An Uncertain Graph Method Based on Node Random Response to Preserve Link Privacy of Social Networks

Jun Yan[1,2], Jiawang Chen[1], Yihui Zhou[1], Zhenqiang Wu[1], and Laifeng Lu[3*]
[1] School of Computer Science, Shaanxi Normal University
Xi'an, 710119 - China
[e-mail: yanrongjunde@snnu.edu.cn, jiawangc@snnu.edu.cn, zhouyihui@snnu.edu.cn, zqiangwu@snnu.edu.cn]
[2]School of Mathematics and Computer Applications, Shangluo College,
Shangluo, 72600 - China
[3]School of mathematics and statistics, Shaanxi Normal University
Xi'an, 710119 - China
[e-mail: lulaifeng@snnu.edu.cn]
*Corresponding author: Laifeng Lu

Received September 8, 2022; revised August 12, 2023; accepted December 12, 2023; published January 31, 2024

Abstract

In pace with the development of network technology at lightning speed， social networks have been extensively applied in our lives. However, as social networks retain a large number of users' sensitive information, the openness of this information makes social networks vulnerable to attacks by malicious attackers. To preserve the link privacy of individuals in social networks, an uncertain graph method based on node random response is devised, which satisfies differential privacy while maintaining expected data utility. In this method, to achieve privacy preserving, the random response is applied on nodes to achieve edge modification on an original graph and node differential privacy is introduced to inject uncertainty on the edges. Simultaneously, to keep data utility, a divide and conquer strategy is adopted to decompose the original graph into many sub-graphs and each sub-graph is dealt with separately. In particular, only some larger sub-graphs selected by the exponent mechanism are modified, which further reduces the perturbation to the original graph. The presented method is proven to satisfy differential privacy. The performances of experiments demonstrate that this uncertain graph method can effectively provide a strict privacy guarantee and maintain data utility.

Keywords: social networks, link privacy, uncertain graph, differential privacy, random response

This research was supported by the National Natural Science Foundation of China (62001273, 61962033) the Fundamental Research Funds for the Central Universities (No.2019CBLY004 and No.GK201903091), and the Scientific and Technological Project of Shangluo (No. 2021-C-0004).

http://doi.org/10.3837/tiis.2024.01.009 **ISSN : 1976-7277**

1. Introduction

Nowadays, with the rapid development of network technology, online social networks have been widely applied in our lives. For example, we are able to do many things on online social networks, such as instant messaging, shopping, mobile payment, live streaming, hotel booking and so on, which make our daily lives more and more convenient [1]. More importantly, since a large number of user behaviors are recorded on online social networks, online social networks store a large amount of data, which can be analyzed and mined by many companies to provide users with better services. However, these data contain a great deal of sensitive information about personal social relations, salary, financial transaction behavior, disease, time and space activities, religious beliefs, political opinions, etc., which can lead to privacy leakage in the case of illegal use[2]. For instance, as the biggest social network platform, FaceBook has fallen into many scandals in recent years, in which a lot of user data is illegally used and a larger number of individual privacy information is breached[3]. Therefore, it is necessary to provide sufficient privacy preserving for online social networks to preserve individual privacy.

In particular, large amounts of data in social networks are often represented as graphs, which can be utilized in many graph analysis tasks, such as information propagation, link prediction, community detection, etc[4]. At the same time, malicious attackers can also use graph analysis methods to mine personal privacy information in these graph data. To preserve these graph data, many modification methods have been designed, which can be grouped into three categories: (1) Edge and Vertex modification methods that modify(add, delete or switch) edges/nodes in a graph. (2) Generalization methods that group vertices and edges into super-vertices and super-edges. (3) Uncertain graph methods that inject uncertainty into the edge of the graph. In the first method, it is easy to preserve the original graph by randomly adding or deleting a node or edge, but this random modification method has insufficient data utility. In addition, to improve data utility, *K*-anonymity methods have been widely applied to preserve the sensitive nodes and edges. In *K*-degree-anonymity method, each node connects *k* nodes with the same degree, which results in the probability of identifying each node being less than 1/*k*[5]. Furthermore, the *l*-diversity method[6], *t*-close method[7] and k-anonymity with edge selection method[8] have been presented to resist all kinds of attacks. In the second method, for improving privacy preserving and anti- attack capability, the generalization methods cluster similar nodes together to generate super-nodes, which preserve the link relationships between nodes in this super-node, and super-edges which combine the edges between these super-nodes[9]. In the third method, by injecting uncertainty semantics into the graph, the uncertain graph method can preserve the sensitive relationship between nodes in the graph while keeping the similar structure of the original graph to achieve notably better data utility.

As a special graph modification method, the uncertain graph method is composed of two steps to generate an uncertain graph. In particularity, the first step modifies the original graph by adding/deleting some edges while maintaining its structure as much as possible, then the second step injects uncertainty into the modified graph to get an uncertain graph which preserves the privacy of the original graph. However, there are some disadvantages to uncertain methods. For example, although the (k,ε1)-obfuscation method has better data utility[10], it is not able to resist the rounding attack. In the random walk method[11], the structure of the original graph is modified by deleting many edges, which leads to insufficient data utility. For the UDGP method[12], the differential privacy can improve the privacy preserving of edges, but it is vulnerable to attacks based on eliminating edge probability.

Because differential privacy can provide a rigorous privacy guarantee against attacks based on background knowledge, many differential privacy based methods have been proposed to preserve the graph structure data since differential privacy was developed by Cynthia Dwork[13]. Especially, the random response with differential privacy has better privacy preserving than the centralized differential privacy. To address these disadvantages in the uncertain methods and accomplish the preservation of link privacy, a novel uncertain graph method based on node random response is devised to generate an uncertain graph. In this method, to improve privacy preserving, the random response is adopted to modify edges on nodes and the node differential privacy injects the noise on edges to generate an uncertain graph. In addition, to minimize the perturbation to the original graph, the original graph is decomposed to get many sub-graphs and some sub-graphs with a larger number of edges selected by the exponent mechanism are modified. Therefore, the proposed uncertain graph method can preserve the link privacy of social networks while maintaining data utility.

The major contributions in this paper are shown as follows:

(1) A general framework to generate an uncertain graph is proposed, which can achieve the trade-off between privacy preserving and data utility. In this framework, after the original graph is decomposed into many sub-graphs, some sub-graphs with a large number of edges are obtained through the exponent mechanism. After that, all obtained sub-graphs are modified. In the end, an uncertain graph is generated by combining all the sub-graphs.

(2) An uncertain graph method based on node random response is presented to provide sufficient privacy preserving for the link privacy of social networks. In this method, the random response mechanism is adopted to modify the edges and the uncertainty is injected on edges through node differential privacy.

(3) The experiments are performed on synthetic and real data sets to demonstrate the effectiveness of our method regarding privacy preserving and data utility. Compared with other methods, the result demonstrates that our method can preserve the link privacy of the original graph with a high level of data utility.

The organization of this paper is described as follows: Section 2 concentrates on the graph modification methods and differential privacy based methods. Some basic knowledge and definitions are introduced in Section 3. Section 4 demonstrates the model of the devised uncertain method, describes the algorithms in detail and explicitly analyzes the privacy guarantees of this method. The experiments are shown in Section 5 to evaluate the proposed method. Finally, the conclusion and future work are described in Section 6.

2. Related Work

For preserving the sensitive information in the social network, many graph modification methods had been widely adapted for privacy preserving before the social network was released. In general, these methods include edge and node modification methods, generalization methods and uncertain graph methods.

Among the existing edge and node modification methods, due to insufficient data utility caused by the random perturbation, X.Ying[14] developed two algorithms that could preserve the original graph while maintaining its spectral properties as much as possible. In [15], only the most important edges were protected to achieve a better trade-off between privacy preserving and data utility. As a useful privacy preserving method, k-degree anonymity had been usually employed to preserve social networks through anonymity graphs. J.Casas in [16] developed a k-degree anonymity method that anonymized the degree sequence of a graph by using the univariate micro-aggregation to achieve the desired data utility. On the basis of this method, the edge relevance was considered to design a k-degree anonymity method that minimized edge perturbation to enhance data utility. To resist structural attacks, the k-isomorphism method was introduced in [17] to preserve social networks, which achieved strong anonymity while maintaining the data utility. In addition, by using graph similarity detection to get subgraphs, [18] proposed a subgraph K+-isomorphism method that satisfied k-isomorphism while reducing information loss.

