DDPLA: A Dynamic Differential Privacy Algorithm for Social Network Based on Local Community

Yuanpeng Long[1], Xianyi Zhou[2], Yang Li[3], Xuena Zhang[4], Bin Xing[5]*

[1] School of Economic Information Engineering, Southwestern University of Finance and Economics, China
[2] School of Computer Science & School of Software Engineering, Sichuan University, China
[3] Science and Technology on Security Communication Laboratory, Institute of Southwestern Communication, China
[4] School of Electronic Engineering, Chengdu Technological University, China
[5] Chongqing Innovation Center of Industrial Big-Data Co. Ltd., China
longpeng302@163.com, xianyi_zhou@163.com, yishuihanly@pku.edu.cn, zhangxuena99@163.com, xingbin@casic.com

Abstract

Social networks contain a large number of privacy information. Personal privacy will be jeopardized if network data without privacy protection is released directly. In view of the current privacy protection technology to protect the social network, there are some problems such as low maintenance of network structure or low accuracy of network data. In order to solve these problems, this paper proposes a dynamic differential privacy algorithm for social network based on local community (DDPLA). The algorithm can divide the social network into different communities, dynamically generate privacy budgets for different communities, and then generate uncertainty graphs. Experiments show that compared with other algorithms, the social network processed by DDPLA algorithm can better balance data utility and privacy protection. Furthermore, the algorithm can better protect important nodes.

Keywords: Differential privacy, Similarity, Community, Privacy protection, Social network

1 Introduction

The online social network provides people with a public network platform, accelerates the dissemination of information and strengthens the connection between people [1]. However, social networks are being threatened by information attacks such as data interception, information fraud, and privacy data leakage. The node attributes of social networks contain a large amount of personal identification information of users [2]. Once an attacker recognizes such identification information, it will cause privacy leakage. In addition, even if the attacker does not know the specific content of a certain node attribute, when he has other relevant background knowledge, he can also infer the attribute information of the node by analyzing the background knowledge [3]. Therefore, how to design privacy protection algorithms to avoid the privacy disclosure of social network data is a hot issue in current research.

Nowadays, many researchers have studied the problem of privacy disclosure in social networks, and some methods such as data anonymity and differential privacy are used to protect the data [4]. Hu et al. [5] proposed an algorithm called UGDP that converts the edge weights in the weighted network structure into probability values. The UGDP algorithm adds Laplace noise to the probability value to prevent the edge weight information from being intercepted by the attacker. Huang et al. [6] proposed an algorithm called PBCN that is based on clustering and noise. This algorithm proposed a non-interactive differential privacy scheme to optimize noise distribution to achieve a desirable privacy protection level while keeping commendable data availability and execution efficiency. Dong et al. [7] consider that some communities in social networks may have important, sensitive information. Clustering algorithm is widely used in data mining [8], text analysis [9] and medical diagnosis [10]. In the existing clustering methods, Ding et al. [11] designed a novel projection method, which is convenient to maintain the triangle count with a stricter definition on the basis of node differential privacy, which makes the original graph more useful. Day et al. [12] proposed two different methods based on the aggregated histogram and cumulative histogram, and realized the release degree distribution instead of publishing the reconstructed graph structure. At the same time, a projection mechanism is designed to effectively reduce the sensitivity. Onan [13] proposed supervised hybrid clustering, which is based on cuckoo search algorithm and k-means, to partition the data samples of each class into clusters so that training subsets with higher diversities can be provided, and the presented classifier ensemble outperforms the conventional classification algorithms and ensemble learning methods for text classification. In the field of machine learning, clustering and neural networks are used in popular fields such as emotion analysis [14]. Li et al. [15] proposed a new algorithm MB-CI based on differential privacy to protect the side information in the weighted social network structure. The algorithm focuses on the protection of the edge weight sequence, and realizes the differential privacy protection of the edge weight based on the histogram combined with the idea of adding less noise in the grouping.

*Corresponding Author: Xuena Zhang; E-mail: zhangxuena99@163.com
DOI: 10.53106/160792642023012401010

Toçoğlu et al. [16] proposed the combination of four classifiers of a text representation scheme (i.e. support vector machine, naive Bayes, logistic regression, and random forest algorithm) has achieved good text analysis results. Hajiabadi et al. [17] proposed a novel approach for general community detection through an integrated framework to extract the overlapping and non-overlapping community structures without assuming prior structural connectivity on networks. Yang et al. [18] proposed a methodology that allows them to compare and quantitatively evaluate how different structural definitions of communities correspond to ground-truth functional communities. In summary, the solutions provided by existing research have the following problems. Firstly, the protection of key nodes is hardly considered in social network protection algorithms, which only focus on protecting the structure of the whole graph, which will lead to a great chance for attackers to obtain the information of the key nodes in the graph. Secondly, most studies do not have a dynamic adaptation process for the allocation of differential privacy budgets, which will lead to a lot of budget waste. Finally, the accuracy of the community partition algorithm used in many privacy protection algorithms for community protection is low and too complex.

