}, edge ^{swap});
13: *end while*
14: *end while*
15: return G^*;

IV. Computational Results

In this section, we show the empirical results to assess the performance of our proposed algorithm. We will compare our method to the results of the two well-known approaches for *K*-degree anonymity. We change the value of *K* to vary from 2 to 10. The two methods are the KDA approach presented in [4] and the UMGA-NC approach proposed in [12]. We run all algorithms on the same dataset and the same configuration. Firstly, we show how far the structural properties of the graph can be conserved. Secondly, we measure how well our anonymization approach could preserve the community structure of the original graph.

A. Datasets and Environment

We test all algorithms on three real datasets which are unlabeled and undirected networks: these networks are Polbooks [33], American College football [13], and Jazz Musicians [34]. Table I shows the original properties of the three networks which include Diameter (D), Average path length (APL), Average Closeness (ACLN), Average betweenness (ABTW), and Transitivity (T). All experiments were tested on Google Colab on a PC with a 2.40 GHz i3 processor, 2 GB RAM, and a 228 GB hard disk running with Microsoft Windows 7 Ultimate. All experiments were implemented using python.

TABLE I. TESTED NETWORKS' PROPERTIES

	Polbooks	**American College football**	**Jazz Musicians**
$\|V\|$	105	115	198
$\|E\|$	441	613	2,742
D	7	4	6
$\overline{deg}$	8.40	10.661	27.697
APL	3.078	2.508	2.235
ACLN	0.329	0.399	0.457
ABTW	0.020	0.013	0.006
T	0.348	0.407	0.520
K	1	1	1

B. Assessment Measures

To assess the performance of our proposed approach compared to the others, we test four important measures that are used commonly in social network analysis. The four used measures are:

- *Average path length* (APL) is the average distance in the graph between every pair of vertices as described in Eq.3. Where V is the vertices set in the graph G, $d(u,w)$ is the shortest path length from vertex u to vertex w , and n is the vertices number in G.

$$APL(\mathrm{G}) = \frac{\sum_{u,w\epsilon V} d(u,w)}{\binom{n}{2}} \quad (3)$$

- *Closeness Centrality* (CLN) [35] is the Inverse of average distances to all reachable vertices. We calculate the Closeness of a vertex of u as follows:

$$CLN(u) = \frac{n}{\sum_{w\epsilon V} d(u,w)} \quad (4)$$

- *Betweenness Centrality* (BTW) [35] of a vertex u is specified as in Eq.5. $\sigma_{st}(u)$ indicate to the number of shortest-paths from the vertex $\boldsymbol{s}$ to $\boldsymbol{t}$ while $\sigma_{st}(u)$ is the number of the shortest-paths that go across u.

$$BTW(u) = \sum_{s\neq t\neq u\in V} \frac{\sigma_{st}(u)}{\sigma_{st}} \quad (5)$$

- *Transitivity* (T) is defined as the fraction of all triangles available in graph G. Available triangles are determined by triads number (two edges with a common vertex). We can compute the Transitivity of a graph G as:

$$T(G) = \frac{3(\text{number of triangles})}{\text{number of triads}} \quad (6)$$

To analyze the performance of our approach compared to the other two methods clearly, we evaluate the perturbation produced during the anonymization process of the four metrics listed above. As in Table II, we calculate mean absolute error (MAE) between the original and anonymized version of the tested networks over ten *K* levels as follows:

$$MAE(G, G^*) = \frac{\sum_{i=1}^{n} | g_i - {g_i}^* |}{n} \quad (7)$$

As ${g_i}^*$ is the value of the tested metric, e.g. (APL, $CC, \dots$), of the anonymized graph G^* at a particular level of *k*, g_i is the true value of the tested metric of the original graph G and n is the number of *K* levels.

C. Structural Analysis of the Perturbed Graph

In this section, we show the results of KDA, UMGA-NC, and our algorithm on the three networks listed in Table I. We calculate the four measures described previously for both the original graph and its anonymized version to show how much information is lost during the anonymization process. The actual metrics values of the original graph are constant for all different *K* values. They are represented by horizontal lines.

Fig. 4a, 5a and 6a show the average path length (APL) of the three anonymized networks as parameter *K* varies from 2 to 10. As we can see, the values of our proposed method are more similar to the actual ones than values of KDA, UMGA-NC, which means that lower information loss on APL.