Different from edge and node modification methods, generalization methods focused on how to generate super-nodes and super-edges, which could hide the details of individuals. In [19], firstly, the mutual information of each node was calculated according to the physical data theory, then the nodes with high mutual information were selected as key nodes. In the end, the key nodes were used as core nodes to cluster similar nodes to generate a clustered graph. To simultaneously protect the characteristics of nodes and communities, F.Yu[20] proposed a clustering algorithm that adopted some perturbation strategies to reduce privacy leakages while maintaining data utility.

In order to preserve social networks with better data utility, Boldi in [10] presented a (k,ε1)-obfuscation method which generated an uncertain graph by injecting uncertainty to the edges of social networks. Due to the insufficiency of obfuscation, the uncertain graph generated by this method was easy to be re-identified through the rounding attack. Compared with the (k,ε1)-obfuscation method, the Rand-Walk method [11] was able to provide strong privacy preserving with insufficient data utility. In [21], Nguyen proposed a Maximum Variance method for a better trade-off between privacy and utility, which utilized the quadratic programming method to assign the probability value of edges. Based on the above work, [22] devised a generalized obfuscation model that could preserve the degree of nodes unchanged and get an uncertain graph by using uncertain adjacency matrices. To provide strong privacy preserving for link privacy of social networks, J.Hu in [12] utilized edge-differential privacy to design an uncertain graph method that also met the requirements of data utility. [23] introduced a method based on the triadic closure to generate an uncertain graph that was suitable for small social networks.

In comparison with the graph modification methods, it is noted that differential privacy had some advantages that could stop attacks based on background knowledge and provide rigorous mathematical proof [24]. Owing to these advantages, many differential privacy methods had been developed to preserve social networks since C. Dwork created differential privacy. In general, these methods usually adopt differential privacy to preserve specific sensitive statistics of graphs and publish synthetic graphs. When publishing the degree distribution of a graph, Day in [25] proposed two node differential privacy methods which used aggregation and cumulative histogram respectively to reduce the error caused by the noise. To release the node strength histogram with fewer errors, [26] designed an edge differential privacy based method that aggregated the original histogram of the graph by using the sequence-aware and local density based clustering approaches. In [27], because of the sub-graph based attacks, Nguyen introduced a method that perturbed all *k*-vertices that linked some sub-graphs by adding noise to some edges. In addition, other statistics in social networks including triangle counts, node centrality and shortest path had been preserved by many differential privacy based methods when they were published [28,29]. Except for the statistical data, the differential privacy had also been employed to obtain a synthetic graph that could preserve the original graph. In [30], V.Karwa used a graphical degree partition of the original graph and perturbed it by differential privacy to get a synthetic graph. In addition, [31] developed an LDPGen method which clustered structurally-similar users together through multiple iterations to construct a synthetic graph in a distributed environment.

Particularly, it is well known that randomized response is an input perturbation algorithm that perturbs the input value by a probability mechanism. For preserving the answer to a sensitive question elicited in the surveys, such a design based on randomized response has been widely used and studied [32]. For example,[33] controlled the statistical disclosure by using the randomized response when publishing data in the form of contingency tables. In addition, it has also been used to release network data preserved by differential privacy in [34], which offered rigorous privacy guarantees for the original network. In this paper, a randomized response mechanism under differential privacy was designed to modify the edges of a graph and construct a perturbed graph to preserve the original graph.

3. Preliminaries

In this section, some definitions used throughout the paper are introduced. In particularity, a social network is abstracted as a simple undirected graph *G*=(*V*, *E*), where *V* denotes nodes and *E* represents edges.

Definition (Uncertain graph[10]).

Let a graph *G*=(*V*, *E*), a function *P: EP* →[0, 1] , which assigns probabilities to edges in *E'*, we can get an uncertain graph *G'* =(*V*, *E'*, *EP*), where *E'* is attained by modifying the *E*, and *EP* represent the probabilities of edges. Compared with a graph *G*, the uncertain graph G' has the same nodes as *G* and has different edges from *G*. In a deterministic graph, the probabilities of all edges are 1.

Definition 2 (Neighboring graph[12]).

For two graphs G_a=(V_a, E_a) and G_b=(V_b, E_b), compared with G_b, if G_a has one different node, $|V_a| = |V_b| + 1$, E_b in E_a, G_a and G_b are neighboring graphs.

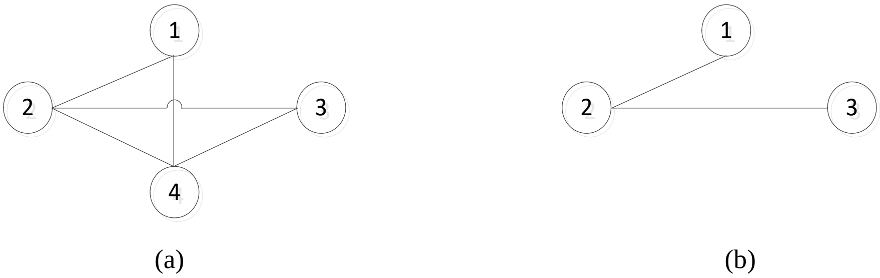


Fig. 1. An example of neighboring graphs

As illustrated in **Fig. 1**, **Fig. 1(a)** has a different node and three different edges compared with **Fig. 1(b)**, so **Fig. 1(a)** and **Fig. 1(b)** are neighboring graphs.

In addition, if there is one different edge between G_a and G_b, $|E_a| = |E_b| + 1$, G_a and G_b are also neighboring graphs.

Definition 3 (Sensitivity[12]).

Let *F* be a sequence of queries: *G* → *E*, the sensitivity of *F* is:

$$\Delta f = \max_{G_a, G_b} \left\| F(G_a) - F(G_b) \right\|_1 \tag{1}$$

The Hamming distance is used to calculate thesensitivity of *F*. If G_a is different from G_b by one node, the sensitivity of *F* is d_{max}, where the d_{max} is the maximum degree of nodes in the graph.

Definition 4 (Differential Privacy[12]**).**

Let $\varepsilon \geq 0$, a randomized algorithm *M* satisfies ε-differential privacy if for any two neighboring graphs G_a and G_b and all *S* ⊆ *Range(M)*, the following holds:

$$\Pr[M(G_a) \in S] = e^{\varepsilon} \times \Pr[M(G_b) \in S] \qquad (2)$$

where G_a and G_b are neighbors, ε denotes a privacy preserving level. To achieve ε-differential privacy for graphs, two ways including the Laplace mechanism and the Exponential mechanism have been adopted to perturb the outputs of *M*.

Definition 5 (Laplace Mechanism[12]).

Let *F* be a sequence of queries: *G* ⟶ *E*, and *M* is a randomized algorithm applied on *G*, there is the following :

$$M(G) = F(G) + lap(\Delta f / \varepsilon) \qquad (3)$$

where $lap(\Delta f / \varepsilon)$ denotes the Laplace noise with μ =0, $b = \Delta f / \varepsilon$.

The Laplace mechanism is the way that adds Laplace noise on *F*(*G*) to ensure the algorithm *Z* can satisfy ε-differential privacy.

In addition, Eq. (4) describes the Laplace noise distribution.

$$L(x) = 1/2b * \exp(-|x - \mu|/b) \qquad (4)$$

where μ represents a position parameter, *b* is a scale parameter and *x* denotes a random variable.

Definition 6 (Exponential Mechanism[13]).

For a data set *D*, let *r* be the output of function *F*, where *r* ∈ *R*, the function *U:* (*D*, *t*) ⟶*R* is the scoring function of *r* on *D*, and the global sensitivity of the scoring function is *Δ U*. If the random algorithm *A* (*D*, *u*, *R*) is proportional to $\exp(\frac{\varepsilon U(D,r)}{2\Delta U})$ to select and output *r* ∈ *R*, the random algorithm *A* is said to satisfy ε-differential privacy. The implementation process of this algorithm is called the exponential mechanism

Definition 7 (Randomized Response[32]).

The randomized response mechanism is defined as follows:

$$P(y_i = k \mid x_i = j) = P_{ij} \qquad (5)$$

where x_i is an input which equals *j*, the probability to output that y_i equals *k* is P_{ij}. When the value ranges of *j* and *k* belong to {0,1}, $i \subset [1, N]$, *N* is the number of the inputs.

The design matrix P_m of the 2-dimensional randomized response is defined as follows:

$$P_m = (\begin{matrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{matrix})$$

where P_{00} indicates the probability that a random output equals 0 when a real input is 0, P_{01} represents the probability that a random output is 1 when the real input equals 0, where P_{00} and P_{01} in [0,1]. At the same time, P_{10} represents the probability that a random output equals 0 when the real input is 1, P_{11} is the probability that a random output is 1 when a real input equals 1, where P_{10} and P_{11} in [0,1]. Particularly, as the sum of probabilities of each row is 1, the design matrix can be simplified to

$$P_m = \begin{pmatrix} p_{00} & 1-p_{00} \\ 1-p_{11} & p_{11} \end{pmatrix}$$

Definition 8 (Randomized Response satisfying ε-Differential Privacy).