In this regard, in response to the above problems, this paper proposes a dynamic differential privacy algorithm for social networks based on local community (DDPLA). The DDPLA algorithm includes four stages. Firstly, the DDPLA algorithm uses structural similarity to calculate the local connection density of any two adjacent nodes in an undirected network, and then uses a local community dictation algorithm based on seed node pairs (LCDA) algorithm to divide the graph into communities. Secondly, we propose a dynamic privacy budget adaptation function to generate different privacy budgets for different communities. In the following third and fourth stages, we generate Laplace noise according to the corresponding budget and calculate the corresponding probability to generate the uncertainty graph. Simulation results show that compared with other algorithms, the social network processed by the DDPLA algorithm can better balance data utility and privacy. The uncertainty graph generated by this algorithm is more protected and better protects important nodes.

In a nutshell, the main contributions of this paper are summarized as follows:

(1) DDPLA (dynamic differential privacy algorithm for social networks based on local community) is proposed to release an uncertain graph to protect the graph of a social network, achieve better protection of important nodes and reduce the loss of data accuracy of ordinary nodes. Then we prove that the algorithm satisfies differential privacy.

(2) We propose a local community dictation algorithm based on seed node pairs (LCDA), and propose a dynamic differential privacy budget adaptation function to generate appropriate privacy budgets for different communities. At the same time, we provide a theory to prove the rationality of the function.

(3) We choose various real-world data sets and run a large number of comparative experiments. By comparing it with the latest algorithms in this research field, we verify the superiority of the proposed algorithm.

The content of this paper is arranged as follows: Section 1 briefly introduces the research background of the algorithm proposed in this work and summarizes the innovations of this paper. Section 2 introduces the relevant basic knowledge used in this paper. Section 3 describes the detailed implementation of the DDPLA algorithm, and several logical proofs are given to verify its rationality. Through comparative experimental analysis, Section 4 verifies the superiority of the DDPLA algorithm proposed in this paper. Finally, the conclusion is in Section 5.

2 Preliminaries

2.1 Differential Privacy Model

To solve the problem that existing privacy preserving algorithms depend on certain background knowledge, Dwork et al. [19] proposed a model called differential privacy. Compared with other algorithms and theories in the field of privacy protection, the differential privacy protection model provides a more standard definition for data privacy protection. Even if the attacker has obtained all the information except the target information, the model can ensure that the attacker cannot judge whether the target information is included in the attacked data. In short, the differential privacy model does not need to rely on any assumptions to achieve privacy protection, and can resist attacks with arbitrary background knowledge to the greatest extent.

The definition of adjacent dataset and differential privacy is shown in both definition 1 and 2.

Definition 1 (adjacent dataset [20]) If the structure of two data sets D and D' is similar, the difference between D and D' is only one record. That is, D and D' are adjacent data sets.

Definition 2 (ε-Differential privacy [20]) If a random algorithm M is given, all the output results of random algorithm m are represented by the set Range (M). For any two adjacent data sets D and D' satisfying definition 1 and any subset S contained in the result set Range (M), if M satisfies the inequality (1), then M satisfies ε-Differential privacy.

$$Pr[M(D)\in S]\le e^{\varepsilon}\times Pr\,[M(D')\in S]\ . \qquad (1)$$

Where Pr [E] represents the leakage risk of event E, which depends on the value of the random algorithm M. ε represents the privacy budget allocated by differential privacy protection. The lower the degree of privacy budget ε, the higher the degree of privacy protection. In particular, when ε = 0, the output of algorithm M cannot reflect any information related to the dataset. The value of ε can also reflect the level of disruption caused by the privacy protection algorithm to data. Under the same other conditions, the smaller the value ε is, the greater the disturbance degree of differential privacy protection to the original data is, and the worse the availability of the protected data is. The core of the implementation process of differential privacy is to inject random noise into the data that needs privacy protection, and finally output the disturbance value instead of the real

value. In this way, the privacy protection of sensitive data is realized. Among them, when the differential privacy disturbs the data, the amount of random noise added is not random. These noises need to meet certain allocations, and can be quantified by using sensitivity. Sensitivity can be divided into two categories, namely global sensitivity [14] and local sensitivity [21].

2.2 The Combination Property Models of Differential Privacy

For a series of differential privacy algorithms, the combination property of differential privacy ensures the overall privacy protection.

Property 1 (Sequence composability [22]) Assumes that given k random algorithms A_1, ..., A_k, each algorithm provides a privacy budget of ε_i-Differential privacy ($1 \le i \le k$).