Fig. 4b, 5b, and 6b refer to the average Closeness (ACLN) of the perturbed networks. All figures indicate that changes produced by our anonymization method on the average closeness also kept much closer to the real ones than existed by the two other methods.

Fig. 4c, 5c, and 6c describe the average node betweenness values. From the indicated figures, we note that our method could preserve the node betweenness values to become identical to the original values with varying anonymity parameter *K* in both football and Jazz Musicians networks. As for the Polbooks network, there are quite a few changes in the betweenness values.

Lastly, Fig. 4d, 5d, and 6d present the transitivity results on the three perturbed graphs. The performance of our proposed method comparing to the two other permutation methods isn't clear. We will quantify the performance of three permutation methods on transitivity obviously in Table II.

TABLE II. ERROR INDICATOR ON THE TESTED METRICS OVER 10 *k* LEVELS

Network	*Algorithm*	*APL*	*ACLN*	*ABTW*	*Transitivity*
Polbooks	*KDA*	0.349	0.042	0.003	0.023
	UMGA-NC	0.201	0.023	0.002	**0.014**
	Our Method	**0.134**	**0.015**	**0.001**	0.031
Football	*KDA*	0.017	0.003	**0.000**	0.012
	UMGA-NC	0.005	0.001	**0.000**	**0.005**
	Our Method	**0.002**	**0.000**	**0.000**	0.006
Jazz Musicians	*KDA*	0.064	0.011	0.004	0.020
	UMGA-NC	0.028	0.006	**0.000**	**0.014**
	Our Method	**0.019**	**0.002**	**0.000**	0.018