Given a parameter ε , if max $\{P_{00}/P_{10}, P_{00}/P_{01}, P_{01}/P_{11}, P_{10}/P_{11}\} < e^{\varepsilon}$, the randomized response scheme based on design matrix P_m in definition 7 will achieve ε-differential privacy.

Definition 9 (Post-Processing[12]).

Assuming a randomized algorithm *M* that satisfies ε-differential privacy, when a graph *G* is input to *M* , the output of the algorithm *M* is *G'*, which can preserve the graph *G*. Let *N* be an arbitrary randomized mapping, when *N* is applied on *G'* to get *G"*, the algorithm $M \circ N : G \rightarrow G^{"}$ satisfies ε -differential privacy.

Definition 10 (Parallel composition properties[13]).

Let a sequence of algorithms be $\{A_1, A_2, ..., A_n\}$, and assuming that each algorithm A_i is ε_i-differential privacy. When these algorithms are utilized respectively to preserve *n* disjoint subsets of the database *D*, then the combination processing of all algorithms satisfies maxε_i differential privacy, and it is called the parallel composition properties of differential privacy.

4. Framework and Method

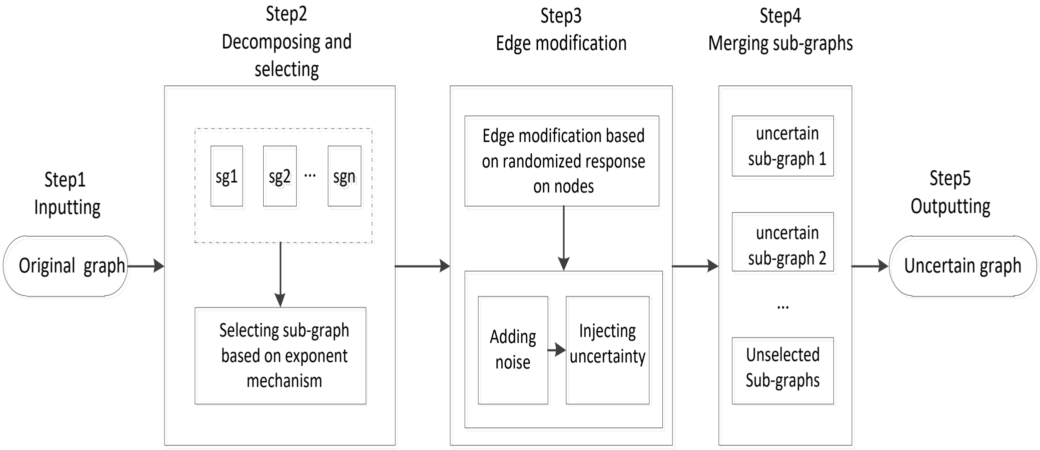


Fig. 2. A general framework to generate an uncertain graph

4.1 A general framework

A general framework proposed consists of five steps, three of which are the main steps of the framework. These three steps are decomposing and selecting, edge modification and merging sub-graphs. The detail of this framework is described in the following.

As shown in **Fig. 2**, step 1 inputs an original graph that denotes a social network. In step 2, the original graph is divided into many sub-graphs by using the Louvain algorithm, then the exponent mechanism selects some sub-graphs with a large number of edges, so the minimal edge modification can be realized on the original graph under a given privacy budget. Then, step 3 utilizes the randomized response and the differential privacy to modify each selected sub-graph through edge modification. In this step, the randomized response is applied on nodes to modify each sub-graph. In particular, this process only adds or deletes edges between one node and its neighbor nodes and second-order adjacent nodes. After that, the node differential privacy adds the Laplace noise on edges and the post-process mechanism injects the uncertainty into these edges through. Finally, a set of uncertain sub-graphs is generated. In step 4, all uncertain sub-graphs and the unselected sub-graphs are merged to generate an uncertain graph. In the end, step 5 outputs an uncertain graph that achieves differential privacy preserving for link privacy.

In summary, a general framework based on node random response is devised, which can preserve the link privacy of the social network while obtaining sound data utility.

4.2 Methods and Algorithms

4.2.1 *UGNRR* (Uncertain graph based on node random response) method

As shown in the general framework, the key work of this framework is to generate an uncertain graph that can provide strong privacy preserving for the link privacy of the original graph. Therefore, the *UGNRR* method based on this framework is proposed, which includes three algorithms, *SGEM*(Selecting sub-graph based on exponent mechanism) algorithm, *EMNR*(Edge modification based on node random response mechanism)algorithm and *UNDP* (Uncertain graph based on node differential privacy) algorithm. In this method, the *SGEM* algorithm utilizes the exponent mechanism to get a set of sub-graphs with a large number of edges. For each sub-graph in this set, the *EMRN* algorithm modifies the edges of each sub-graph according to random response, then the *UDPM* algorithm transforms each modified sub-graph into an uncertain sub-graph through node differential privacy. At last, this method generates an uncertain graph that achieves privacy preserving for link privacy of the original graph.

Algorithm 1. *The UGNRR algorithm*

```
Input: a undirected graph G, the privacy budget ε
Output: an uncertain graph Gu
1. a set of sub-graphs $S_s$ ← decomposing an undirected graph G
2. a set of selected sub-graphs $S_{sub}$ ← SGEM algorithm( $S_s$, ε)
3. a set of $S_{Gu}$= {} .
4. for $S_{Gi}$ in $S_{sub}$:
5.         $Sn_{Gi}$ ← EMNR algorithm( $S_{Gi}$, ε)
6.         $Sm_{Gi}$ ← UNDP algorithm( $Sn_{Gi}$, ε)
7.         $S_{Gu}$ adding $Sm_{Gi}$
8. $S_r$ ← $S_s$ - $S_{sub}$
9. Gu ← merge( $S_{Gu}$ , $S_r$)
10. Return an uncertain graph Gu
```

To achieve the proposed method, the *UGNRR* algorithm is presented above. In line 1, an inputted undirected graph is decomposed into a set of sub-graphs S_s. Line 2 selects a set of sub-graphs S_{sub} from S_s through the *SGEM* algorithm. For each sub-graph S_{Gi}, it is dealt with by two algorithms from line 4 to line 7. Line 5 adds and deletes edges by the *EMNR* algorithm, then the *UNDP* algorithm generates an uncertain sub-graph in line 6. Lastly, the two sub-graph sets S_{Gu} and S_r are merged to generate an uncertain graph in line 9.

4.2.2 *SGEM*(Selecting sub-graph based on exponent mechanism) algorithm

After an original graph is decomposed into many subgraphs, the edge modification is used to modify these subgraphs. In *UGNRR* method, the number of edges in each sub-graph is used to denote the size of this sub-graph. Due to the different sizes of these subgraphs, the edge modification will perturb them differently. When the same perturbation of the edge modification is added to each sub-graph, the smaller the size of the sub graph, the larger the perturbation is.To reduce the perturbation, some small-size sub-graphs are deleted and the perturbation is added on the remained sub-graphs. In this way, there are two kinds of perturbation. One is caused by the edge modification, the other is brought by the deleted sub-graphs. In order to gain the minimum perturbation, the exponent mechanism is utilized to select some larger size sub-graphs. In this exponent mechanism, the Laplace noise denotes the perturbation of edge modification.

Given an undirected graph *G*, the *GN* algorithm divides it into *n* sub-graphs. For each sub-graph S_i, the Sw_i, which is the sum of the edges in S_i, denotes the size of the sub-graph S_i. Then, there is a sequence W_G , described as[Sw_1, Sw_2, ..., Sw_n].

