applied sciences

MDPI

Article

A Node Differential Privacy-Based Method to Preserve Directed Graphs in Wireless Mobile Networks

Jun Yan [1,2,3], Yihui Zhou [1] and Laifeng Lu [4,*]

[1] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; yanrongjunde@snnu.edu.cn (J.Y.); zhouyihui@snnu.edu.cn (Y.Z.)
[2] School of Mathematics and Computer Applications, Shangluo College, Shangluo 726000, China
[3] Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi Province, Shangluo 726000, China
[4] School of Mathematics and Statistics, Shaanxi Normal University, Xi'an 710119, China
* Correspondence: lulaifeng@snnu.edu.cn

Abstract: With the widespread popularity of Wireless Mobile Networks (WMNs) in our daily life, the huge risk to disclose personal privacy of massive graph structure data in WMNs receives more and more attention. Particularly, as a special type of graph data in WMNs, the directed graph contains an amount of sensitive personal information. To provide secure and reliable privacy preservation for directed graphs in WMNs, we develop a node differential privacy-based method, which combines differential privacy with graph modification. In the method, the original directed graph is first divided into several sub-graphs after it is transformed into a weighted graph. Then, in each sub-graph, the node degree sequences are obtained by using an exponential mechanism and micro-aggregation is adopted to get the noised node degree sequences, which is used to generate a synthetic directed sub-graph through edge modification. Finally, all synthetic sub-graphs are merged into a synthetic directed graph that can preserve the original directed graph. The theoretical analysis proves that the proposed method satisfies differential privacy. The results of the experiments demonstrate the effectiveness of the presented method in privacy preservation and data utility.

Keywords: wireless mobile networks; directed graph; differential privacy; graph modification

Citation: Yan, J.; Zhou, Y.; Lu, L. A Node Differential Privacy-Based Method to Preserve Directed Graphs in Wireless Mobile Networks. *Appl. Sci.* **2023**, *13*, 8089. https://doi.org/10.3390/app13148089

Academic Editor: Andrea Prati

Received: 10 June 2023
Revised: 2 July 2023
Accepted: 3 July 2023
Published: 11 July 2023

1. Introduction

Over the past several years, wide applications of 4G mobile wireless networks have brought us tremendous convenience. For example, through the 4G mobile wireless networks, we can enjoy a large number of online services, including mobile shopping/payment, mobile office, mobile gaming, etc. [1]. Nowadays, with the wide popularity of various innovative applications, such as Vehicle-to-Everything (V2X), AR, holographic communications, etc. [2], mobile wireless networks make our daily life more convenient. However, mobile wireless networks also present us with a great challenge while providing tremendous convenience for us. For instance, as a large amount of data including sensitive information is published or shared in mobile wireless networks without privacy preservation, a lot of individual privacy is leaked, which results in many social security problems [3]. In particular, [4] points out that data leakage is one of the most frequent mobile security threats. Therefore, it is crucial to pay close attention to individual privacy in mobile wireless networks.

More importantly, there is a large amount of personal privacy information, including identity privacy, semantic attribute privacy, and link privacy, in the graph structure data in mobile wireless networks [5]. To address the privacy issue in the graph structure data, many graph modification methods have been proposed, which are divided into three categories: edge/node modification, clustering, and uncertain graph [6]. In the edge/node modification method, the edge randomization method randomly adds or deletes edges in the

original graph while retaining the characteristics of an original graph as much as possible [7]. To overcome the shortcoming of this method, many k-anonymity methods that can resist attacks based on the structure of the graph have been devised, such as (k,l)-anonymity [8], k2-anonymity [9] and k-neighborhood sub-graph anonymity [10]. Clustering-based methods, called generalization methods, usually group nodes and edges into super-nodes and super-edges, which hide the detail of nodes and edges in the graph [11]. Furthermore, the method combining k-anonymity with node clustering is designed, which can provide sufficient privacy preservation while retaining data utility [12]. Compared with the two methods mentioned above, the uncertain graph method, which rejects the uncertainty on the edges of a graph to generate an uncertain graph, can get better data utility than them. Although graph modification can preserve the graph structure data, it is not able to resist attacks based on background knowledge.

As a gold-standard notion of privacy that can provide a strict privacy guarantee [13], differential privacy has been adopted to preserve graph structure data [14]. For instance, differential privacy has been extensively applied to preserve various statistical values of the graph, such as the degree distribution [15], frequent graphics patterns [16], and triangle count [17]. In addition, it can also be used to generate a synthetic graph to preserve the original graph. In [18], a synthetic graph is released by using a differential private estimator of the parameters of a special model, which is an exponential family model with the degree sequence as a sufficient statistic. To improve the data utility of a differential private synthetic graph [19], devises a differential private graph generator based on the dK-graph model. Different from graph modification, differential privacy usually employs noise to achieve privacy preservation, which results in insufficient data utility.

However, the methods introduced above mainly concentrate on undirected graphs. As a special graph, directed graphs, such as the who-follows-whom social graph on Twitter, not only possess the relations in graphs but also have the direction information. Therefore, it is hard to adopt these methods to preserve directed graphs when it is published or shared. By considering the direction information of edges, a few k-anonymity methods have been designed in [20,21]. But the k-anonymity method is not able to resist attacks based on background knowledge and only withstand some special attacks, these methods cannot provide sufficient privacy preservation for directed graphs. As a result, it is a great challenge to preserve directed graphs.

To solve this problem, we propose a useful method that combines differential privacy with graph modification to preserve directed graphs. In particular, compared with edge differential privacy, node differential privacy can provide stronger privacy preservation. Thus, node differential privacy is used to add noise on degree sequences, and edge modification utilizes noised degree sequences to generate a synthetic directed graph, which provides strong privacy preservation for the original directed graph. Additionally, to improve data utility, the original directed graph is divided into many sub-graphs, and the perturbations are only added in each sub-graph, which is useful to maintain the whole graph structure. In particular, the exponent mechanism is adopted to truncate degree sequences, which can ensure that the minimum noise is added to the degree sequences. Moreover, the ranking micro-aggregation effectively reduces the noise added to the degree sequences. According to the noised degree sequences, the relationship between two nodes is utilized to modify the edges of nodes, which can retain the original graph structure. Therefore, the designed method not only provides strong privacy preservation but also maintains the data utility.

In this paper, our contributions can be summarized as follows:

We propose a method based on node differential privacy to preserve directed graphs in wireless mobile networks. Particularly, node differential privacy and edge modification are combined to generate a synthetic directed graph that provides strong privacy preservation for the original directed graph.

We present four algorithms to maintain data utility in the proposed method. First of all, the Louvain algorithm is used to divide the original directed graph into several sub-graphs. Then, the node degree sequences in each sub-graph are generated by the

GSEM (generating degree sequence based on exponent mechanism) algorithm and ADPRA (adding noise based on differential privacy with the ranking micro-aggregation) algorithm adds less noise on these node degree sequences. In the end, GGM (generating synthetic graphs based on graph modification) algorithm generates a synthetic directed sub-graph that maintains the properties of an original sub-graph.

We demonstrate the performances of the proposed method on several different real data sets, and the experimental results show that the proposed method is effective in privacy preservation and data utility.

The rest of this paper is organized as follows. Section 2 reviews the related methods to preserve the graph structure data. In Section 3, the preliminaries are introduced. Then, the proposed method is described in detail in Section 4. Section 5 demonstrates the performance of the proposed method in privacy preservation and data utility. In Section 6, the existing challenges and promising future directions are discussed.

2. Related Works

With individual privacy on MWNs attracting more and more attention, various techniques have been proposed to provide privacy preservation. In this section, we will focus on methods that include two categories: graph modification and differential privacy.