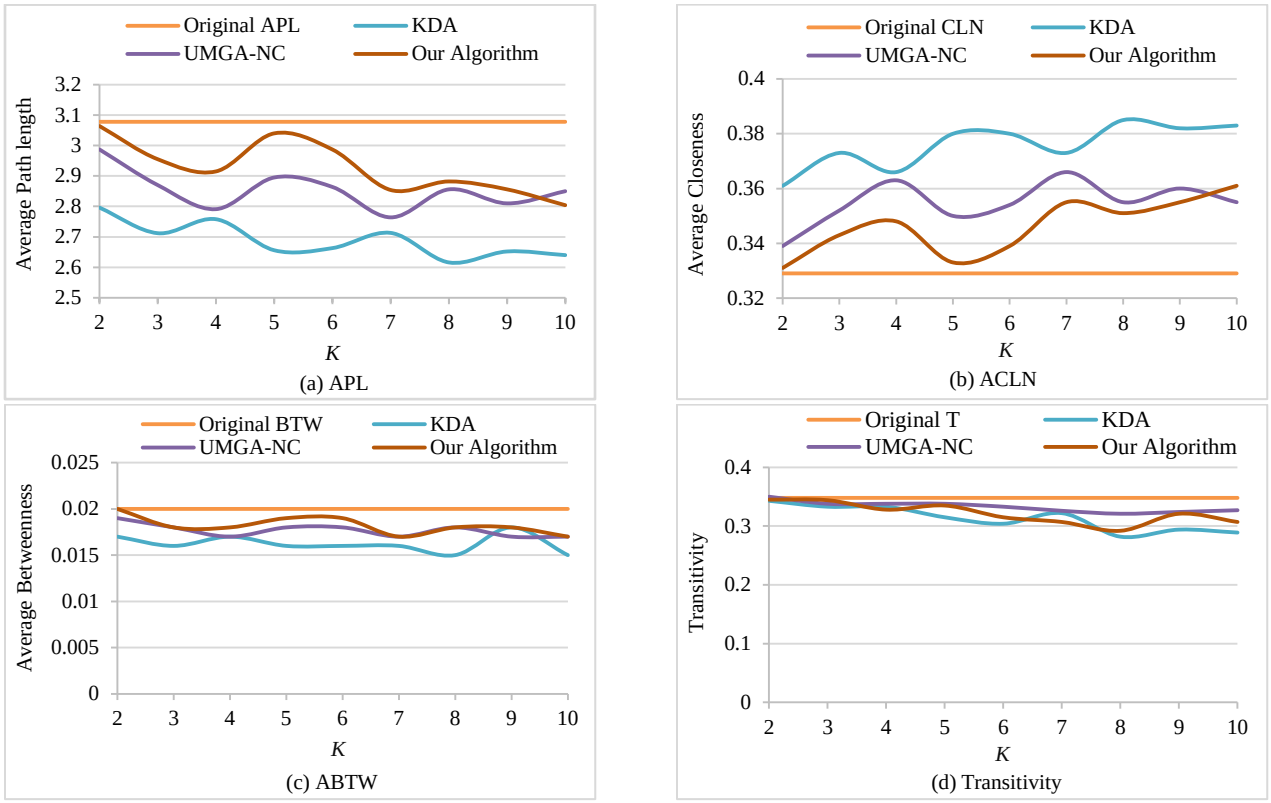

Fig. 4. Utilities of Polbooks Network for different K.

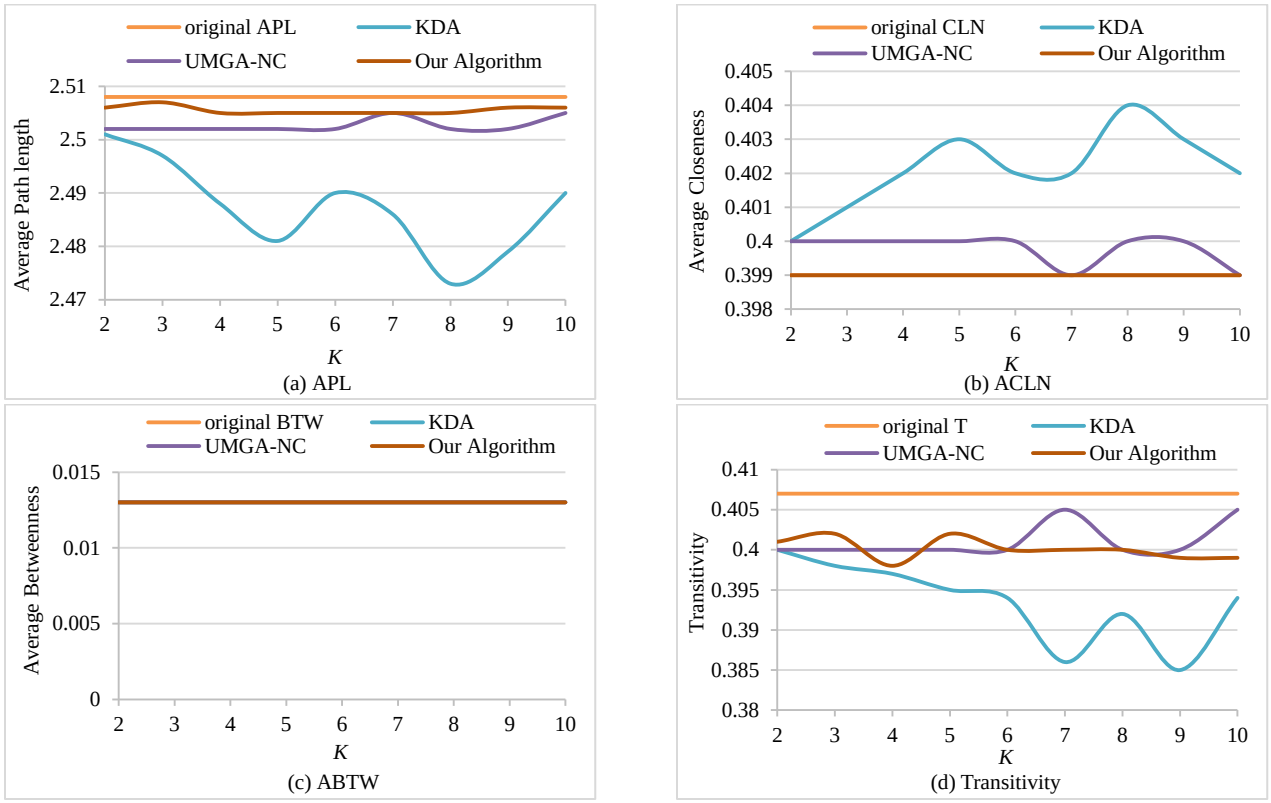

Fig. 5. Utilities of Football Network for different K.