Algorithm 2. *The SGEM algorithm*

Input: a set of sub-graphs S_s, the privacy budget ε

Output: a set of selected sub-graphs S_{sub}

```
1.$n \leftarrow |S_s|$
2.$W_G \leftarrow S_s$
3.for $m$ in $n$ :
4.        scoring function
          $-U(G,m)=\sqrt{\sum_{i=m+1}^{n}|s_{w_i}|^2}+\sqrt{2*m}*\frac{\Delta f}{\varepsilon}$
5.  selecting m with probability
          $P_r(m)\propto \exp(-\frac{\varepsilon U(G,m)}{2*\Delta U})$
6.  $W_{Gm} \leftarrow$ truncating $W_G$ with $m$
7.  a set of sub-graphs $S_m \leftarrow$ selecting sub-graphs from $S_s$ according to $W_{Gm}$
8.  $S_{sub} \leftarrow S_m$
9.  Return $S_{sub}$
```

After sorting the W_G from larger to small, we select the first *m* units from it and add noise to them. Then, we will get the *Error*(W_G), which is illustrated as follows.

$$Error(W_G) = DE(W_G) + LE(W_G)$$

where *DE*(W_G) represents the error caused by the deleted units, *LE*(W_G) is the Laplace noise added.

$$DE(W_G) = E(\sqrt{\sum_{i=m+1}^{n} |Sw_i|^2})$$

$$LE(W_G) = E(\sqrt{\sum_{i=1}^{m} lap(\Delta f / \varepsilon)^2})$$

$$DE(W_G) + LE(W_G) = \sqrt{\sum_{i=m+1}^{n} |Sw_i|^2} + \sqrt{2*m} * \frac{\Delta f}{\varepsilon}$$

Here, a query function is f : $f(G) \to W_G$

$$\Delta f = \left| f(G) - f(G^{'}) \right| = \left| W_G - W_{G^{'}} \right| = d_{\max}$$

where Δf is the sensitivity of a query function f , d_{max} is the maximum degree of nodes in *G*, and *G* and G' are neighboring graphs with one different node .

In order to gain a minimum value of *Error*(W_G), the exponent mechanism selects a best threshold *m* which can be used to select some sub-graphs. Thus, a scoring function *U* is set up:

$$U(G,m) = \sqrt{\sum_{i=m+1}^{n} |Sw_i|^2} + \sqrt{2*m} * \frac{\Delta f}{\varepsilon}$$

In this algorithm, the node differential privacy is applied to realize strong privacy preserving. Therefore, the ΔU is:

$$\Delta U = U(G,m) - U(G^{'},m) = \Delta RE + \Delta LE$$

the ΔU is:

$$\Delta RE \le \max\left| \sqrt{\sum_{i=m+1}^{n} |Sw_i|^2} - \sqrt{\sum_{i=m+1}^{n} |Sw_i^{'}|^2} \right|$$

$$\le \quad \max\left| \sum_{i=m+1}^{n} |Sw_i| - \sum_{i=m+1}^{n} |Sw_i^{'}| \right| \le d_{\max}$$

$$\Delta LE = \Delta f$$

$$\Delta U = \Delta RE + \Delta LE \le 2d_{\max}$$

The probability to select the threshold *m* is

$$p_r(m) = \frac{\exp(-\frac{\varepsilon * U(G,m)}{2\Delta U})}{\sum_{i=1}^{n} \exp(-\frac{\varepsilon * U(G,i)}{2\Delta U})}$$

Then the best threshold *m* is used to truncate the W_G. Finally, a set of sub-graphs S_m is obtained, which can be utilized to realize the minimal noise perturbation in the original graph.

Line 1 is the number of sub-graphs and line 2 is a sequence W_G that records the size of each sub-graph. From line 3 to line 5, the exponent mechanism gains a threshold *m*. According to *m*, line 6 truncates the sequence W_G and line 7 selects a set of sub-graphs S_m from the set of the sub-graphs S_s. In the end, line 8 gets a set of sub-graphs S_{sub}.

4.2.3 *EMNR* (Edge modification based on node random response mechanism) algorithm

In the set of sub-graphs S_{sub}, the random response applied on nodes is used to add and delete edges in each sub-graph. To maintain data utility, this algorithm only adds and deletes between this node and its adjacent nodes and the second-order adjacent nodes. In order to add some edges in S_{Gi}, an edge sequence is created firstly, in which each edge links node *i* and one of its second-order adjacent nodes, and each edge in it is assigned a value 0. Then, we input the value of each edge into the random response mechanism. If the value of one edge becomes 1, this edge will be added to this sub-graph. In addition, when deleting edges from the graph, another edge sequence is generated, in which each edge links node *i* and one of its adjacent nodes, and each edge in it is assigned a value 1. After entering the value of one edge into the random response mechanism, this edge will be deleted from the sub-graph S_{Gi} if its value becomes 0. Since deleting edges will destroy the structure of the graph, only one-*i*th of selected edges will be deleted, where *i* usually take 3. Finally, the edge modification modifies each sub-graph S_{Gi} by the node random response mechanism. The detail of the *EMNR* algorithm is described in Algorithm 3.

Algorithm 3. *The EMNR algorithm*

Input: S_{Gi}, the privacy budget ε

Output: Sn_{Gi}

```
$S_{pGi} \leftarrow S_{Gi}$
a set of nodes $N_t \leftarrow S_{pGi}$
for $i$ in $N_t$:
    a set of nodes $N_{2i} \leftarrow$ selecting the second order adjacent nodes of $i$
    an edge sequence $S_e \leftarrow$ linking node $i$ to the nodes in $N_{2i}$
    $S_{ae}$= {}
    for $e_i$ in $S_e$:
        input $j$=0
        $P_{01}$=1/1+$e^{\varepsilon}$
        then if $j$=1 with prob $P_{01}$, adding $e_i$ into $S_{ae}$
    adding edges in $S_{ae}$ into $S_{pGi}$
    an edge sequence $S_{ce} \leftarrow$ edges connected to node $i$
    $S_{de}$= {}
    for $e_i$ in $S_{ce}$:
        input $j$=1
        $P_{10}$=1/1+$e^{\varepsilon}$
        then if $j$=1 with prob $P_{10}$, adding $e_i$ into $S_{de}$
    deleting one-$i$th of edges in $S_{de}$ from $S_{pGi}$
$Sn_{Gi} \leftarrow S_{pGi}$
Return $Sn_{Gi}$
```

4.2.4 *UNDP* (Uncertain graph based on node differential privacy) algorithm

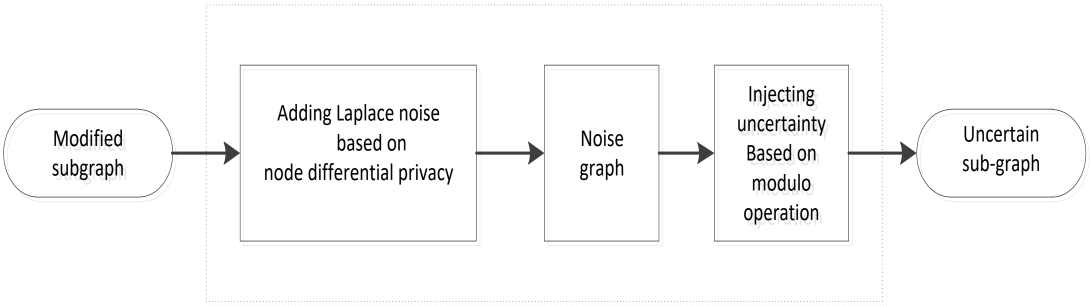

Modified subgraph
Adding Laplace noise based on node differential privacy
Noise graph
Injecting uncertainty Based on modulo operation
Uncertain sub-graph

Fig. 3. The *UNDP* algorithm

To generate an uncertain graph, the *UNDP* algorithm is shown in **Fig. 3**. As shown in **Fig. 3**, to generate an uncertain sub-graph, the *UNDP* algorithm contains three steps. First of all, according to the Laplace mechanism, the Laplace noise is added on each edge of a graph $Sn_{Gi.}$ In this process, the node differential privacy is applied, which provides better privacy preserving than edge differential privacy. After that, each edge of a graph Sn_{Gi} becomes a noised edge with a noise value. Therefore, a graph Sn_{Gi} is transformed into a noised subgraph. Finally, according to the principle of post-processing, the noise value of each edge is calculated based on the modulo operation. In this algorithm, the modulo operation is to modulo 1 then taking the remainder, so the result of this operation is in [0,1]. Thus, this result is regarded as a probability value and it is assigned on each edge of Sn_{Gi}. Therefore, an uncertain graph Sm_{Gi} is generated. The detail of *UNDP* algorithm is described in Algorithm 4.

Algorithm 4. *The UNDP algorithm*

```
Input: $Sn_{Gi,}$ the privacy budget ε
Output: an uncertain subgraph $Sm_{Gi}$
the maximum degree $d_{max}$ in $Sn_{Gi} \leftarrow \Delta f$
a Laplace noise sequence $E_n \leftarrow Lap(\Delta f$ /the privacy budget ε)
for $e_i$ in $Sn_{Gi}$
    $e_i \leftarrow E_{ni}$
    $p_i \leftarrow$ the modulo operation($E_{ni}$)
    if $p_i < 0.5$
        $p_i = 1 - p_i$
    $e_i \leftarrow p_i$
Return an uncertain subgraph $Sm_{Gi}$
```

4.3 The analysis of method

Theorem : The *UGNRR* method satisfies ε -differential privacy.