In graph modification, there are three important graph modification methods: edge and node modification methods, generalization or clustering methods, and uncertain graph methods. In edge and node modification methods, to improve data utility, Ying X. [7] proposed two algorithms to preserve the original graph while keeping the spectral properties of the graph unchanged as much as possible. In [22], Casas-Roma designed a method to protect the most important edges, which obtained a better trade-off between privacy preservation and data utility. In generalization methods that focus on how to generate so-called super-nodes and super-edges, Yu F. [23] developed a clustering perturbation algorithm that adopted some perturbations to maintain the whole structure of the social net work and reduce privacy leakages. In uncertain graph methods, Boldi in [24] designed a (k,)-obfuscation method based on injecting uncertainty to get an uncertain graph, which was similar to the original graph. To prevent link attacks based on background knowledge, Hu J developed an uncertain graph method based on edge-differential privacy, which also had better data utility in [25].

In addition, k-anonymity [5] had been widely used to generate anonymous graphs to preserve graph data. Considering the number of mutual friends (NMF) between two users, [26] developed a k-anonymity method that made use of the mutual friend sequence to ensure the existence of at least k elements holding the same value for better data utility. In [27], the new (k, l)-degree anonymity algorithm was devised to modify the original networks based on a sequence of edge editing operations. In this algorithm, a location entropy metric was considered to select the important edges so it could achieve minimum edge modification to increase data utility. Meanwhile, to resist insider attacks in collaborative social networks, [28] developed a k-anonymity method based on the clustering, in which a scalable non-deterministic clustering was utilized to prevent the structure attacks.

In differential privacy methods, many methods based on differential privacy have been presented for graph data since C. Dwork came up with differential privacy, which was classified into two kinds: preserving specific sensitive statistics of graphs and generating differential private graphs. For publishing higher order network statistics, i.e., joint degree distribution, Iftikhar [29] designed a general framework for releasing dK-distributions under node differential privacy, in which sensitivity was regulated by a graph projection algorithm, which transformed graphs into bounded graphs. To accurately estimate sub-graph counts, [30] proposed a novel multi-phase framework under DDP (decentralized differential privacy), which was able to control the minimum local noise scale to preserve the sub-graph counts. Furthermore, some statistical data in graph data, such as triangle counts, centrality and shortest paths were preserved by differential privacy before they

were released [31,32]. Apart from preserving the statistical data, differential privacy is also applied to generate a synthetic graph. In [33], Vishesh Karwa developed an algorithm to attain a graphical degree partition of a graph preserved by differential privacy, which could also be used to construct synthetic graphs. Ref. [34] proposed an LDPGen, which could generate a synthetic graph after structurally similar users were clustered together according to optimal parameters.

3. Preliminaries Knowledge

In this paper, a directed network is regarded as a simple, directed graph $G = (V, E)$, where $V = (v_1, v_2, \ldots, v_n)$ is the set of nodes, and E is the links table, each link (i, j) denotes a relationship from v_i to v_i.

Definition 1 (The undirected graph and the directed graph)**.** *As shown in Figure 1, the Figure 1a is an undirected graph, while the Figure 1b represents a directed graph, where each edge denotes a relationship from one node to another node. In the Figure 1b, the edge (v_1, v_4) denotes a link relation from node v_1 to node v_4.*

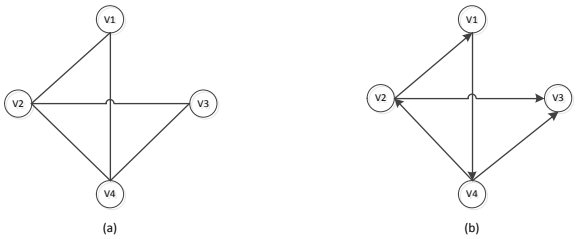

Figure 1. The undirected graph and the directed graph.

Definition 2 (Neighboring directed graph)**.** *For two directed graphs $G_1 = (V_1,E_1)$, $G_2 = (V_2,E_2)$, if $|V_1 \bigoplus V_2| + |E_1 \bigoplus E_2| = 1$, where $\bigoplus$ is Exclusive—OR operation, we can say G_1 and G_2 are neighbors.*

As shown in Figure 2, compared with the Figure 2b, Figure 2a has one more different node with three directed edges. So the Figure 2a,b are neighboring graphs.

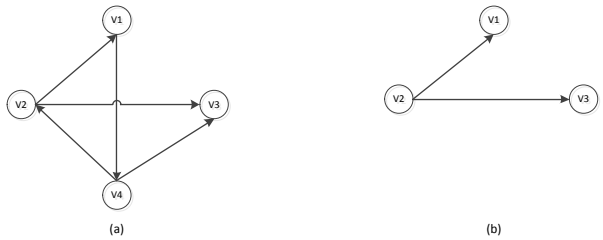

Figure 2. The example of neighboring graph of node.

Definition 3 (Differential Privacy)**.** *For all outputs* S *belong to Range(*Z*), if we can obtain the result as follows:*

$$P_r[Z(Ga) \in S] = e^{\epsilon} \times P_r[Z(Gb) \in S] \tag{1}$$

where Ga and Gb are neighbors, ϵ is a privacy preservation level, we can see that the algorithm Z satisfies ϵ-differential privacy.

In order to achieve ϵ-differential privacy, we must perturb the outputs of queries in two ways, which include the Laplace mechanism and the exponential mechanism.

Definition 4 (Laplace Mechanism)**.** *For a sequence of queries F: G→G, if the following holds:*

$$Z(G) = F(G) + Lap(\Delta f/\epsilon) \tag{2}$$

where $\mu = 0$, $b = \Delta f/\varepsilon$ and $Lap(\Delta f/\varepsilon)$ represents the Laplace noise, the way that makes an algorithm Z satisfies ε-differential privacy by adding Laplace noise is the Laplace mechanism.

In the Laplace mechanism, the Laplace noise distribution is shown in Equation (4).

$$n(x) = 1/2b * exp(-|x - \mu|/b) \tag{3}$$

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

Definition 5 (Exponential Mechanism)**.** *Given a dataset D, an output range T, a privacy budget ϵ, and a utility function* U*: (*D*,* t*)* $\rightarrow$ R*, a mechanism* M *that selects an output* t $\in$ T *with probability proportional to exp($\frac{\epsilon \cdot U(D,t)}{2\triangle U}$) satisfies ϵ-differential privacy.*

Definition 6 (Parallel composition properties)**.** *Given a sequence of algorithms $\{A_1, A_2, \ldots, A_n\}$, and each algorithm A_i satisfies ϵ_i differential privacy, if these algorithms are applied independently on a disjoint subset of the input database* D*, this data process is called the parallel composition properties of differential privacy, which satisfies* max ϵ_i *differential privacy.*

4. The Proposed Method

4.1. The Framework of Method

To preserve the directed graph in wireless mobile networks, we propose a novel method based on differential privacy, which combines node differential privacy and graph modification to provide sufficient privacy preservation while retaining data utility. In addition, we assume that the original directed graph is a simple connected static directed graph without node attributes.

As shown in Figure 3, the model of the developed method consists of four steps. In step 1, after the original directed graph is converted into a weight graph, the weight graph is divided into some sub-graphs according to the optimal modularity [35]. Then, the node differential privacy is utilized to generate two differential private degree sequences (an in-degree sequence and an out-degree sequence) from each sub-graph in step 2. In particular, the exponent mechanism is used to get the degree sequences of nodes in each sub-graph, while the ranking micro-aggregation is applied to add noise to them. Next, in step 3, each sub-graph is modified by adding or deleting edges according to the noised degrees. Simultaneously, the relationship between nodes is considered in edge modification. At last, all modified sub-graphs are merged into a differential private directed graph which provides privacy preservation for the original directed graph in Section 4.