For any data set D, the sequence combination algorithm $A_i(D)$ on data set D satisfies ($P\varepsilon_i$)-Differential privacy protection.

Property 2 (Parallel combinatorial [23])

Assumes that given K random algorithms A_1, ..., A_k, each algorithm provides a privacy budget of εi-Differential privacy ($1 \le i \le k$). For any data set D, the parallel combinatorial algorithm Ai(D) on disjoint data sets {D1, D2, ..., Dk} satisfies max(εi)-Differential privacy protection.

2.3 Uncertain Graph

Definition 5 (Uncertain Graph) We use G(V, E) to label a graph, if the mapping $P : V_P \rightarrow [0, 1]$ is the probability function of the existence of each edge in the edge set, then the graph G' = (V, P) is an uncertain graph about graph G, where VP represents all possible vertex pairs in set V, that is, $V_P = \{(V_i, V_j)\}$.

3 Algorithm Designing

3.1 Problem Solving Strategy

The existing social network privacy protection methods mainly involve three categories: data anonymity [26-28], data encryption [29] and differential privacy. Differential privacy is a privacy protection model that can resist strong background knowledge attacks. Therefore, this paper will study the privacy protection of social networks based on differential privacy protection technology.

When applying differential privacy to protect social network data, the focus is to balance the effectiveness of data and the effect of privacy protection, many existing studies achieve the effect of privacy protection by sacrificing the effectiveness of data, and few focus on the protection of key nodes. This paper considers how to protect key nodes on the premise of balancing data utility and privacy protection effects, so we first consider the community division of the network. By dividing similar nodes into the same community, it is convenient to consider the protection budget for the community as a unit.

In addition, in most studies, the privacy budget is a constant value, which means the protection of all nodes at the same level. However, the importance of each node in the graph is different. The fixed privacy budget makes the unimportant nodes waste the budget, while the important nodes are not better protected. Therefore, this paper will separately consider the protection levels of different communities. Through the dynamic differential privacy adaptation function proposed in this paper, each community will be allocated a privacy budget corresponding to the degree of protection, so as to obtain the corresponding level of protection. In addition, directly adding and deleting edges or nodes of the original graph will seriously affect the data utility of the original graph. However, by transforming the original graph into an uncertain graph, that is, the existence of edges in the original graph is determined by a probability, it can better protect the original structure. Therefore, this paper will take the original graph as the final release graph in the form of uncertain graph.

To sum up, the DDPLA algorithm will include the following four stages: Firstly, DDPLA proposed a local community dictation algorithm based on seed node pairs (LCDA) to divide communities. In the second stage, the privacy budget of each community will be calculated by using the dynamic privacy budget adaptation function. In the next stage, appropriate Laplace noise will be added to each community. Finally, the deterministic graph will be transformed into an uncertain graph based on the community. The main steps of the proposed algorithm in this paper are shown in Figure 1.

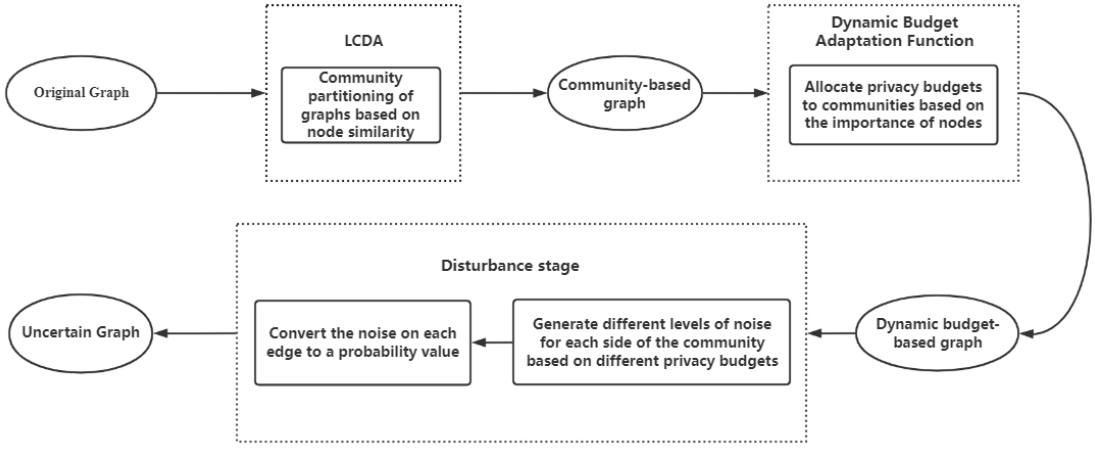

Figure 1. DDPLA algorithm execution flow chart

3.2 The LCDA Algorithm

For each node vi in the node set V, the node v_j closest to the node vi is more likely to belong to the same community. Therefore, we regard the node pair s = (v_i, v_j) as a seed node pair. As shown in Table 1, current researchers have proposed many measures of similarity between nodes, such as Jaccard similarity and Cosine similarity. In order to reduce the complexity of the algorithm, we use the formula (2) for calculating the similarity between nodes at the edge removal stage, where $N(v_i)$ represents the number of neighbors of vi and $N(v_j)$ represents the number of neighbors of v_j.