Proof: In this method, the exponent mechanism that satisfies differential privacy is adopted in the *SGEM* algorithm, the *EMNR* algorithm uses a randomized response with differential privacy and the *UNDP* algorithm utilizes the node differential privacy. Therefore, these three algorithms all satisfy differential privacy. To achieve minimal edge modification to the original graph under a given privacy budget, the *SGEM* algorithm selects some sub-graphs from the original graph, which satisfies differential privacy. Then, according to the parallel composition properties principle of differential privacy, the process that the *EMNR* algorithm and the *UNDP* algorithm are applied to generate uncertain sub-graphs also satisfies differential privacy. In the end, the process of merging all uncertain sub-graphs satisfies the post-processing . In summary, the *UGNRR* algorithm satisfies ε -differential privacy.

5. Experimental Analysis

The developed method is evaluated in this section. First, some experiment data sets are introduced. Then, the developed method is analyzed from different aspects. Finally, the proposed method is also compared with other uncertain graph approaches.

5.1 Data set

In our experiments, two kinds of experiment data are utilized, which include synthetic data sets and real data sets. The synthetic data sets are obtained from ER graphs, which contain 500 and 1000 nodes. The real data sets contain Face-book data with 4039 nodes and 63731 nodes, and Enron email network with 36692 nodes, which is from [35].

To evaluate the proposed method, (k,ε1)-obfuscation method, Rand-Walk method and *UGDP* method are adopted for comparison. All simulation experiments run on an HP computer, which has an Intel Core i5-8500 with 3.00GHz and 12GB memory. For programming, Python is used on the Microsoft Windows 7 operating system.

5.2 Privacy evaluation

5.2.1. Privacy measurement

When a graph is converted into an uncertain graph, there is a certain gap between them which can be measured by the editing distance. Because the edge in uncertain graphs is uncertain, the expectation of editing distance is introduced to measure the gap between an original graph and an uncertain graph, which also can be used to evaluate preserving privacy.The larger the *EED*, the better privacy preserving.

It is well-known that the definition of edit distance between two deterministic graphs G_1, G_2 is: $D(G_1,G_2)=|E_1\setminus E_2|+|E_2\setminus E_1|$

According to the formula above, the expected edit distance between the uncertain graph *G*’’ and the deterministic graph *G* is:

$$EED\left[D(G,G^{''})\right]=\sum_{G_1^{'}}P_r(G_1^{'})D(G,G_1^{'})=\sum_{e_i\in G}(1-p_i)+\sum_{e_i\notin G}p_i$$

where G_1’is sampled from *G*’’, *Pr*(G_1’) indicates the probability of obtaining G_1’ from the uncertain graph *G*’’.

In *UGNRR* algorithm, when we get an uncertain graph *Gu*, the expected edit distance between *Gu* and the graph *G* is:

$$EED\left[D(G,Gu)\right]=EED\left[D(G,G^{'})\right]+EED\left[D(G^{'},Gu)\right]$$

where *G’* is obtained by the *MERN* algorithm, and *Gu* is generated by the *UNDP* algorithm.

$$EED\left[D(G,G^{'})\right] = e_k$$

where e_k equals the edit distance between two deterministic graphs *G* and *G'*, which is calculated by the following formula: $e_k = |E_a| + |E_d|$

where $|E_a|$ denotes the number of edges which are added in *G*, where $|E_d|$ is the number of edges which are deleted from *G*.

Then there are no edges added and removed in the *UNDP* algorithm,, thus, the expected edit distance between *Gu* and *G'* is

$$EED\left[D(G^{'},Gu)\right] = \sum_{e_i \in G^{'}} (1 - p_i)$$

where e_i belongs to the edges set of *Gu*, p_i is the probability of the edge e_i.

The expectation of editing distance(*EED*) between *Gu* and *G* is shown as follows:

$$EED\left[D(G,Gu)\right] = e_k + \sum_{e_i \in G^{'}} (1 - p_i)$$

5.2.2. Privacy analysis

To evaluate the different uncertain graph algorithms in privacy preserving, the *EED* is used. The greater the *EED*, the better privacy preserving this uncertain graph method achieves. all data sets are executed 10 times by the proposed method and other methods to gain the average results.

In the comparative experiments, the parameter of three methods is shown in **Table 1**. In (*k*,ε1)-obfuscation method, the obfuscation level *k* belongs to 10, 20, the tolerance parameter ε l equals 0.1, the multiplier factor *c* is 1 and the white noise *q* is equal to 0.01. In Rand-Walk method, the parameter *t* denotes the size of the noise. In addition, the privacy budget ε is in [0.2, 0.5, 1, 1.5, 2] in the *UGNRR* method.

Table 1. The *EED* values of four methods in differential data sets

method	**Parameter**	**ER 500**	**ER 1000**	**Facebook 4039**	**Enron 36692**	**Facebook 63731**
UGNRR	ε =0.2	21687	78986	76894	322733	678579
UGNRR	ε =0.5	20143	78176	76243	321648	676793
UGNRR	ε =1	18687	77496	75876	321087	675156
UGNRR	ε =1.5	18078	77153	75567	320675	674765
UGNRR	ε = 2	17243	76902	75354	317582	673456
(*k*,εl)-obfuscation	*k*=10	13243	43512	48934	197865	457783
(*k*,εl)-obfuscation	*k*=20	13654	43876	49263	198243	458495
Rand-Walk	*t*=5	24754	81654	80432	357784	704356
Rand-Walk	*t*=10	24421	81243	79894	356465	703218
UGDP	ε =0.2	17023	67889	62785	257863	572742
UGDP	ε =0.5	16593	67254	61523	256890	571465
UGDP	ε =1	16298	66865	60231	256135	569243
UGDP	ε =1.5	15753	66734	59643	255764	568786
UGDP	ε = 2	15597	66452	59132	255365	568215

The result of *EED* values can be seen in **Table 1**. In **Table 1**, the *EED* values in the *UGNRR* method are shown from the first to the five row, where the *EED* increases as the value of ε decreases, which means that the privacy preserving of the *UGNRR* method becomes stronger. For example, in the FaceBook data set with 4039 nodes, when ε is 2, the value of *EED* is 75354. As ε ascends to 0.5, the value of *EED* rises to 76243, which means that the privacy preserving of *UGNRR* method is improved. Additionally, as the number of nodes in the original graph increases, the *EED* of *UGNRR* method rises simultaneously, which indicates that the *UGNRR* method is able to provide privacy preserving for the different social networks. For instance, in table1, when ε is 1, it is clear that the *EED* of *UGNRR* method increases from 18687 to 675156 as the number of nodes changes from 500 to 63731, which illustrates this method can be applied in different social networks.

As shown in **Table 1**, the *EED* of (*k*,ε1)-obfuscation method is shown from the sixth row to the seventh row while the rows from the eighth to the ninth indicate the *EED* values of Rand-walk method. In addition, the detail of the *UGDP* method is described in the rest rows. Compared with the other three methods in the same data set, the value of *EED* obtained by using *UGNRR* method is larger than that through (*k*,ε1)-obfuscation method and *UGDP* method, meanwhile it is smaller than that through Rand-Walk method. For example, in the FaceBook data set with 4039 nodes, when ε is 0.5, the value of *EED* in the *UGNRR* method is 76243, which is larger than that in the (*k*,ε1)-obfuscation method with *k*=20 and that in the *UGDP* method while being less than that in the Rand-Walk method with *t*=10. In particular, compared with the *UGDP* method, the results show that the edge modification and node differential privacy applied in the *UGNRR* method take effect on the value of *EED*. Therefore, according to the definition of *EED*, it is clear that the *UGNRR* method can provide stronger privacy preserving than (*k*,ε1)-obfuscation method and the *UGDP* method, but it is weaker than Rand-Walk method.

5.3. Utility evaluation

5.3.1. Utility metrics

In order to evaluate the data utility, the *NE*, *AD* and *DV* are used in our experiments. Due to the uncertainty of edges in an uncertain graph, the degree of a node in an uncertain graph is the expected degree which is equal to the sum of probabilities of its adjacent edges. Therefore, the definitions the *NE*, *AD* and *DV* are shown as follows:

$$d_v = \sum p(i,j) \qquad NE = {}^{1}\!/_{2}\sum\nolimits_{v\in V} d_v$$

$$AD = {}^{1}\!/_{n}\sum\nolimits_{v\in V} d_v \qquad DV = {}^{1}\!/_{n}\sum\nolimits_{v\in V} (d_v - AD)^2$$

Then, several structural measures are adopted. The first one is the diameter (S_{Diam}) which denotes the maximum distance among all path-connected pairs of nodes. The second measure is the average distance(S_{APD}) which is the average shortest distance among all path-connected pairs of nodes.