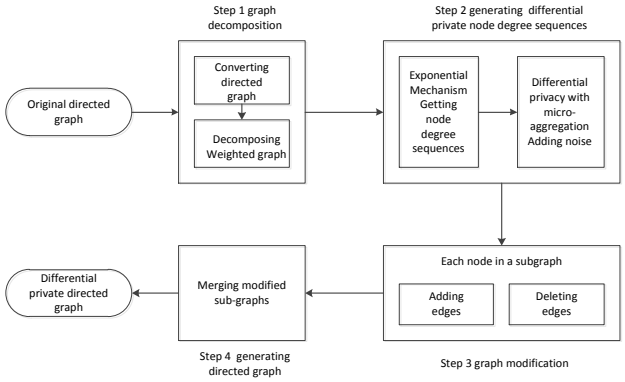

Figure 3. The framework of method.

In particular, as node differential privacy provides stronger privacy preservation than graph modification and edge differential privacy, it is adopted to provide better privacy preservation for the directed graphs in mobile wireless networks. At the same

time, to maintain data utility, the Louvain algorithm is used to divide the original directed graph into many sub-graphs, and graph modifications are limited in them. Then, the exponent mechanism truncates the degree sequences to add the minimum noise to the degree sequences when the privacy budget is given. Next, the ranking micro-aggregation effectively reduces the noise added to the degree sequences, which is proved by the mathematical analysis. Finally, the relationship between two nodes is utilized to modify the edges of nodes, which can retain the original graph structure. Therefore, the proposed method achieves the trade-off between privacy and data utility.

To summarize, we develop a novel method that can achieve privacy preservation for the directed graph while maintaining the data utility.

4.2. DGNDP (Synthetic Directed Graph Based on Node Differential Privacy) Algorithm

In Algorithm 1, the goal is to generate a differential private directed graph. At first, to better gather the nodes of the directed graph, we convert this directed graph into a weighted graph and utilize the Louvain algorithm to divide the weighted graph into several sub-graphs. In each sub-graph, the GSEM algorithm is used to get an out-degree sequence and an in-degree sequence. Then, the ADPRA algorithm adds noise to these degree sequences. After that, the GGM algorithm generates a synthetic directed sub-graph according to the noised degrees. Finally, all synthetic directed sub-graphs are merged into a differential private directed graph.

Algorithm 1 DGNDP algorithm

Input: an original directed graph G
Output: a synthetic directed graph $G^{'}$
1: a weighted graph $G_w \leftarrow$ converting an original directed graph G
2: a set of directed sub-graph $Ssub \leftarrow$ decomposing a weighted graph G_w
2: a set of $S_{Gn} = \{\ \}$
3: for S_{Gi} in S_{sub}:
4: S_{dout} = GSEM algorithm (S_{Gi}, ϵ_1)
5: S_{din} = GSEM algorithm (S_{Gi}, ϵ_1)
6: an out-degree sequence $S_{doutn} \leftarrow$ ADPRA algorithm (S_{Gi}, ϵ_2)
7: an in-degree sequence $S_{dinn} \leftarrow$ ADPRA algorithm (S_{Gi}, ϵ_2)
8: $Sn_{Gi} \leftarrow$ GGM algorithm (SG_i, S_{doutn}, S_{dinn})
9: S_{Gn} adding Sn_{Gi}
10: $G^{'} \leftarrow$ merging S_{Gn}
11: Return a synthesis directed graph $G^{'}$

4.2.1. GSEM (Generating Degree Sequence Based on Exponent Mechanism) Algorithm

For an in-degree sequence or an out-degree sequence in a directed sub-graph, when the Laplace noise is added to this degree sequence, the smaller the degree of the node, the greater the damage caused by the added noise. To reduce noise added to the degree sequence, the noise is only added on nodes with large degrees in this sequence and nodes with small degrees are deleted from this sequence.

In particular, after a degree sequence of a sub-graph sorted from large to small is truncated according to a certain threshold t, the Laplace noise is only added to the rest of the degree sequence. Thus, there are two kinds of errors: One is the reconstruction error caused by the truncated part of the degree sequence, and the other is Laplace noise added on the rest of this degree sequence. The larger the certain threshold t is, the more parts of this sequence are deleted, which will result in a larger reconstruction error. On the contrary, the smaller the certain threshold t, the more Laplace noise is added. Therefore, the exponent mechanism is applied to get an optimal t, which can be used to add minimum noise to the degree sequence.

Given a directed sub-graph S_{Ga} = (V_a, E_a), the out degree sequence of (S_{Ga}) is $Seq_{out}(S_{Ga})$:$[d_1, d_2, \ldots, d_n]$, where n is the number of nodes, then a query function f is given:

$$f \rightarrow Seq_{out}(S_{Ga})$$

After the $Seq_{out}(S_{Ga})$ is sorted from small to large and is truncated by t, there are two kind of errors in $Seq_{out}(S_{Ga})$: the reconstruction error and the Laplace noise.

$$\begin{aligned} &Error(Seq_{out}(S_{Ga})) \\ &= RE(Seq_{out}(S_{Ga})) + LE(Seq_{out}(S_{Ga})) \end{aligned}$$

where $RE(Seq_{out}(S_{Ga}))$ is the error caused by the truncation, $LE(Seq_{out}(S_{Ga}))$ represents the error brought by the Laplace noise.

$$RE(Seq_{out}(S_{Ga})) = \sqrt{\sum_{i=1}^{t} |d_i|^2}$$

$$LE(Seq_{out}(S_{Gi})) = E(\sqrt{\sum_{i=t+1}^{m} lap(\frac{\triangle f}{\varepsilon})^2})$$

where m is $n - t$.

$$\begin{aligned} &RE(Seq_{out}(S_{Ga})) + LE(Seq_{out}(S_{Ga})) \\ &= \sqrt{\sum_{i=1}^{t} |d_i|^2} + \sqrt{2 * (n-t)} * \frac{\triangle f}{\varepsilon} \end{aligned}$$

To gain a minimum value of $Error(Seq_{out}(S_{Ga}))$, the exponent mechanism is utilized to select a best threshold t. Then, there is a scoring function:

$$U(S_{Ga}, t) = \sqrt{\sum_{i=1}^{t} |d_i|^2} + \sqrt{2 * (n-t)} * \frac{\triangle f}{\varepsilon}$$

As the node differential privacy is employed to achieve differential privacy, the $\triangle U$ is:

$$\triangle U = U(S_{Ga}, t) - U(S_{Ga'}, t) = \triangle RE + \triangle LE$$

where there is only one node difference between S_{Ga} and $S_{Ga'}$.

$$\begin{aligned} \triangle RE = &\leq max(\sqrt{\sum_{i=1}^{t} |d(S_{Ga})_i|^2} - \sqrt{\sum_{i=1}^{t} |d(S_{Ga'})_i|^2}) \\ &\leq max(\sum_{i=1}^{t} |d(S_{Ga})_i| - \sum_{i=1}^{t} |d(S_{Ga'})_i|) \\ &\leq d_{max} \end{aligned}$$

$$\triangle LE = \triangle f$$

$$\begin{aligned} \triangle f_t = |f_t(S_{Ga}) - f_t(S_{Ga'})| &= max(degree(S_{Ga})) \\ &= d_{max} \end{aligned}$$

where $\triangle f$ is the sensitive of a query function f.

Therefore, $\triangle U$ is:

$$\triangle U = \triangle RE + \triangle LE \leq 2d_{max}$$

The probability that the threshold t can be selected is

$$p_r(t) = \frac{\exp\left(-\frac{\varepsilon_1 U(S_{Ga},t)}{2\Delta U}\right)}{\sum_{i=1}^{d_{max}-1}\exp\left(-\frac{\varepsilon_1 U(S_{Ga},t)}{2\Delta U}\right)}$$

Finally, a threshold t is obtained through the exponent mechanism and used to generate an out-degree sequence S_{dout}. In the same way, an in-degree sequence S_{din} is also obtained.

Thus, to ensure that the minimal noise is added to a degree sequence when the privacy budget is given, the noise is added to the truncated degree sequence.

As shown in Algorithm 2, in line 1, according to a directed sub-graph S_{Ga}, an out degree sequence $Seqt_{out}$ is generated. Then, the d_{max} is obtained from the out degree sequence $Seqt_{out}$ in line 2. From line 3 to line 5, the exponent mechanism is used to get a threshold t. In line 6, an out-degree sequence Seq_{out} is truncated to get an out-degree sequence S_{dout}, and it is returned in line 7.