$$S(v_i, v_j) = |N(v_i) \cap N(v_j)|. \quad \textbf{(2)}$$

At this stage, a local community dictation algorithm based on seed node pairs (LCDA) will divide the graph into communities. Initially, the seed node pair is composed of node vi and the neighbor node v_j with the highest similarity among its neighbor nodes. The fitness function F will be used to construct a local community structure in the expansion phase. The description of the fitness function F is as follows.

Assuming it is an unweighted undirected graph, v_j is the selected node and the seed node pair s = (v_i, v_j) is the seed node pair constructed. The calculation formula of fitness function F is shown in the formula (3), where v_k is the node to be determined whether to join the community s, $N(v_k)$ is the number of neighbors of the v_k .

$$F(v_k, s) = \frac{|(N(v_k)) \cap s|}{|N(v_k)|}. \quad \textbf{(3)}$$

The above fitness function F has a value range of 0 to 1. When F = 1, all elements in s are neighbor nodes of v_k. The formula (3) describes the ratio between the neighbor nodes of v_k and the number of s, and it can reflect the degree of association between v_k's neighbors and the community s. By using the measurement method, this stage only selects nodes that can improve internal density for the expansion of coupling seeds. Therefore, when the F value of a node is large, the node will affect s, which in turn increases the local gain of the initial community.

The LCDA algorithm is mainly divided into four steps. Firstly, select a new seed node pair (v_i, v_j), then use the local community adaptability function to select the neighbors of v_j to expand the seed node pair based on the idea of greed. Then, the LCDA algorithm repeats the process until all sub-communities are established. Subsequently, nodes without coupling seeds will be assigned to the largest community. After that, in the community merger phase, clusters with a large number of shared nodes will be connected, so that the final community can be obtained. Here are the detailed implementation details of the four steps of the LCDA algorithm.

(1) Seed node pair phase

First, this phase sorts the node set V according to the input order of the nodes, and the sorting result is an ordered node set Vlist. Then, a new node vi is selected Sequentially from the ordered node set Vlist, then the most similar neighbor node of the selected node vi is calculated using formula (2), and a new coupled seed node pair is constructed. At this time, the selected node vi can be regarded as the core of a community or the center of the community, and the neighbor node with the highest similarity to the node vi will be used to help the community expand.

In order to reduce the time complexity of the algorithm, the algorithm will select the community center according to the input order of the nodes. The algorithm of the seed node pair stage is shown in Algorithm 1, and the expansion phase of fourth line refers to the community expansion phase of the LCDA algorithm.

Table 1. Different similarity metrics

Symbol	Definition
CN [23]	$score_{uv}^{CN} = \left\|N(u) \bigcap N(v)\right\|$
Salton [22]	$score_{uv}^{Salton} = \frac{\left\|N(u) \bigcap N(v)\right\|}{\sqrt{k_x k_y}}$
Jaccard [24]	$score_{uv}^{Jaccard} = \frac{\left\|N(u) \bigcap N(v)\right\|}{\left\|N(u) \bigcup N(v)\right\|}$
AA [20]	$score_{uv}^{AA} = \sum_{z \in \left\|N(u) \bigcap N(v)\right\|} \frac{1}{\log k_z}$
RA [21]	$score_{uv}^{RA} = \sum_{z \in \left\|N(u) \bigcap N(v)\right\|} \frac{1}{k_z}$
Deep Walk with Cosine (DWC) [25] Node2vec with Cosine (NWC) [26]	$\cos = \frac{\sum_{i=1}^{n} (u_i \times v_i)}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \times \sqrt{\sum_{i=1}^{n} (v_i)^2}}$

Algorithm 1. Seed node pair phase
Input: network G(V,E), the threshold τ
Begin
V_{list} ← sort the nodes in V according to their order of appearance inputted
select a node v_i from Vlist
select v_j, the most similar neighbor of v_i according to the node similarity
V_F ← Perform the expansion phase for $\{v_i, v_j\}$
$V_{list} = V_{list} - V_F$
resume the algorithm from the second line until no coupled seed is selected
End

(2) Community expansion phase

The main purpose of the expansion phase is to increase the neighbors of the community center node Vi, and build a primary community through iteration. This phase uses formula (3) to calculate the fitness function to help select the most suitable neighbor of newly joined nodes V_k to join the local coupled seed community s. Then, the new local community and newly joined nodes will continue to go through the expansion process. Finally, continue to repeat this process until all the remaining neighbors have not reached the threshold for joining the current local community. Among them, the threshold of the fitness function is 0.5. This makes it more likely that a node with a small number of neighboring nodes will join community s, while for a node with a large number of neighboring nodes, it is considered more reasonable to form a community with its neighbors if the number of nodes in s does not occupy 0.5 of all neighbors. The expansion stage is shown in the description of Algorithm 2, where i represents the number of the newly joined node.