Furthermore, the Utility(function) defined as follows is utilized to measure the data utility of each method. The greater the *Utility*, the better the data utility of this method.

$$Utility = (1 - \frac{|UV - RV|}{RV}) \times 100\%$$

where *UV* is the graph metrics in uncertain graphs achieved by different methods, *RV* is the real metrics in the original graphs.

Finally, to compare the *UGNRR* method with other three methods in the data utility, the error on one graph metric is used, which is described as follows:

$$\Delta q(G,Gu)=\left|q(G)-q(Gu)\right|$$

where q represents one graph metric.

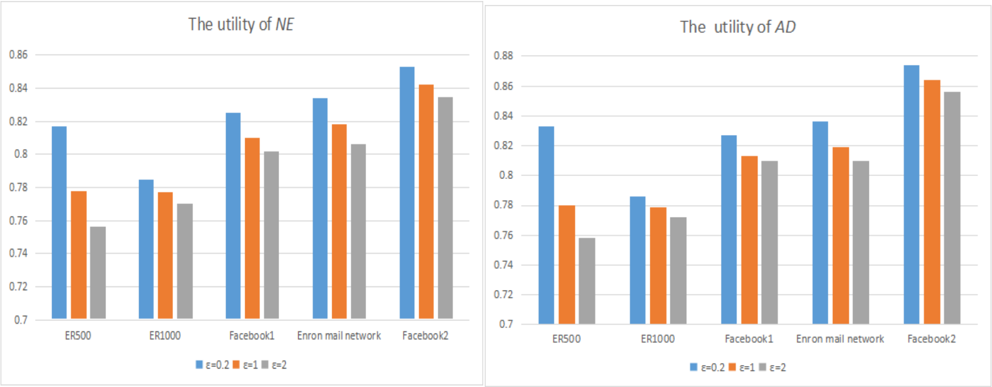
5.3.2. Utility analysis

To evaluate the data utility of the uncertain graph method, the experimental results are obtained by averaging the results 10 times and taking the final value. **Table 2** illustrates the graph metrics in the original graph and the *UGNRR* method, **Table 3** shows the graph metrics in the original graph, (k,ε1)-obfuscation method and Rand-walk method while **Table 4** demonstrates the graph utility metrics in the *UGDP* method.

As shown in **Table 2**, the value of *NE* in five data sets decreases as the ε rises, so does the value of *AD*. For instance, in the Facebook data set with 4039 nodes, the value of *NE* descends from 72846 to 70895 with the ε changing from 0.2 to 2, while the value of *AD* decreases from

Table 2. The metrics in the *UGNRR* method

data sets	metrics	original network	ε =0.2	ε =1	ε =2
ER graph 500	*NE*	24844	20631	19328	18787
ER graph 500	*AD*	99	82.52	77.31	75.14
ER graph 500	*DV*	2607	3220.41	3016.23	2863.86
ER graph 500	S_{Diam}	4	2.13	2.26	2.32
ER graph 500	S_{APD}	1.80	1.38	1.55	1.62
ER graph 1000	*NE*	99902	78425	77648	76954
ER graph 1000	*AD*	199	156.85	155.29	153.91
ER graph 1000	*DV*	8376	11283.26	9642.62	9524.32
ER graph 1000	S_{Diam}	4	2.32	2.63	2.72
ER graph 1000	S_{APD}	1.80	1.46	1.62	1.86
Facebook 4039	*NE*	88234	72846	71532	70895
Facebook 4039	*AD*	44	36.01	35.42	35.10
Facebook 4039	*DV*	3262	4789.76	4203.89	4084.23
Facebook 4039	S_{Diam}	4	2.67	2.85	2.96
Facebook 4039	S_{APD}	3	2.03	2.13	2.17
Enron 36692	*NE*	183831	153452	150375	148243
Enron 36692	*AD*	10	8.36	8.19	8.08
Enron 36692	*DV*	1328	2167.32	1887.43	1763.78
Enron 36692	S_{Diam}	4	2.73	2.92	3.06
Enron 36692	S_{APD}	33.9	23.5	26.2	27.4
Facebook 63731	*NE*	817092	697090	688765	682896
Facebook 63731	*AD*	25	21.87	21.61	21.43
Facebook 63731	*DV*	1785	4389	3565	3198
Facebook 63731	S_{Diam}	4	2.12	2.65	2.72
Facebook 63731	S_{APD}	1.32	1.02	1.08	1.11



(a) The *Utility* of *NE* **(b)** The *Utility* of *AD*

Fig. 4. The *Utility* of *NE* and *AD* in *UGNRR* method

36.01 to 35.10. In addition, the value of *DV* descends from 4789.76 to 4084.23, while the S_{APD} rises to 2.17. In the *UGNRR* method, the smaller ε, the more edges are modified in the original graph, so the greater the value of *NE* and *AD*. On the contrary, the larger ε, the fewer edges are modified, thus the value of *DV* becomes smaller and the S_{APD} gets close to that of original graph. Therefore, the *UGNRR* method can provide sufficient data utility regardless of the privacy budget ε .

Then the Utility is used to evaluate the data utility of UGNRR method. As shown in **Fig. 4 (a)**, the maximum *Utility* of *NE* is 85%. In **Fig. 4 (b)**, the highest *Utility* of *AD* can reach 87%, the lowest is 75%, so the average *Utility* of *AD* is about 81%. According to the results in **Table 2**, in the Facebook data set with 4039 nodes, the highest *Utility* of S_{Diam} is about 74%, while that of S_{APD} is 72%. Especially, the highest *Utility* of S_{APD} can reach 84% in the Facebook data set with 63731 nodes. Therefore, the data utility of *UGNRR* method is feasible.

Table 3. The metrics in $(k, \varepsilon 1)$-obfuscation method and Rand-walk method

data sets	**metrics**	**original network**	$(k,\varepsilon 1)$ **k=10**	$(k,\varepsilon 1)$ **k=20**	**Rand-Walk t=5**	**Rand-Walk t=10**
ER graph 500	*NE*	24844	20910.52	20910.88	12416.62	12608.66
ER graph 500	*AD*	99	83.64	83.65	49.66	48.44
ER graph 500	*DV*	2607	4393.93	4391.24	649.02	668.30
ER graph 500	S_{Diam}	4	3	3	2	2
ER graph 500	S_{APD}	1.80	1.83	1.84	2.26	2.30
ER graph 1000	*NE*	99902	76975.76	77142.11	50175.43	50326.13
ER graph 1000	*AD*	199	157.95	156.28	100.35	100.65
ER graph 1000	*DV*	8376	17794.66	17750.49	2585.74	2602.80
ER graph 1000	S_{Diam}	4	3	3	1.60	1.62
ER graph 1000	S_{APD}	1.80	1.82	1.82	2.23	2.36
Facebook 4039	*NE*	88234	86284.81	86253.19	45178.14	45383.70
Facebook 4039	*AD*	44	42.72	42.71	22.37	22.74
Facebook 4039	*DV*	3262	3245.78	3246.13	704.29	705.23
Facebook 4039	S_{Diam}	4	4	4	5.7	5.9
Facebook 4039	S_{APD}	3	3.29	3.32	4.68	4.69