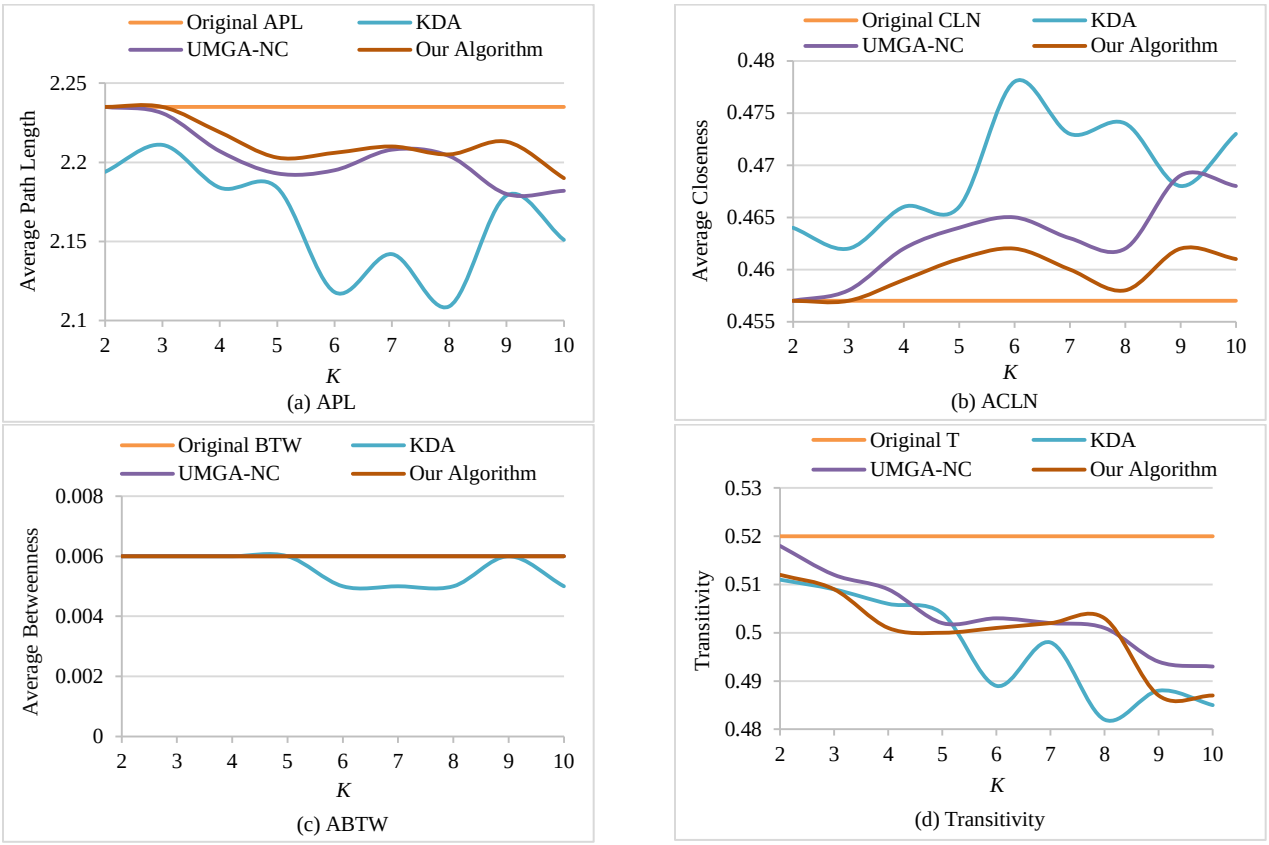

Fig. 6. Utilities of Jazz Musicians Network for different K.

We calculate the amount of error on the four tested metrics over 10 *K* levels as we referred to in Eq.7. As can be seen in Table II, our method gets the best results on the APL, ACLN, and ABTW except for the Transitivity results which are much affected in the anonymous graph. UMGA-NC method ranks first for the Transitivity metric on the three tested networks. Where the values of the mean computed error by the UMGA-NC method are lower than the ones obtained by both KDA and our method. As for KDA, our method achieves better results for Transitivity on both Football and Jazz Musicians.