Algorithm 2. Community expansion phase
Input: the graph G = (V, E), seed node pair $\{v_i, v_j\}$.
Output: a initial community
Begin
add v_j to the nodes list numList
i=0
while (i < the length of numList) then
v_i = numList[i]
for each $v_k \in$ neighbors of v_i
if $F(v_k, s) \geq 0.5$
add v_k to s
add v_k to numList
end if
end for
i = i + 1
end while
End

(3) Transmission phase

After the initial stage of building the community is completed, these communities are called initial communities at this time, then the remaining nodes that are not in the community need to be added to the most suitable initial community. In some cases, two nodes may not have neighbor nodes shared by both to form a seed node pair. The propagation phase assigns each of these nodes to the initial community with the largest neighbor node data.

(4) Community merger phase

The community merger phase will produce a final community based on the initial community. In the propagation phase, different communities may contain many common nodes. Research shows that there may be some overlapping nodes in social networks. Therefore, this phase will consider merging communities with a large number of shared nodes on this basis. For this reason, the phase needs to consider the conditions for community merging in the community merging stage. According to the experience of reference [30], we set the threshold to one-third of the total number of nodes in the minimum community. The community merger stage is shown in Algorithm 3.

Algorithm 3. Community merging phase
Input: initial community $C = \{c_1, c_2, ..., c_k\}$
Begin
sort $c_i \in C$ in descending order according to their length
for each $c_i \in C$
if exist y in C such that $(c_i \cap y) \geq (len(y)) \div 3$
add y to ci
end if
end for
End

3.3 Dynamic Budget Adaptation Function

The LCDA algorithm divides social networks into different communities, which contain different numbers of nodes, and the importance of nodes is different from each other. Generally speaking, if a user is very active in social network or occupies an important position in the society, he will have contact with many other users, and the degree of the node corresponding to the user will be very large. Therefore, we believe that the node with a large degree will be more important. In this paper, if the average degree of nodes in a community is greater than the average degree of the whole graph, then we think that this community is a key community. The dynamic budget adaptation function we proposed is as follows.

$$F = \begin{cases} (2 - \dfrac{N_G \sum_0^{N_C} d_{v_i}}{N_C \sum_0^{N_G} d_{v_j}})\varepsilon, & \text{if } 0 < \dfrac{N_G \sum_0^{N_C} d_{v_i}}{N_C \sum_0^{N_G} d_{v_j}} < 2 \\ 0.1\varepsilon, & \text{if } \dfrac{N_G \sum_0^{N_C} d_{v_i}}{N_C \sum_0^{N_G} d_{v_j}} \geq 2 \end{cases} . \quad (4)$$

In the formula (4), F is the dynamic budget adaptation function, N_G represents the number of nodes of the whole graph, and N_C represents the number of nodes of a certain community, d_{v_i} and d_{v_j} is the degree of the node, $\sum_0^{N_C} d_{v_i}$ represents the sum of node degrees of a certain community, $\sum_0^{N_G} d_{v_j}$ represents the sum of node degrees of the whole

graph. Furthermore, $\frac{N_G\sum_0^{N_C} d_{v_i}}{N_C\sum_0^{N_G} d_{v_j}}$ is the ratio of the community average degree to the whole graph average degree. The rationality of the formula is proved as follows.

Proof 1. We assume that $\frac{N_G\sum_0^{N_C} d_{v_i}}{N_C\sum_0^{N_G} d_{v_j}} = x$. Our dynamic adaptation function needs to make the key nodes get more protection, and appropriately reduce the interference of non key nodes so that they will not lose too much accuracy. In short, we allocate less than ε privacy budget to key communities and more than ε privacy budget to non key communities. At the same time, we can't deviate too much from the privacy budget ε submitted by users, so we consider that the value is 1.yε or 0.yε. The value range of x is $(0,1]$ or $(0,\max)$, and 2-x can make the value of ε at 1.yε or 0.yε. When $0 < x \le 1$, it indicates that the average degree of the community is less than the average degree of the whole graph, then the community is not a key community. At this time, the greater the value of x, the greater the importance of the community, and the smaller the value of 2-x. the closer the privacy budget is to ε, the greater the protection is. At the same time, the value of F is bigger than ε, which is reasonable. When the value of x is 1-2, it indicates that the community is an important community, and 2-x will be less than 1. At this time, a budget smaller than the regular budget will be added. If the average degree of the community is larger, x will be larger and 2-x will be smaller, so the privacy budget will be smaller and the protection level will be higher, which is reasonable. Finally, we think that the community whose average degree is more than twice the average degree of the whole graph is a very important community. At this time, we allocate a privacy budget to it, because in all communities, the number of communities whose average degree is more than twice the average degree of the whole graph is very small, so there will be no great damage to the original structure of the graph.