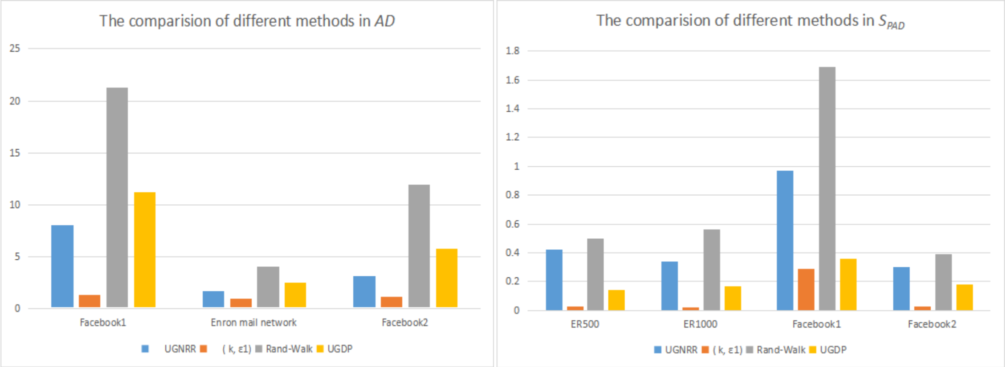
Enron 36692	*NE*	183831	183761.76	183730.92	100215.88	99815.90
Enron 36692	*AD*	10	9.05	9.06	5.96	5.95
Enron 36692	*DV*	1328	1328.33	1328.33	387.37	396.34
Enron 36692	S_{Diam}	4	4	4	5.2	5.4
Enron 36692	S_{APD}	33.9	29.2	29.2	16.2	16.7
Facebook 63731	*NE*	817092	816286.26	816278.78	425702.09	424627.74
Facebook 63731	*AD*	25	23.86	23.81	13.12	13.04
Facebook 63731	*DV*	1785	1764.44	1764.44	493.18	498.01
Facebook 63731	S_{Diam}	4	3	3	1.51	1.43
Facebook 63731	S_{APD}	1.32	1.29	1.29	1.72	1.71

Furthermore, Δq is utilized to compare *UGNRR* method with (k,*ε*1)-obfuscation method, Rand-walk method and the *UGDP* method in data utility. In the Facebook data set with 63731 nodes, the *NE* of the original graph is 817090, the *NE* obtained by the *UGNRR* method is 697090(*ε*=0.2) while the *NE* of the other three methods is 816286(*k*=10), 425702(*t*=5), 612833 (*ε*=0.2) respectively. Thus, the value of the Δq of *NE* obtained by the *UGNRR* method is larger than that in (*k*,ε1)-obfuscation method , but it is less than that in the *UGDP* method

Table 4. The metrics in the *UGDP* method

data sets	metrics	original network	ε =0.2	ε =1	ε =2
ER graph 500	*NE*	24844	18584.36	18593.00	18599.10
ER graph 500	*AD*	99	74.33	74.37	74.39
ER graph 500	*DV*	2607	2218.41	2216.23	2203.86
ER graph 500	S_{Diam}	4	2.80	2.91	2.94
ER graph 500	S_{APD}	1.80	1.94	1.88	1.85
ER graph 1000	*NE*	99902	74925.09	74948.40	74946.77
ER graph 1000	*AD*	199	149.85	149.89	149.90
ER graph 1000	*DV*	8376	7936.28	7990.28	7904.30
ER graph 1000	S_{Diam}	4	2.70	2.73	2.75
ER graph 1000	S_{APD}	1.80	1.97	1.96	1.95
Facebook 4039	*NE*	88234	66174.94	66186.07	66198.39
Facebook 4039	*AD*	44	32.76	32.77	32.87
Facebook 4039	*DV*	3262	1555.22	1556.06	1554.83
Facebook 4039	S_{Diam}	4	2.96	3.01	3.06
Facebook 4039	S_{APD}	3	3.63	3.56	3.45
Enron 36692	*NE*	183831	137860.14	137893.07	137911.36
Enron 36692	*AD*	10	7.51	7.51	7.52
Enron 36692	*DV*	1328	799.00	798.01	797.86
Enron 36692	S_{Diam}	4	3.21	3.24	3.26
Enron 36692	S_{APD}	33.9	28.2	28.6	28.8
Facebook 63731	*NE*	817090	612833.04	612890.90	612896.75
Facebook 63731	*AD*	25	19.23	19.23	19.23
Facebook 63731	*DV*	1785	926	925	924

Facebook 63731	S_{Diam}	4	2.76	2.81	2.85
Facebook 63731	S_{APD}	1.32	1.14	1.15	1.15



(a) The comparison of methods in *AD* **(b)** The comparison of methods in S_{APD}

Fig. 5. The comparison of different methods

and the Rand-walk method. The result indicates that the *UGNRR* method is not better than the (k,ε1)-obfuscation method in the data utility, but it is better than the *UGDP* method and the Rand-walk method. Additionally, the details of the Δq of other graph metrics are described in **Fig. 5**, where **Fig. 5 (a)** shows the Δq about *AD* in different methods, while **Fig. 5 (b)** demonstrates the Δq about S_{APD}. According to the results, (k,ε1)-obfuscation method has better data utility than the *UGNRR* method, while the *UGNRR* method is better than Rand-walk method. In addition, the *UGNRR* method is better than the *UGDP* method in some graph metrics, such as *AD*.

5.4. Computational complexity evaluation

Given a social network G=(V, E), where the number of nodes V is n and the number of edges E is m. In *UGNRR* method, the Louvain algorithm is adopted to decompose G into k sub-graphs $G_s(V_s, E_s)$, where $|V_s|=n_s$ and $|E_s|=m_s$. As *UGNRR* method consists of three main steps, the computational complexity O(x) of *UGNRR* method is the total execution time of these three steps. For step 1, G is decomposed into k sub-graphs through the Louvain algorithm, then *SGEM* algorithm is used to select k_t sub-graphs, the computational complexity is O($n$log$n$)+O(k). For step 2, *EMRN* algorithm and *UNDP* algorithm are utilized to gain k_t uncertain sub-graphs, so the computational complexity is k_{t*}(O($n_{s*}d_{max}^2$)+O(m_e)), where d_{max} is maximum degrees of nodes and m_e is the number of edges of each modified sub-graph. For step 3, as all sub-graphs are merged into an uncertain graph, the operation is within constant time and the computational complexity is O(1). Therefore, it is clear that the total computational complexity is O($n$log$n$)+O(k)+k_{t*}(O($n_{s*}d_{max}^2$)+O(m_e))+O(1). Especially, without considering the low order of magnitude, *UGNRR* method finally has the computational complexity O($n$log$n$)+k_{t*}(O($n_{s*}d_{max}^2$)+O(m_e)).

In other methods, *UGDP* method has the computational complexity O(m), while the computational complexity O(x) of (k,ε1) method and Rand-walk method is O((1+|c|)$_*m$) and O(n) respectively. Compared with *UGDP* method, after using O($n$log$n$) to decompose the original graph, *UGNRR* method spends k_{t*}O($n_{s*}d_{max}^2$)+O(m_e) to get an uncertain graph, which is smaller than O(m). In addition, the computational complexity of *UGNRR* method to get an uncertain graph is also smaller than (k,ε1) method and larger than Rand-walk method. Therefore, as the computational complexity O($n$log$n$) is feasible to decompose the original graph, $O(n\log n)+k_{t*}(O(n_{s*}d_{max}^{2})+O(m_e))$ of *UGNRR* method is viable to generate an uncertain graph.

In summary, the results of experiments show that the *UGNRR* method can not only provide sufficient privacy preserving, but also maintain data utility when it is applied in practice.

6. Conclusion

In the conclusion, you can reiterate the main points of the paper, but do not duplicate the abstract as a conclusion. With the increasing attention to individual privacy, many graph modification methods and differential privacy methods have been widely adopted to preserve the graph structure data in social networks, which contain personal sensitive link privacy. As a special useful graph modification method, the uncertain graph methods provide effective privacy preserving while maintaining data utility. To improve the privacy preserving of uncertain methods, an uncertain graph method based on node random response is developed, which can provide stronger privacy preserving than other uncertain graph methods. In particular, the random response is utilized to modify the original graph and the node differential privacy is applied to inject uncertainty on edges. In addition, to maintain data utility, after the original graph is decomposed into many sub-graphs, some sub-graphs with a larger number of edges are selected through the exponent mechanism and are modified by the edge modification. In particular, the edge modification only adds and deletes edges between the node and its neighbor nodes and second-order adjacent nodes in each sub-graph. According to the properties of differential privacy, the proposed uncertain graph method satisfies differential privacy. Meanwhile, the experiment results indicate that the proposed method owns better privacy preserving while attaining sound data utility. Therefore, the developed uncertain graph method can be widely applied to preserve the link privacy of social networks.

In the future, although the presented method achieves a better balance between privacy preserving and data utility, whether it can be applied to complex networks, such as dynamic graphs and directed graphs, is our next work.

References

[1] S.R.Sahoo, B.B.Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” *Applied Soft Computing*, vol.100, Mar.2021. Article(CrossRef Link)

[2] A.K.Jain, S.R.Sahoo, J.Kaubiyal, “Online social networks security and privacy: comprehensive review and analysis,” *Complex & Intelligent Systems*, vol.7, no.5, pp.2157-2177, Jun. 2021. Article (CrossRef Link)

[3] J. Isaak, M. J. Hanna, “User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection,” *Computer*, vol.51, no.8, pp.56-59, Aug.2018. Article (CrossRef Link)

[4] K.Swati. [Online]. Available: https://the hacker news.com/2018/03/facebook-cambridge-analytica.html.