Algorithm 2 GSEM algorithm

Input: a directed $S_{Ga} = (V_a, E_a)$, a privacy budget ϵ_1
Output: an out-degree sequence S_{dout}
1: an out degree sequence$Seq_{out} \leftarrow$ a directed graph $S_{Ga} = (V_a, E_a)$
2: d_{max}= the maximum degree of Seq_{out}

3: for 1 to t:
4: the scoring function

$$U(S_{Ga}, t) = \sqrt{\sum_{i=1}^{t} |d_i|^2} + \sqrt{2 * (n-t)} * \frac{\triangle f}{\epsilon}$$

5: selecting t with probability

$$p_r(t) \propto exp(-(\epsilon_1 U(S_{G1}, t)/2 * \triangle U)$$

6: an out-degree sequence $S_{dout} \leftarrow$ an out degree sequenceSeq_{out}
7: Return an out-degree sequence S_{dout}

4.2.2. ADPRA (Adding Noise Based on Differential Privacy with the Ranking Micro-Aggregation) Algorithm

Given an ordered degree sequence $d = [d_1, d_2, \ldots, d_n]$, it is aggregated into $\frac{n}{k}$ clusters. In each cluster, there are k continuous degree values, except perhaps one cluster that contains up to $2k-1$ consecutive values. Then, there is a sequence of the centroid of these clusters denoted by $[dc_1, dc_2, \ldots, dc_{n/k}]$. In this ordered degree sequence, if any single d_i in d is replaced by $\bar{d}$, $|d_i - \bar{d}| \leq \triangle$, then there is a new sequence of the centroid of these new clusters which is described as $[\bar{dc_1}, \bar{dc_2}, \ldots, \bar{dc_{n/k}}]$. As a result, it holds that

$$\sum_{m=1}^{[n/k]} \left|\overline{dc_m} - dc_m\right| \leq \Delta/k$$

Compared with edge differential privacy, node differential privacy can provide stronger privacy preservation. Nevertheless, node differential privacy results in insufficient data utility. To mitigate this problem, the ranking micro-aggregation is introduced to improve data utility. In particular, with the help of the ranking micro-aggregation, this algorithm generates two differential private degree sequences with effective data utility, which are useful for the graph modification in the next step.

Without losing generality, assume $\bar{d}_i > d_i$, and n can be divided by k. Thus, there are n/k clusters, with each cluster m having consecutive values from $d_{(m-1)k+1}$ to d_{mk}. In particular, each d_i belongs to a cluster m_i.

Then two cases are discussed.

Case 1: if $\bar{d}_i \leq d_{m_i k+1}$, then is still in cluster $\bar{m}_i$. Except for the cluster m_i, the centroids of other clusters are unchanged. The centroids of the cluster $\bar{m}_i$ increase $\frac{\Delta}{k}$, because $\bar{d}_i = d_i + \triangle$. Therefore, this case meets the requirements of the ranking micro-aggregation.

Case 2: if $\bar{d}_i \geq d_{m_i k+1}$, then $\bar{d}_i$ is not in cluster m_i. Therefore, two and more changes for the $\bar{d}_i$ replace the d_i: the clusterm_i lose d_i and a cluster $\bar{m}_i$ obtain $\bar{d}_i$(for $\bar{m}_i > m_i$). For keeping the number of clusters m_i unchanged, the cluster m_i gains $d_{m_i k+1}$; in return, the cluster m_i+1 loses $d_{m_i k+1}$ and obtains $d_{m_{i+1} k+1}$, until the cluster $\bar{m}_i$ gives its smallest value $d_{(\bar{m}_i-1)k+1}$ to cluster $\bar{m}_i - 1$ and obtains $\bar{d}_i$. From cluster $\bar{m}_i + 1$ to the end cluster, there is nothing that takes place. The change of the centroids is shown as follows:

$$\begin{aligned}
&\triangle \sum_{m=1}^{[n/k]} \left| \overline{dc_m} - dc_m \right| \\
&= \sum_{m=m_i}^{\bar{m}_i} \left| \overline{dc_m} - dc_m \right| \\
&= \frac{d_{m,k+1} - d_i}{k} + \frac{d_{(m_i+1)k+1} - d_{m,k+1}}{k} + \\
&\ldots + \frac{\overline{d_i} - d_{(\bar{m}_i-1)k+1}}{k} \\
&= \frac{\overline{d_i} - d_i}{k} = \frac{\Delta}{k}
\end{aligned}$$

when n is not a multiple of k, there are n/k clusters and one of them holds values between $k + 1$ and $2k - 1$. If in case 1, when the larger cluster is cluster m_i, the change of centroid of m_i is less than Δ/k. While if the larger cluster is another cluster, nothing will happen and it will meet the requirements of the theorem. If in case 2, a changed cluster is the larger cluster, one of the fractions in the third term of expression shown above has a denominator that is greater than k and the overall sum is less than Δ/k. Therefore, the lemma is set up; if the larger cluster is not affected, the lemma also holds.

In the ADPRA algorithm, given a query function

$$f(S_{Gi}) \rightarrow AS_{dout}$$

because the node differential privacy is used in this algorithm, according to the analysis mentioned above, the sensitivity of the query function f is

$$\Delta f = \max_{S_{Gi}, S'_{Gi}} \| f(S_{Gi}) - f(S'_{Gi}) \|_1 = \frac{d_{\max}}{k}$$

Therefore, due to the ranking micro-aggregation, the sensitivity of the query function decreases, so that the noise added to the degree sequence is reduced. To sum up, the ADPRA algorithm provides differential privacy for the degree sequence while maintaining data utility.

As illustrated in the Algorithm 3, line 1 sorts an out-degree sequence from small to large. After this sequence is aggregated in line 2, line 3 adds the Laplace noise to obtain a differential private out-degree sequence. In the same way, from line 4 to line 6, a differential private in-degree sequence is also got.

Algorithm 3 ADPRA algorithm

Input: an out-degree sequence S_{dout}, an in-degree sequence S_{din}, a privacy budget ϵ_2, the number of elements in a cluster, k
Output: a noised out-degree sequence NS_{dout}, a noised in-degree sequence NS_{din}
1: Sorting S_{dout} from large to small
2: $AS_{dout} \leftarrow$ micro-aggregating S_{dout}
3: $NS_{dout} \leftarrow$ adding the Laplace noise (($\triangle f$)/k*ϵ_2) on AS_{dout}
4: Sorting S_{din} from large to small
5: $AS_{din} \leftarrow$ micro-aggregating S_{din}
6: $NS_{din} \leftarrow$ adding a Laplace noise (($\triangle f$)/k*ϵ_2) on AS_{din}
7: Return a noised out-degree sequence NS_{dout}, a noised in-degree sequence NS_{din}

4.2.3. GGM (Generating Synthetic Graph Based on Graph Modification) Algorithm

To generate a synthetic directed graph by using a noised degree sequence, the graph modification method is adopted to present the GGM algorithm which consists of three steps. In this algorithm, step 1 compares two out-degree sequences NS_{dout} and S_{dout} as well as two in-degree sequences NS_{din} and S_{din}, and records the difference between them. In step 2, some edges are added into a sub-graph Sn_{Gi} as the value of the out and in the degree of a node increases. To reduce the perturbation caused by adding edges, the nodes with increased out-degree and the nodes with increased in-degree are paired to add edges between them. In the last step, an out or in-degree sequence is selected to delete edges between these nodes in it and their neighborhood nodes. In particular, the relationships between nodes are considered when edges are added or deleted, which can preserve the original structure as much as possible. In the end, the GGM algorithm generates a differential private synthesis directed graph, which can effectively preserve the original directed graph.