The DDPLA algorithm will use the Dynamic Budget Adaptation function to calculate the privacy budget for each community after the LCDA algorithm divides the community, so as to prepare for adding noise in the next step. The algorithm for this step is as follows (Algorithm 4).

3.4 Disturbance Stage

At this stage, we will add Laplacian noise to each community according to their respective privacy budget to meet the differential privacy. Then a probability value is generated according to the noise and assigned to the corresponding edge. Finally, we take the obtained uncertainty graph as the final release graph. After the implementation of the previous step, we have obtained the privacy budget adapted to each community. These privacy budgets are related to the importance of each community. The more important community will be allocated a smaller privacy budget, while the less important community will have a larger privacy budget. The DDPLA algorithm will generate noise at each edge of a node in the community according to each community's own budget, then convert this noise into probability. The specific algorithm is shown in Algorithm 5.

As shown in Algorithm 5, the input parameter communities is the output of the LCDA algorithm, budget_list is the calculation result of the dynamic budget adaptation function, and the data of communities and budget_list correspond one by one. At this stage, DDPLA use probability calculation function Pr[y] to generate probability p, and assign p to the corresponding edge e of node in a community. The probability calculation function Pr[y] is shown in formula (5), where b = Δf/ε and y is the noise generated by Laplace (Δf/ε), we assume that sensitivity Δf = 1.

Algorithm 4. Dynamic budget calculate

Input: the communities output by LCDA algorithm
Output: budget_list of comunities
Begin
for ci ∈ communities

$$x = \frac{N_G\sum_0^{N_C} d_{v_i}}{N_G\sum_0^{N_G} d_{v_j}}$$

if 0 < x < 2
add (2 − x)ε to budget_list[i]
end if
if x >= 2
add 0.1ε to budget_list[i]
end if
end for
End

Algorithm 5. Disturbance stage

Input: Comunities, budget_list, G(V, E)
Output: uncertain Graph G'(V, P)
Begin
processed_list = null
for the i-th community Ci in ∈ Comunities
budget = budget_list[i]
edges = the edges of Ci
for each v ∈ Ci
for e ∈ edges
while(y = Laplace(Δf/budget) < 0)
continue
p = Pr[y]
add p to edge e
end for
end for
end for
End

$$P_r[y] = \int_{-\infty}^{\frac{1}{y}} \frac{1}{2b} e^{-\frac{|x|}{b}} dx . \quad \textbf{(5)}$$

The rationality of the formula is proved as follows.

Proof 2. $P_r[y]=\int_{-\infty}^{\frac{1}{y}}\frac{1}{2b}e^{-\frac{|x|}{b}}dx=1-\frac{1}{2}e^{-\frac{\varepsilon}{y}}$, which means $P_r[y] > 0.5$, and if the privacy budget is small, the value of noise is wider, the greater the probability of getting a smaller value of $P_r[y]$. On the contrary, if the privacy budget is larger, the value of noise is relatively small, the greater the probability of getting a larger value of $P_r[y]$. In other words, if the privacy budget of the edge in the graph is very small, it means that it is an important edge, and the probability of it appearing in the uncertain graph should be small. On the contrary, the same is true, so the formula is reasonable.

3.5 Algorithm Privacy Analysis

This section will give the proof that the DDPLA algorithm satisfies differential privacy. Since the operation of converting noise value into probability value belongs to the post technology of differential privacy, it is only necessary to prove that the original graph to noise graph meet the differential privacy.

Proof 3. In the DDPLA algorithm, we take the community as the unit to add Laplace noise, and the data of each community has no intersection, so we can regard the community as an independent data set, that is, an independent graph, so we only need to prove that the processing of a community meets the differential privacy. Assume that f is the function that $f:G_c\to G'_c$, G'_c is the noise graph of the community c. G'_c and G_c are adjacent graphs with only one edge difference. P_{G1_c} represents the probability density function of $DDPLA(G_{1c},f,\varepsilon_c)$, P_{G2_c} represents the probability density function of $DDPLA(G_{2c},f,\varepsilon_c)$, then the following inequality holds.