[5] L.Sweeney, “k-Anonymity: a model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol.10, no.5, pp.557-570, Oct.2002. Article (CrossRef Link)

[6] B. Zhou, J. Pei, “The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighbourhood attacks,” *Knowledge and Information Systems*, vol.28, no.1, pp.47-77, Jul. 2011. Article (CrossRef Link)

[7] S.Chester, M.B.Kapron, G.Srivastava, "Complexity of social network anonymization," *Social Network Analysis and Mining*, vol.3, pp.151-166, Jun.2013. Article (CrossRef Link)

[8] J.Casas-Roma, J.Herrera-Joancomartí, V.Torra. "k-Degree anonymity and edge selection: improving data utility in large networks," *Knowledge and Information Systems*, vol.50, no.2, pp.447-474, Feb.2017.Article (CrossRef Link)

[9] K.R.Langari, S.Sardar, A.A.S.Mousavi, "Combined fuzzy clustering and firefly algorithm for privacy preserving in social networks," *Expert Systems with Applications*, vol.141, pp.1-12, Mar.2020.Article (CrossRef Link)

[10] P. Boldi, F. Bonchi, A. Gionis, "Injecting uncertainty in graphs for identity obfuscation," *Proceedings of the VLDB Endowment*, vol.5, No.11, pp.1376-1387, Aug.2012. Article (CrossRef Link)

[11] P.Mittal, C.Papamanthou, D. Song, "Preserving Link Privacy in Social Network Based Systems," in *Proc. of NDSSS*, San Diego, USA, pp.1-16, Feb. 2013. Article (CrossRef Link)

[12] J.Hu, J.Yan, Z Wu, "A Privacy-Preserving Approach in Friendly-Correlations of Graph Based on Edge-Differential Privacy," *Journal of Information Science and Engineering*, vol.35, no.4, pp.821-837, Jul.2019. Article (CrossRef Link)

[13] C. Dwork, "Differential Privacy," in *Proc. of ICALP*, Venice, Italy, pp.1-12, Jul. 2006. Article (CrossRef Link)

[14] X.Ying, X.Wu, "Randomizing social networks: a spectrum preserving approach," in *Proc. of SDM*, Atlanta, GA, USA, pp.739–750, Apr.2008. Article (CrossRef Link)

[15] R.Casas, "Privacy-Preserving on Graphs Using Randomization and Edge-Relevance," in *Proc. of MDAI*, Tokyo, Japan, pp.204-216, Oct.2014. Article (CrossRef Link)

[16] J.Casas, J.Herrera, V.Torra, "An Algorithm For k-Degree Anonymity On Large Networks," in *Proc. of ASONAM*, Niagara, Ontario, Canada, pp.671-675, Aug.2013. Article (CrossRef Link)

[17] J.Cheng, A.W.Fu, J.Liu, "K-isomorphism: privacy preserving network publication against structural attacks," in *Proc. of ICMD*, Indianapolis, Indiana, USA, pp.459-470, Jun. 2010. Article (CrossRef Link)

[18] H.Rong, T.Ma, M.Tang, "A novel subgraph K+-isomorphism method in social network based on graph similarity detection," *Soft Computing*, vol.22, no.8, pp.2583-2601, Apr. 2018. Article (CrossRef Link)

[19] Y.Liu, J.Jin, Y.Zhang, "A new clustering algorithm based on data field in complex networks," *Journal of Supercomputing*, vol.67, no.3, pp.723-737, Mar. 2014. Article (CrossRef Link)

[20] F.Yu F, M.Chen, B.Yu, "Privacy preservation based on clustering perturbation algorithm for social network," *Multimedia Tools and Applications* ,vol.77, no.9, pp. 11241-11258, 2018. Article (CrossRef Link)

[21] H. H. Nguyen, A.Imine, M. Rusinowitch, "A Maximum Variance Approach for Graph Anonymization," in *Proc. of FPS*, Montreal, Canada, pp.49-64, Nov.2014. Article (CrossRef Link)

[22] H.H.Nguyen, A.Imine, M.Rusinowitch, "Anonymizing Social Graphs via Uncertainty Semantics," in *Proc. of ICCS*, Singapore, pp.495-506, Apr. 2015. Article (CrossRef Link)

[23] J.Yan, L.Zhang L, C.W.Shi. "Uncertain Graph Method Based on Triadic Closure Improving Privacy Preserving in Social Network," in *Proc. of NaNA*, Kathmandu, Nepal, pp.190-195, Oct.2017. Article (CrossRef Link)

[24] C. Dwork, "Differential privacy: a survey of results," in *Proc. of TAMODELSC*, Xi'an, China, pp.1-19, Apr. 2008. Article (CrossRef Link)

[25] R.K.Macwan, J.S.Patel, "Node differential privacy in social graph degree publishing," *Procedia computer science*, vol.143, pp.786-793, 2018. Article (CrossRef Link)

[26] Q.Qian, Z.Li, P.Zhao, "Publishing Graph Node Strength Histogram with Edge Differential Privacy," in *Proc. of DSAA*, Gold Coast, QLD, Australia, pp.75-91, May, 2018. Article (CrossRef Link)

[27] B.P.Nguyen, H.Ngo, J.Kim, "Publishing Graph Data with Subgraph Differential Privacy," in *Proc. of DSAA*, Hanoi, Vietnam, pp.134-145, Apr. 2015.Article (CrossRef Link)

[28] T.Dong, Y.Zeng, H.Z.Liu, “A Differential Privacy Topology Scheme for Average Path Length Query,” *Journal of Information Science & Engineering*, vol.37, no.4, pp.134-145, Jul. 2021. Article (CrossRef Link)

[29] H.Jiang, J.Pei, D,Yu, “Applications of Differential Privacy in Social Network Analysis: A Survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol.35, no.1, pp. 108-127, 2023. Article (CrossRef Link)

[30] V.Karwa, A.B.Slavkovi´c, “Differentially private graphical degree sequences and synthetic graphs,” in *Proc. of ICPSD*, Palermo, Italy, pp.273-285, Sep. 2012. Article (CrossRef Link)

[31] Z.Qin, T.Yu, Y.Yang, “Generating Synthetic Decentralized Social Graphs with Local Differential Privacy,” in *Proc. of CCS*, Dallas, Texas, USA, pp.425-438, Oct. 2017. Article (CrossRef Link)

[32] C.Liu, S.Chen, S.Zhou, “A general framework for privacy-preserving of data publication based on randomized response techniques,” *Information Systems*, vol.96, pp.1-12, Feb. 2021. Article (CrossRef Link)

[33] A.van den Hout A, P. G. M.van der Heijden, “Randomized response,statistical disclosure control and misclassificatio: a review,” *International Statistical Review*,vol.70, no.2, pp.269-288, 2002. Article (CrossRef Link)

[34] V.Karwa V, B.A.Slavkovi´c, P.Krivitsky, “Differentially private exponential random graphs,” in *Proc. of ICPSD*, Ibiza, Spain, pp.143-155, September, 2014.

[35] Stanford Large Network Dataset Collection. [Online]. Available: http://snap.stanford.edu/data/.

Jun Yan received the M.S. degree in College of Earth Exploration Science and Technology from Jilin University. He is currently pursuing the Ph.D. degree in School of Computer Sciensce, Shaanxi Normal University. His research interests include network security and privacy preserving.

Jiawang Chen received the PhD degree in computer science and technology from Shaanxi Normal University, China. His research interests include network security, deep learning and graph neural networks.

Yihui Zhou received her B.E. degree, M.S. degree and Ph.D. degree in College of Mathematics and Information Science from Shaanxi Normal University, Shaanxi, China, in 2003, in 2006 and in 2009, respectively. Now she is a lecturer in School of Computer Science, Shaanxi Normal University. Her research interests include information security and privacy preserving.

Zhenqiang Wu received his B.S. degree in 1991 from Shaanxi Normal University, China, and received his M.S. and Ph.D degrees in 2002, and 2007 respectively, all from Xidian University, China. He is currently a full professor of Shaanxi Normal University, China. Dr. Wu’s research interests include computer communications networks, mainly wireless networks, network security, anonymous communication, and privacy protection etc. He is a member of ACM and senior of CCF.

Laifeng Lu received M.S.degree and Ph.D.degree in Computer system architecture from Xi'dian University, Shaanxi, China. Now she is an associate professor in Shaanxi Normal University. Her research interests include security and privacy protection.