The detail of the Algorithm 4 is demonstrated as follows. In line 2 to line 3, after the NS_{dout} and S_{dout} are compared, Do_1 and Do_2, which record nodes and their increased/decreased out-degree values are obtained. Then, Di_1 and Di_2, which record nodes and their increased/decreased in-degree values are obtained. Starting on line 4, some edges are added into the graph Sn_{Gi} as few as possible. For each node, i in sorted Do_1, k nodes in Di_1, which are closer to node i than other nodes, are selected and edges are added between them from line 5 to 8. After that, the value in Di_1 is modified from line 9 to line 11. In order to delete edges the least amount possible, Do_2 and Di_2 are compared in line 12. If Do_2 is selected, as for each node i in Do_2, a set of nodes Ina, which is the intersection of N_a and Di_2, is gained. Then, if the number of IN_a is more than zero, we delete min {k,$|IN_a|$} edges from Sn_{Gi}; otherwise, the min {k,$|N_a|$}edges are removed from Sn_{Gi}. If the Di_2 is chosen, some edges are deleted in the same way as from line 22 to line 29. Therefore, through the graph modification method, a synthetic directed graph is generated.

Algorithm 4 GGM algorithm

Input: NS_{dout}, NS_{din}, S_{dout}, S_{din}, a directed sub-graph $S_{Gi} = (V_i, E_i)$
Output: a synthesis directed sub-graph Sn_{Gi}
1:$Sn_{Gi} \leftarrow S_{Gi}$
2: (Do_1, Do_2) $\leftarrow$ comparing S_{dout} with NS_{dout}
3: (Di_1, Di_2) $\leftarrow$ comparing S_{din} with NS_{din}
4: Sorting Do_1 and Di_1 from large to small
5: for node i in Do_1:
6: $k = Do_1$[i]
7: if $|Di_1| > 0$
8: selecting k nodes D_{i1} and adding edges from node i to those k nodes in S_{Gi}

Algorithm 4 *Cont.*

```
9:          modifying the values of those k nodes in Di_1
10:          if the values of node j in Di_1 nodes < 0:
11:              abandoning node j from Di_1
12: if sum (Do_2) > sum ( Di_2)
13:          for node i in Do_2:
14:              k = Do_2[i]
15:              a set of node N_a = neighborhood nodes of node i
16:              a set of node IN_a = intersection of N_a and nodes in Di_2
17:              if |IN_a|>0
18:                  selecting min {k,|IN_a|} nodes from IN_a and deleting edges from
node i to those nodes
19:              else:
20:                  selecting min {k,|N_a|} nodes from N_a and deleting edges from
node i to those nodes
21: else:
22:          for node i in Di_2:
23:              k = Di_2[i]
24:              a set of node Na = predecessors nodes of node i
25:              a set of node Ina = intersection of Na and nodes in Do_2
26:              if number of Ina > 0
27:                  selecting min {k,|IN_a|} nodes from IN_a and deleting edges from
node i to those nodes
28:              else:
29:                  selecting min {k,|N_a|} nodes from N_a and deleting edges from
node i to those nodes
30: Return Sn_Gi
```

4.2.4. Analysis of DGNDP Algorithm

Theorem 1. *The GSEM algorithm satisfies ε-differential privacy.*

Proof of Theorem 1. As discussed before, the probability to select the threshold t is

$$p_r(t) = \frac{\exp\left(-\frac{\varepsilon U(Sg,t)}{2\Delta U}\right)}{\sum_{t' \in O} \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)}$$

In this algorithm, we assumed Sg and $Sg^{'}$ are neighborhood graphs, where there is one node difference between them. For any variable t, the following result is obtained.

$$\frac{p_r(E(Sg,t))}{p_r(E(Sg^{'},t))} = \frac{\frac{\exp\left(-\frac{\varepsilon U(Sg,t)}{2\Delta U}\right)}{\sum_{t' \in O} \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)}}{\frac{\exp\left(-\frac{\varepsilon U(Sg^{'},t)}{2\Delta U}\right)}{\sum_{t' \in O} \exp\left(-\frac{\varepsilon U(Sg^{'},t')}{2\Delta U}\right)}}$$

$$= \left(\frac{\exp\left(-\frac{\varepsilon U(Sg,t)}{2\Delta U}\right)}{\exp\left(-\frac{\varepsilon_1 U(Sg^{'},t)}{2\Delta U}\right)}\right) \times \left(\frac{\sum_{t' \in O} \exp\left(-\frac{\varepsilon U(sG^{'},t')}{2\Delta U}\right)}{\sum_{t' \in O} \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)}\right)$$

$$\leq \exp\left(\frac{\varepsilon}{2}\right) \times \left(\frac{\sum_{t' \in 0} \exp\left(\frac{\varepsilon}{2}\right) \times \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)}{\sum_{t' \in 0} \exp\left(-\frac{\varepsilon U(sG,t')}{2\Delta U}\right)} \right)$$

$$\leq \exp\left(\frac{\varepsilon}{2}\right) \times \exp\left(\frac{\varepsilon}{2}\right) \times \left(\frac{\sum_{t' \in 0} \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)}{\sum_{t' \in 0} \exp\left(-\frac{\varepsilon U(Sg,t')}{2\Delta U}\right)} \right)$$

$$= \exp(\varepsilon)$$

It is clear that the process of selecting the threshold t satisfies differential privacy. In addition, in view of the principle of post-processing, it satisfies differential privacy to obtain an out-degree sequence and an in-degree sequence from the input graph through the exponent mechanism. Therefore, the GSEM algorithm satisfies differential privacy. □

Theorem 2. *The ADPRA algorithm satisfies ε-differential privacy.*

In this algorithm, the important task is to add the Laplace noise to the degree sequence by using differential privacy with micro-aggregation. In a graph Sg, let S be a query function : $Sg \rightarrow Ds_1$, where Ds_1 is a degree sequence from SDEM algorithm. Given that there is only one node difference between Sg and $Sg^{'}$, the sensitivity of S is

$$\begin{aligned} \Delta S &= \max_{Sg, Sg'} \left| S(Sg) - S\left(Sg'\right) \right|_1 \\ &= \max_{Sg, Sg'} \left| Ds1 - Ds1^{'} \right|_1 \\ &= d_{\max} \end{aligned}$$

After the micro-aggregation degree sequence Dms_1 is obtained from Ds_1, given a query function Sm: $Sg_1 \rightarrow Dms_1$, when there is only one node difference between Sg and $Sg^{'}$, there is $Sm(Sg) = Dms_1$, $Sm(Sg) = Dms_2$. The sensitivity of Sm is shown as follows.

$$\begin{aligned} \Delta Sm &= \max_{Sg, Sg'} \left| Sm(Sg) - \mathrm{Sm}\left(Sg'\right) \right|_1 \\ &= \max_{Sg, Sg'} \left| Dms1 - Dms2 \right|_1 \end{aligned}$$

According to the previous analysis, if the difference between two degree sequences is $dmax$, the difference between two micro-aggregation degree sequences is $\frac{dmax}{k}$, k is the number of entities contained in a cluster. Thus, the sensitivity of Sm is $\frac{dmax}{k}$. Because the sensitivity is reduced, less noise is added on the Dms_1, which improves the data utility of this algorithm. In summary, the ADPRA algorithm satisfies differential privacy, which is proved as follows.