$$\frac{P_{G1c}[G_{3c}]}{P_{G2c}[G_{3c}]}=\prod_{i=1}^{j}\left[\frac{\exp(-\frac{|f(G_{1c})i-G_{3ci}|}{\Delta f/\varepsilon_c})}{\exp(-\frac{|f(G_{2c})i-G_{3ci}|}{\Delta f/\varepsilon_c})}\right]$$
$$=\prod_{i=1}^{j}\exp\left(\frac{|f(G_{2c})_i-G_{3ci}|-|f(G_{1c})_i-G_{3ci}|}{\Delta f/\varepsilon_c}\right)$$
$$\le\prod_{i=1}^{j}\exp\left(\varepsilon_c\cdot\frac{|f(G_{2c})_i-f(G_{1c})_i|}{\Delta f}\right)$$
$$=\exp\left(\varepsilon_c\cdot\frac{\|f(G_{2c})-f(G_{1c})\|_1}{\Delta f}\right)$$
$$\le\exp(\varepsilon_c)$$

Assuming $\varepsilon=\max\{\varepsilon_c\mid c\in communities\}$, it can be seen from the parallel characteristics of differential privacy that the DDPLA algorithm meets ε-differential privacy.

3.6 Algorithm Calculation Cost Analysis

This section will analyze the computational cost of the DDPLA algorithm proposed in this paper. The DDPLA algorithm is mainly divided into two parts. The first part divides nodes into communities based on node similarity. The second part adds Laplacian noise to different communities according to a dynamic privacy adaptation function and converts the noise into edge probability value. Therefore, we will analyze the algorithm complexity of these two parts.

When dividing communities based on node similarity, the worst case is that the social network graph is a complete graph. Suppose there are n nodes in the graph. After selecting the seed node, when using the algorithm of the expansion stage for the selected seed node pair, the remaining n-2 nodes will be accessed. The fitness function value of the neighbor node of the seed node is shown in Formula (6).

$$F(v_k,s)=\frac{2}{\text{n-1}}. \tag{6}$$

For a social network graph, it is obvious that n is greater than 5, then all the neighbor nodes cannot reach the threshold of the local community s formed by the seed node pair, so the sequential node queue V_{list} to be selected will have n-2 nodes left. Similarly, n-2 nodes in the V_{list} queue will perform the same operation. At this time, the number of basic operations of the algorithm is $(\text{n-2})^2$, and the time complexity of the algorithm is $T(n)=O(n^2)$. However, the real-world social network graph cannot be a complete graph, so the complexity of the algorithm will certainly be less than this value.

When allocating different privacy budgets to divided communities, the DDPLA algorithm needs to traverse each edge of each community once. Therefore, in the worst case, when the social network is a complete graph, the number of basic operations of the algorithm is $\frac{n(n-1)}{2}$, and the algorithm time complexity of this part is $T(n)=O(n^2)$.

To sum up, the time complexity of the DDPLA algorithm is $T(n)=O(n^2)$. In addition, because the algorithm needs to store the information from the graph, the occupied space is directly proportional to the size of the graph, which means the spatial complexity of the algorithm is $S(n)=O(n)$, which is an acceptable value. And no matter how large the scale of the graph is, the time complexity remains the same, so it can be well applied to large-scale social network graphs.

4 Simulation Analysis

4.1 Simulation Settings

4.1.1 Experimental Environment

In this section, we use real data sets to analyze the proposed DDPLA algorithm, and evaluate the privacy protection effect of the algorithm and the utility of the generated social network. The experimental environment of this paper is shown in Table 2.

4.1.2 Experimental Data Set

This work selects two public real number sets and uses these data sets to verify the algorithm proposed in this paper. These data originate mainly from the social network field, but also include some other fields of data sets. These data sets are from the Stanford Large Network.

Table 2. Different similarity metrics

Experimental software and hardware environment information	Experimental software and hardware environment information
CPU Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41GHz	CPU Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41GHz
RAM 16.0 GB	RAM 16.0 GB
System type 64-bit Operating System, x64-based processor	System type 64-bit Operating System, x64-based processor
Operating System Win 10	Operating System Win 10

Table 3. Basic attributes of the experimental data set

Dataset	Number of nodes	Number of edges
Ego-Facebook	4039	88243
Musae-Github	5001	4402

Dataset Collection website [31]. The specific details of the data set are shown in Table 3.

(1) Ego-Facebook data set: this is a social network data set. The data comes from the Facebook APP used by the survey respondents. The data set contains a total of 4039 nodes and 84243 edges.

(2) Musae-Github dataset: this is a social network for GitHub developers. It consists of 5001 nodes and 4402 edges.