D. Community Structure Preservation

Community detection algorithms are one of the most significant tasks for the processes of graph mining. This section will appreciate the utility of the perturbed graph for three different community algorithms. The three algorithms are (1) Girvan-Newman algorithm (*GN*) [13], which is is a hierarchical decomposition algorithm where edges deleted in descending order according to their edge betweenness scores. (2) Walktrap (*WT*) introduced in [36] is based on the concept of the random walk where the short random walks are likely to be kept in the same community. (3) Label Propagation (*LP*) proposed in [37], the main notion of the algorithm is to assign each vertex in a graph into a specific community, to which most of its adjacent vertices belong. For more details, see [38].

Using the networkX library, we extract community structure for both the original and the anonymized version of the three previous networks described in Table I. We use the f1-score measure [39] to assess the accuracy of our approach in preserving the actual community structure as described in Eq.8. This measure is used to test the similarity between the predicted communities set of the anonymized graph and the ground truth communities of the original version. We compute the f1-score values of *K*-anonymity for our algorithm and UMGA-NC using the three community algorithms. Then, we estimate the mean error on the f1-score over ten *K* levels. Table III presents the results.

$$f1-score = \frac{2\times recall\times precision}{recall+precision} \quad (8)$$

Whereas: $Precision = \frac{|C_P \cap C_T|}{|C_P|}$ and, $Recall = \frac{|C_P \cap C_T|}{|C_T|}$

Where C_T, is the vertices set that belong to the ground truth communities and C_P, denotes the set of vertices in the predicted communities produced by the community algorithm.

As shown in Table III, our method-EBC could present the lowest error on the tested networks using the three community algorithms. Consequently, a less information loss and better preservation for the community structure compared to UMGA-NC. Comparing the three community algorithms, The Girvan-Newman algorithm (*GN*) performs best on the three networks anonymized by our method. The reason behind this is that the Girvan-Newman algorithm (*GN*) is essentially based on the edge betweenness centrality values to detect communities, and our approach could preserve this metric well during the anonymization process.

TABLE III. MEAN F1-SCORE ERROR OVER 10 *K* LEVELS

	UMGA-NC			***Our Method-EBC***		
Network	***GN***	***WT***	***LP***	***GN***	***WT***	***LP***
Polbooks	0.058	0.073	0.344	0.029	0.030	0.192
Football	0.014	0.044	0.038	0.004	0.004	0.012
Jazz Musicians	0.042	0.442	0.044	0.021	0.309	0.035

V. CONCLUSION

Most of the previous works seek to anonymize graph data, regardless of the role of some edges that have proven their usefulness in analyzing the graph data. In this paper, we focus on optimizing the utility of an anonymized graph by minimizing the changes made to these edges. For this reason, we introduce the edge betweenness measure to identify and preserve the most relevant edges in the graph during the modification operation. Those edges, if modified, will cause large distortion to the local structure of the anonymized graph.

We perform an analysis using many structural metrics and different community algorithms on the graph structure. The final results proved that our method achieves the best performance as less information is lost comparing with other popular anonymization algorithms. Besides that, it can provide better preservation of the community structure compared to other similar methods.

In our future work, we plan to enhance the performance of our proposed approach. We intend to implement our algorithm on big data platforms to utilize graph computation systems such as GraphX on the Apache Spark platform and to test our proposed method on large graphs.

REFERENCES

[1] Nettleton, “Data mining of social networks represented as graphs,” Computer Science Review. 2013.

[2] Y. Mengmeng, Z. H. U. Tianqing, Z. Wanlei, and X. Yang, “Attacks and countermeasures in social network data publishing,” ZTE Communications, vol. 14, no. S0, pp. 2–9, 2019.

[3] C. Watanabe, T. Amagasa, and L. Liu, “PrIvacy risks and countermeasures in publishing and mining social network data,” 2011.

[4] K. Liu and E. Terzi, “Towards identity anonymization on graphs,” 2008.

[5] E. Zheleva and L. Getoor, “Privacy in social networks: A survey,” in Social network data analytics, Springer, pp. 277–306, 2011.

[6] P. Wicks, M. Massagli, J. Frost, C. Brownstein, et al., “Sharing health data for better outcomes on patientslikeme,” Journal of Medical Internet Research, 2010.

[7] L. Backstrom, C. Dwork, and J. Kleinberg, “Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography,” 2007.