Let $Pr[Sg]$ represents the probability density function of $LA(Sg,Sm,\epsilon)$, and $Pr[Sg^{'}]$ indicates the probability density function of $LA(Sg^{'},Sm,\epsilon)$.

$$\begin{aligned} \frac{p_r[LA(Sg)]}{p_r[LA(Sg)]} &= \frac{p_r[D - \mathrm{Sm}(Sg)]}{p_r[D - Sm(Sg^{'})]} \\ &= \frac{\frac{1}{2\frac{\Delta Sm}{\varepsilon}} \exp\left(-\frac{|D - \mathrm{Sm}(Sg)|}{\frac{\Delta Sm}{\varepsilon}}\right)}{\frac{1}{2\frac{\Delta Sm}{\varepsilon}} \exp\left(-\frac{|D - Sm(Sg^{'})|}{\frac{\Delta Sm}{\varepsilon}}\right)} \\ &= \frac{\exp\left(-\frac{|D - \mathrm{Sm}(Sg)|}{\frac{\Delta Sm}{\varepsilon}}\right)}{\exp\left(-\frac{|D - \mathrm{Sin}(Sg^{'})|}{\frac{\Delta Sm}{\varepsilon}}\right)} \\ &= \exp\left(\frac{\varepsilon |D - \mathrm{Sm}(Sg)|}{\Delta Sm} - \frac{\varepsilon |D - \mathrm{Sm}(Sg^{'})|}{\Delta Sm}\right) \end{aligned}$$

$$= \exp\left(\frac{\varepsilon(|D - \mathrm{Sm}(S_g)| - |D - Sm(S_g^{'})|)}{\Delta Sm}\right)$$
$$\leq \exp\left(\frac{\varepsilon(|Sm(S_g) - Sm(S_g^{'})|)}{\Delta Sm}\right)$$
$$\leq \exp\left(\frac{\varepsilon \cdot \Delta Sm}{\Delta Sm}\right) = \epsilon$$

Theorem 3. *The DGNDP algorithm satisfies ε-differential privacy.*

Proof of Theorem 3. In this algorithm, each sub-graph is handled by the GSEM algorithm and ADPRA algorithm, which all satisfy differential privacy. According to the principle of sequence combination in differential privacy, each sub-graph is preserved by differential privacy. After all the sub-graphs are merged into a complete directed graph, on the basis of the principle of parallel processing in differential privacy, it is evident that the GDNDP algorithm satisfies differential privacy. □

5. Experiments and Results

In this paper, the proposed method focuses on a special directed graph, which is a simple connected directed graph without self-cycles and node attributes. In this section, five real-world data sets, which describe five directed graphs are applied to demonstrate the efficiency of the proposed method. In privacy preservation, the change rate of the edge is utilized to evaluate the performance of methods. In data utility, the metrics of the graph are used to measure the effectiveness of methods. In addition, we compare the proposed method with other methods in [5,20]. The experiments are conducted on a Laptop with an Intel i7 3.5 Ghz and 8GB RAM, which works with Windos10 and Python 2.6.

5.1. Data Sets

(1) Physicians: This directed network captures innovation spread among 246 physicians in towns in Illinois, Peoria, Bloomington, Quincy, and Galesburg. A node represents a physician and an edge between two physicians shows that the left physician told that the right physician is his friend or that he turns to the right physician if he needs advice or is interested in a discussion. There are 240 nodes and 1098 edges.
(2) Blogs: This directed network contains front-page hyperlinks between blogs in the context of the 2004 US election. A node represents a blog and an edge represents a hyperlink between two blogs. There are 1224 nodes and 19,025 edges.
(3) Wikipedia−link: This network consists of the wikilinks of Wikipedia in the Gagauz language (gag). Nodes are Wikipedia articles and directed edges are wikilinks, i.e., hyperlinks within one wiki. There are 2929 nodes and 118,603 edges.
(4) Gnutella: This is a network of Gnutella hosts from 2002. The nodes represent Gnutella hosts, and the directed edges represent connections between them. There are 12,717 nodes and 51,525 edges.
(5) Twitter lists: This directed network contains Twitter user–user following information. A node represents a user. An edge indicates that the user represented by the left node follows the user represented by the right node. There are 23,370 nodes and 1,231,177 edges.

5.2. Metrics and Parameters

5.2.1. Metrics and Parameters in Privacy Preservation

To evaluate privacy preservation, a metric, the change rate of the edge is shown as follows.

$$CRE = \frac{M_e}{S_e} \times 100\%$$

where M_e denotes the sum of all edges that are added and deleted in this method and S_e represents the sum of edges of the synthesis graph. This metric indicates how much the original graph has been modified to generate a synthesis graph. The larger *CRC*, the better privacy preservation.

Moreover, three methods including the independent (*ki*, *ko*)-degree anonymity method in [5], *k*-anonymity method in [20] and the GDGMP method without micro-aggregation are used to compare with the proposed method. The method in reference [5] is a k-degree anonymity method without considering the direction of edges, which minimizes changes in degree sequences as much as possible. Compared with the method in reference [5], the method in reference [20] focuses on directed graphs and provides a k-degree anonymity method. As node differential privacy can provide stronger privacy preservation than k-anonymity methods, the proposed method achieves better privacy preservation for directed graphs.

Correspondingly, the privacy budget in experiments is set as the sum of $\varepsilon 1$ and $\varepsilon 2$, where $\varepsilon 1 = \varepsilon 2 = \varepsilon$. Meanwhile, ε is in [0.2, 0.5, 1.0, 1.5, 2] and *k*, the number of elements in a cluster, is the integer between 2 and 5, which is also used in *k*-anonymization. Due to the uncertainty of the noise, all data sets are executed 10 times by using the proposed method and other methods to average out the results.

5.2.2. Metrics and Parameters in Data Utility

In the graph structure measure, the edge intersection *EI* is the ratio of the edges in the original graph to edges in the perturbed graph, as shown below.

$$EI = \frac{|E \cap E^{'}|}{max(|E|, |E^{'}|} \times 100\%$$

In the properties of nodes, the betweenness centrality(*Cb*) is the fraction of the shortest paths that go through each node. Then, the closeness centralities based on the in-degree(*in-Cc*) and out-degree(*out-Cc*) are used to measure how many steps are required to access every other node from a given node.

5.3. Results and Discussion

5.3.1. Analysis of Privacy Preservation

At first, the proposed method is conducted in the five data sets and the results are kept in Table 1. As shown in Table 1, when *k* is 3 and ε is 1, the value of *CRE* in the Hamsterster friendships data set is 47.62, while that in the Gnutella data set is 49.76. In particular, the value of *CRE* increases along with the decrease of ε when *k* is fixed. For example, when *k* is 3, the value of *CRE* in the Wikipedia−link data set increases from 25.74 to 69.13 with ε decreasing from 2 to 0.2, while that in the Gnutella data set also changes from 36.32 to 71.38. The results show that the smaller the ε, the larger *CRE*, which indicates that the proposed method can gain better privacy preservation for data sets. In Table 2, when ε is 1, if the *k* increases from 2 to 5, the value of *CRE* in Wikipedia-link will decrease from 50.89 to 36.74, as does that in other data sets, which indicates that the value of *k* can affect the privacy preservation. It is clearer that the smaller *k*, the better privacy preservation.

Table 1. The value of CRC in proposed method when k = 3.

K	ε	Physicians	Hamsterster Friendships	Wikipedia−Link	Gnutella	Twitter Lists
3	0.2	72.32	74.38	69.13	71.38	68.18
3	0.5	61.25	58.23	57.79	60.32	56.67
3	1	50.13	47.62	57.21	49.76	45.23
3	1.5	43.62	41.87	38.11	43.98	40.62
3	2	38.78	36.05	25.74	36.32	34.17

Table 2. The value of CRC in proposed method when $\varepsilon = 1$.

K	ε	Physicians	Hamsterster Friendships	Wikipedia—Link	Gnutella	Twitter Lists
2	1	56.39	52.14	50.89	54.67	51.22
3	1	50.13	47.62	45.21	49.76	45.23
4	1	45.82	43.68	40.31	45.33	41.87
5	1	41.22	39.98	36.74	40.79	38.49

Then, the performance of the proposed method in Gnutella and Twitter lists is illustrated in Figures 4 and 5. In Figure 4, with ε increasing from 0.2 to 2, the value of *CRE* decreases from about 70 to about 30 regardless of the value of *k*, which indicates that the ε controls the degree of privacy preservation. In addition, no matter what the value of ε is, the value of *CRE* increases with *k* decreasing from 5 to 2, which shows that the micro-aggregation can control privacy preservation. In addition, the same results as that in Figure 4 are demonstrated in Figure 5, which implies that the proposed method also can be applied in the big network.

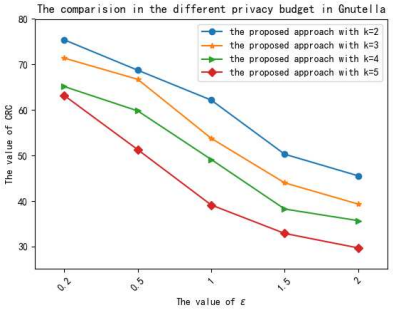

Figure 4. The comparison in the different privacy budgets in Wikipedia.

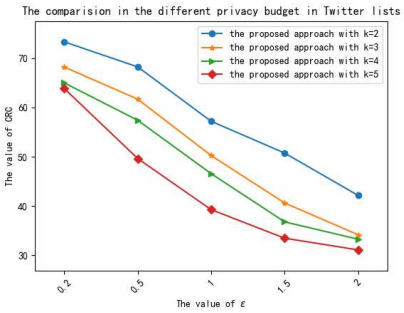

Figure 5. The comparison of the different privacy budgets in Twitter lists.

In the end, the proposed method is compared with other methods, and the results are illustrated in Figures 6 and 7. In Figure 6, when ε is 1, the value of *CRE* obtained by the proposed method is larger than that in (ki,ko)-degree anonymity method and *k*-anonymity method regardless of the value of *k*. Although the values in the (ki,ko)-degree anonymity method and *k*-anonymity method increase with the value of *k* rising, the proposed method provides better privacy preservation than these two methods. However, compared with the value of *CRE* in the proposed method without micro-aggregation, the value of *CRE* in the proposed method is smaller regardless of the value of *k*, which shows that the micro-aggregation can weaken privacy preservation. Moreover, in the data set Twitter lists, the same results as that in Figure 6 are shown in Figure 7. Therefore, the proposed method provides better privacy preservation than the two anonymity methods and has better data utility than the method without micro-aggregation.

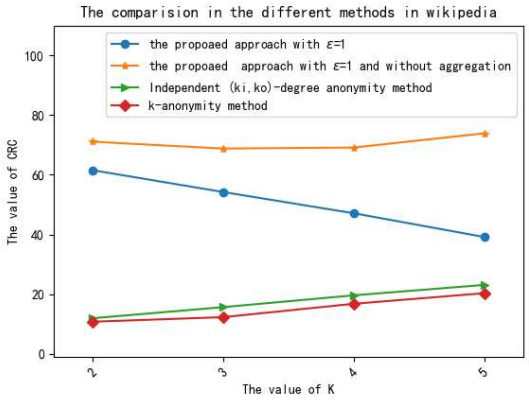

Figure 6. The comparison of the different methods in Wikipedia.

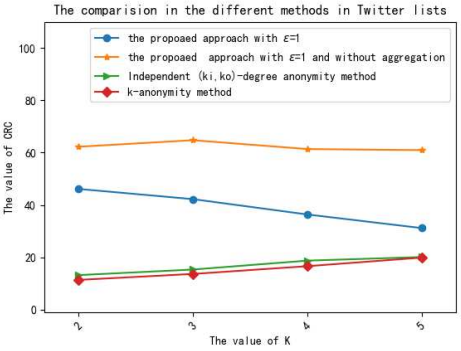

Figure 7. The comparison of the different methods in Twitter lists.

To sum up, the experiment results show that the proposed method can preserve directed graphs. In addition, micro-aggregation can be applied to control privacy preservation in this method.

5.3.2. Analysis of Data Utility

As shown in Figure 8, when the ε is 1, with k rising, the values of EI in five data sets all increase, which means more and more edges in the original graph retained in the synthetic directed graph generated by the proposed method. In Figure 9, as the ε is 1, the value of $\triangle$ *av-Cb* decreases gradually with the value of k increasing. The result indicates the property of nodes in the synthetic directed graph is close to that of the original directed graph. As illustrated in Figures 10 and 11, it is clear that the $\triangle$ *av-in Cc* and the $\triangle$ *av-out Dc* decline with k rising. To sum up, the results show that the proposed method can provide effective data utility.

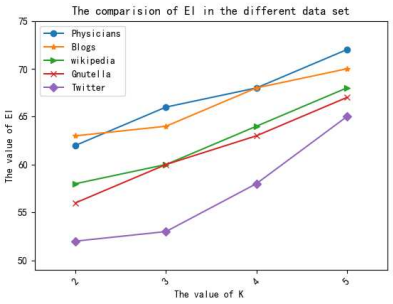

Figure 8. The comparison of EI in the different data sets.

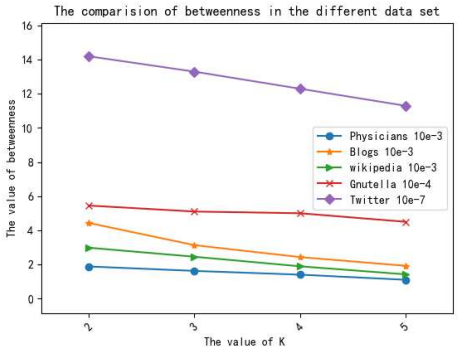

Figure 9. The comparison of betweenness in the different data sets.

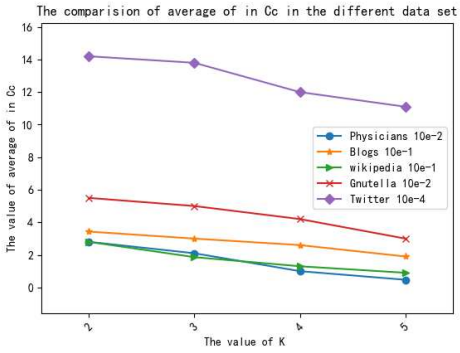

Figure 10. The comparison of the average of in Cc in the different data sets.

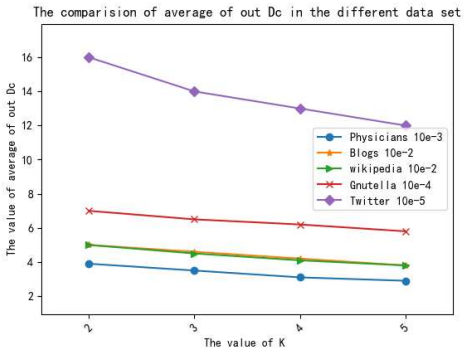

Figure 11. The comparison of the average of our Dc in the different data sets.

In particular, with the size of the network and the amount of data increasing, the computational overhead significantly increases because more and more edges and nodes are modified. In order to preserve a large directed graph, it is divided into many sub-graphs. Compared with the original directed graph, each sub-graph is much smaller. Therefore, the proposed algorithms can be well applied in these sub-graphs. In real-world deployments, for the scalability of the method, the scalability of the Louvain algorithm is mainly considered, which determines the scalability of the proposed method.

6. Conclusions

In this paper, to preserve directed graphs in MWNs, the DGNDP method is designed, which combines node differential privacy and graph modification. In this method, as node differential privacy can provide stronger privacy preservation than edge differential privacy and graph modification, it is used to add noise on degree sequences. Then, edge modification utilizes noised degree sequences to generate a synthetic directed graph, which

can strongly preserve the original directed graph. Additionally, to improve data utility, the original directed graph is divided into many sub-graphs, and the perturbations are only added in each sub-graph. In particular, the exponent mechanism is adopted to truncate degree sequences, which can ensure that the minimum noise is added to the degree sequences. Moreover, the ranking micro-aggregation effectively reduces the noise added to the degree sequences. According to the noised degree sequences, the relationship between two nodes is utilized to modify the edges of nodes, which can retain the original graph structure. Moreover, the theoretical analysis and the performance of experiments show that the DGNDP method not only satisfies ϵ-differential privacy but also retains data utility.