[8] L. Sweeney, “k-anonymity: A model for protecting privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 557–570, 2002.

[9] B. Zhou and J. Pei, “Preserving privacy in social networks against neighborhood attacks,” 2008.

[10] B. Zhou and J. Pei, “The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks,” Knowledge and Information Systems, 2011.

[11] L. Zou, L. Chen, and M. Tamer Özsu, “K-automorphism: A general framework for privacy preserving network publication,” Proceedings of the VLDB Endowment, 2009.

[12] 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, 2017.

[13] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” Proceedings of the National Academy of Sciences of the United States of America, 2002.

[14] P. Bedi and C. Sharma, “Community detection in social networks,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2016.

[15] K. Macwan and S. Patel, “Privacy Preservation Approaches for Social Network Data Publishing,” in Studies in Computational Intelligence, 2021.

[16] B. Ouafae, R. Mariam, L. Oumaima, and L. Abdelouahid, “Data Anonymization in Social Networks,” 2020.

[17] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” Foundations and Trends in Theoretical Computer Science, 2013.

[18] E. Zheleva and L. Getoor, “Preserving the privacy of sensitive relationships in graph data,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008.

[19] A. Campan, Y. Alufaisan, and T. M. Truta, “Preserving communities in anonymized social networks,” Transactions on Data Privacy, 2015.

[20] E. Sargolzaei, M. J. Khazali, and F. Keikha, “Privacy preserving approach of published social networks data with vertex and edge modification algorithm,” Indian Journal of Science and Technology, 2016.

[21] C. Sun, P. S. Yu, X. Kong, and Y. Fu, “Privacy preserving social network publication against mutual friend attacks,” 2013.

[22] T. M. Truta, A. Campan, and A. L. Ralescu, “Preservation of structural properties in anonymized social networks,” 2012.

[23] J. Vadisala and V. Kumari, “Anonymized Social Networks Community Preservation,” International Journal of Advanced Computer Science and Applications, 2017.

[24] H. Wang, P. Liu, S. Lin, and X. Li, “A local-perturbation anonymizing approach to preserving community structure in released social networks,” 2017.

[25] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, “A Summary of $$$$-Degree Anonymous Methods for Privacy-Preserving on Networs,” in Advanced Research in Data Privacy, Springer, pp. 231–250, 2015.

[26] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, “Evolutionary algorithm for graph anonymization,” arXiv preprint arXiv:1310.0229, 2013.

[27] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, “An algorithm for k-degree anonymity on large networks,” in 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), pp. 671–675, 2013.

[28] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh, “Why Waldo befriended the dummy? k-Anonymization of social networks with pseudo-nodes,” Social Network Analysis and Mining, 2013.

[29] S. Hamzehzadeh and S. M. Mazinani, “ANNM: A New Method for Adding Noise Nodes Which are Used Recently in Anonymization Methods in Social Networks,” Wireless Personal Communications, 2019.

[30] S. Rajabzadeh, P. Shahsafi, and M. Khoramnejadi, “A graph modification approach for k-anonymity in social networks using the genetic algorithm,” Social Network Analysis and Mining, 2020.

[31] V. K. Sihag, “A clustering approach for structural k-anonymity in social networks using genetic algorithm,” 2012.

[32] S. L. Hansen and S. Mukherjee, “A polynomial algorithm for optimal univariate microaggregation,” IEEE Transactions on Knowledge and Data Engineering, 2003.

[33] V. Krebs, “polbooks | Miscellaneous Networks | Network Repository,” 2001. http://networkrepository.com/polbooks.php (accessed Apr. 08, 2021).

[34] P. M. GLEISER and L. Danon, “Community Structure in Jazz,” Advances in Complex Systems, 2003.

[35] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, Anonymizing social networks. 2007.

[36] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” 2005.

[37] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 2007.

[38] N. R. Smith, P. N. Zivich, L. M. Frerichs, J. Moody, and A. E. Aiello, “A Guide for Choosing Community Detection Algorithms in Social Network Studies: The Question Alignment Approach,” American Journal of Preventive Medicine, 2020.

[39] G. Rossetti, L. Pappalardo, and S. Rinzivillo, “A novel approach to evaluate community detection algorithms on ground truth,” 2016.