In this paper, we only focus on the simple static directed graph without considering node attributes. However, node attributes play an important role in the directed graphs. Thus, in the future, we will concentrate on the application of node differential privacy in complex attribute graphs. In addition, there is still a demand to achieve privacy preservation for dynamic directed graphs.

Author Contributions: Conceptualization, J.Y. and Y.Z.; methodology, J.Y.; software, J.Y.; validation, J.Y., Y.Z. and L.L.; formal analysis, J.Y.; investigation, J.Y.; resources, J.Y.; data curation, J.Y.; writing—original draft preparation, J.Y.; writing—review and editing, Y.Z.; visualization, J.Y.; supervision, Y.Z.; project administration, L.L.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

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

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Letaief, K.B.; Chen, W.; Shi, Y.; Shi, Y.; Zhang, J.; Zhang, Y.J.A. The roadmap to 6G: AI empowered wireless networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
2. Sharma, T.; Chehri, A.; Fortier, P. Review of optical and wireless backhaul networks and emerging trends of next generation 5G and 6G technologies. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4155. [CrossRef]
3. Van Hoboken, J.; Fathaigh, R.O. Smartphone platforms as privacy regulators. *Comput. Law Secur. Rev.* **2021**, *41*, 105557. [CrossRef]
4. Weichbroth, P.; Łysik, Ł. Mobile security: Threats and best practices. *Mob. Inf. Syst.* **2020**, *2020*, 8828078. [CrossRef]
5. Liu, K.K.; Terzi, E. Towards Identity Anonymization on Graphs. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008.
6. Casas-Roma, J.; Herrera-Joancomartı, J.; Torra, V. A survey of graph-modification techniques for privacy-preserving on networks. *Artif. Intell. Rev.* **2017**, *47*, 341–366. [CrossRef]
7. Ying, X.; Wu, X. Randomizing Social Networks: A Spectrum Preserving Approach. In Proceedings of the SIAM International Conference on Data Mining, SDM, Atlanta, GA, USA, 24–26 April 2008.
8. Mortazavi, R.; Erfani, S.H. GRAM: An efficient (k, l) graph anonymization method. *Expert Syst. Appl.* **2020**, *153*, 113454. [CrossRef]
9. Tai, C.H.; Yu, P.S.; Yang, D.N.; Chen, M.S. Privacy-Preserving Social Network Publication Against Friendship Attacks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 April 2011.
10. Zhou, B.; Pei, J.; Luk, W.S. A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. *ACM Sigkdd Explor. Newsl.* **2008**, *10*, 12–22. [CrossRef]
11. Tian, Y.L.; Zhang, Z.Y.; Xiong, J.; Chen, L.; Ma, J.F. Achieving graph clustering privacy preservation based on structure entropy in social IoT. *IEEE Internet Things J.* **2021**, *9*, 2761–2777. [CrossRef]
12. Zhang, H.; Lin, L.; Xu, L.; Wang, X. Graph partition based privacy-preserving scheme in social networks. *J. Netw. Comput. Appl.* **2021**, *195*, 103214. [CrossRef]
13. Dwork, C. Differential Privacy. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2006.
14. Jiang, H.; Pei, J.; Yu, D.; Yu, J.; Gong, B.; Cheng, X. Applications of differential privacy in social network analysis: A survey. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 108–127. [CrossRef]
15. Lan, S.; Xin, H.; Yingjie, W.; Yongyi, G. Sensitivity reduction of degree histogram publication under node differential privacy via mean filtering. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5621. [CrossRef]

16. Cheng, X.; Su, S.; Xu, S.; Xiong, L.; Xiao, K.; Zhao, M. A two-phase algorithm for differentially private frequent subgraph mining. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1411–1425. [CrossRef] [PubMed]
17. Ding, X.; Zhang, X.; Bao, Z.; Jin, H. Privacy-Preserving Triangle Counting in Large Graphs. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018.
18. Karwa, V.; Slavković, A. Inference using noisy degrees: Differentially private β-model and synthetic graphs. *Ann. Stat.* **2016**, *44*, 87–122. [CrossRef]
19. Iftikhar, M.; Wang, Q.; Lin, Y. dK-Microaggregation: Anonymizing Graphs with Differential Privacy Guarantees. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2020.
20. Casas-Roma, J.; Salas, J.; Malliaros, F.D.; Vazirgiannis, M. k-Degree anonymity on directed networks. *Knowl. Inf. Syst.* **2019**, *61*, 1743–1768. [CrossRef]
21. Zhang, X.L.; Liu, J.; Bi, H.J.; Li, J.; Wang, A.Y. Personalized K-InOut-Degree Anonymity Method for Large-scale Social Networks Based on Hierarchical Community Structure. *Int. J. Netw. Secur.* **2021**, *23*, 314–325.
22. Casas-Roma, J. Privacy-Preserving on Graphs Using Randomization and Edge-Relevance. In Proceedings of the Modeling Decisions for Artificial Intelligence, Tokyo, Japan, 29–31 October 2014.
23. Yu, F.; Chen, M.; Yu, B.; Li, W.; Ma, L.; Gao, H. Privacy preservation based on clustering perturbation algorithm for social network. *Multimed. Tools Appl.* **2018**, *77*, 11241–11258. [CrossRef]
24. Boldi, P.; Bonchi, F.; Gionis, A.; Tassa, T. Injecting uncertainty in graphs for identity obfuscation. *Proc. Vldb Endow.* **2012**, *5*, 1376–1387. [CrossRef]
25. Hu, J.; Yan, J.; Liu, Z.W.Z.H.; Zhou, Y.H. A Privacy-Preserving Approach in Friendly-Correlations of Graph Based on Edge-Differential Privacy. *J. Inf. Sci. Eng.* **2019**, *35*, 821–837.
26. Macwan, K.R.; Patel, S.J. k-NMF Anonymization in Social Network Data Publishing. *Computer J.* **2018**, *61*, 601–613. [CrossRef]
27. Medková, J. Anonymization of Geosocial Network Data by the (k, l)-Degree Method with Location Entropy Edge Selection. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Online, 25–28 August 2020.
28. Kadhiwala, B.; Patel, S.J. A Novel k-Anonymization Approach to Prevent Insider Attack in Collaborative Social Network Data Publishing. In Proceedings of the International Conference on ISS, Hyderabad, India, 16–20 December 2019.
29. Iftikhar, M.; Wang, Q. dK-Projection: Publishing Graph Joint Degree Distribution with Node Differential Privacy. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Delhi, India, 11–14 May 2021; Springer: Cham, Switzerland, 2021.
30. Sun, H.; Xiao, X.; Khalil, I.; Yang, Y.; Qin, Z.; Wang, H.; Yu, T. Analyzing Subgraph Statistics from Extended Local Views with Decentralized Differential Privacy. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019.
31. Lv, T.; Li, H.; Tang, Z.; Fu, F.; Cao, J.; Zhang, J. Publishing Triangle Counting Histogram in Social Networks Based on Differential Privacy. *Secur. Commun. Netw.* **2021**, *2021*, 7206179. [CrossRef]
32. Task, C.; Clifton, C. What Should We Protect? Defining Differential Privacy for Social Network Analysis. In *State of the Art Applications of Social Network Analysis*; Springer: Cham, Switzerland, 2014.
33. Karwa, V.; Slavković, A.B. Differentially Private Graphical Degree Sequences and Synthetic Graphs. In Proceedings of the International Conference on Privacy in Statistical Databases, Palermo, Italy, 26–28 September 2012.
34. Qin, Z.; Yu, T.; Yang, Y.; Khalil, I.; Xiao, X.; Ren, K. Generating Synthetic Decentralized Social Graphs with Local Differential Privacy. In Proceedings of Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017.
35. Xueqin, Z.; Qianru, Z.; Chunhua, G. Published Weighted Social Networks Privacy Preservation Based on Community Division. In Proceedings of the Conference on Communication and Network Security, Tokyo, Japan, 24–26 November 2017.