4.2 Evaluation Indicators

(1) Total number of network edges

In a social network graph, edge is an important factor in expressing the structural information of the original network graph, and can sensitively reflect the structural changes of the processed graph. Therefore, we chose to study the changes of edge directly to analyze the impact of the algorithm on the effectiveness of the original data. In a deterministic graph, the statistical formula of edges is as follows.

$$NE = \frac{1}{2}\sum_{v \in V} d_v . \tag{7}$$

In the formula (7), represents a member of the node set in a graph, and is the degree of the node . However, the existence of edges is expressed in the form of probability in uncertain graphs, so the formula can not be directly used in the uncertain graph. The formula of the degree of a vertex in an uncertain graph is shown in formula (8), and is equal to the sum of probabilities of its adjacent edges. Furthermore, the calculation formula of total edges in uncertain graphs is shown in formula (9).

$$d_v = \sum p(i, j) . \tag{8}$$

$$NE = \frac{1}{2}\sum_{v \in V}\sum_{u \in V/v} p(u, v) = \sum_{e \in E} p(e) . \tag{9}$$

(2) The change of degree of the key nodes

Because the algorithm DDPLA and comparison algorithm proposed in this paper meet the differential privacy, but DDPLA algorithm can provide better protection for important nodes, we propose the change of degree of the key nodes (CDKN) to analyze the privacy protection effect. In order to facilitate the experiment, we assume that nodes whose degree is greater than the average degree of the whole graph are important nodes, because they are more connected with other nodes, so they are more active.

When an algorithm is used to process a social network graph, the structure of the graph will change to achieve the effect of privacy protection, for example, adding or deleting an edge. In differential privacy protection, if you want to achieve a better privacy protection effect, you need to add more noise. At this time, the interference degree of data will become larger, and the degree of data distortion will become larger as well. If a node does not change before a comparison, the node is not specially protected. Based on this theory, we propose formula (10) to measure the privacy protection effect of the algorithm, and the calculation method for the uncertain graph is shown in formula (8).

$$CDKN = \sum_{v \in V_k}\sum_{u \in V/v} p(u, v) = \sum_{e \in E} p(e) . \tag{10}$$

4.3 Analysis of Simulation Results

4.3.1 Data Availability Analysis

The results of the three algorithms running on two data sets are shown in Figure 2 and Figure 3. We set 0.5 as the span of each level of privacy budget. When the budget is 0.1, it means that the data needs to be protected to a great extent. Therefore, a large amount of noise will be added to the original data, resulting in a large amount of interference to the data. When the budget is 3, the amount of noise added will become smaller, so that the data of the disturbed graph will be closer to the original graph.

As shown in the two graphs of experimental results, in general, with the increase of variance privacy budget ε, the number of edges of graph (NE) processed by different algorithms shows a growth trend. Based on the theory given in the previous paragraph, it can be seen that the experimental results in Figure 2 and Figure 3 are theoretically reasonable. Because a large number of nodes and edges will be added during the processing of the graph structure by the PBCN algorithm [6], and due to different execution strategies, the number of edges in the graph processed by the PBCN algorithm will be greater than that of the original graph. Because the algorithm strives to ensure the effectiveness of data, the change range of the curve corresponding to the algorithm is small. However, with the increase of privacy budget. The algorithm continuously adds edges to the original graph, which makes the graph structure that does not need to be protected lose its usability because the original data is destroyed. The UGDP algorithm [5] is also an algorithm for privacy protection by injecting probability into the original graph, and also meets the requirement for differential privacy. However, it only adds noise to the whole graph and does not consider the importance of each node in the graph. Therefore, the protection degree of this algorithm for network graph is lower than that of the DDPLA algorithm. Because the effect of privacy protection is inversely proportional to the data utility. Therefore, the curve representing the UGDP in the experimental results is slightly higher than that represented by DDPLA, but it can be seen from the figures that the gap between the two is not too large with the increase in the amount of data.

To sum up, because the DDPLA algorithm has a better protection effect on important nodes, it will be slightly worse than other algorithms in data utility. However, with the increase of data, the proportion of ordinary nodes also increases, and the data utility gap will become small.

4.3.2 Privacy Protection Analysis

The comparative experimental results of the DDPLA algorithm with the UGDP algorithm and PBCN algorithm are shown in Figure 4 and Figure 5. It can be seen from the experimental results that the edges of key nodes in the graph generated by PBCN algorithm do not change regularly, because it does not consider special protection for important nodes and will add edges or nodes randomly. At the same time, the UGDP algorithm does not consider the privacy protection of key nodes but only generates an uncertain graph to protect the whole graph information. The DDPLA algorithm proposed in this paper considers the special protection of key nodes, so with the increase of privacy budget, the number of edges of important nodes in the graph generated by the DDPLA algorithm will be less than that of the UGDP algorithm. Therefore, compared with the other two algorithms, the DDPLA algorithm can not only meet the differential privacy protection, but also play a special role in protecting key nodes. At the same time, combined with Figure 4 and Figure 5, it can be seen that when the scale of the data set increases, the ordinary nodes in a graph will also increase, and the degree of a key node will account for a larger proportion of the total degree of a community. Because the privacy budget added by the DDPLA algorithm to ordinary nodes will be slightly larger than the standard budget, the utility of data will increase.

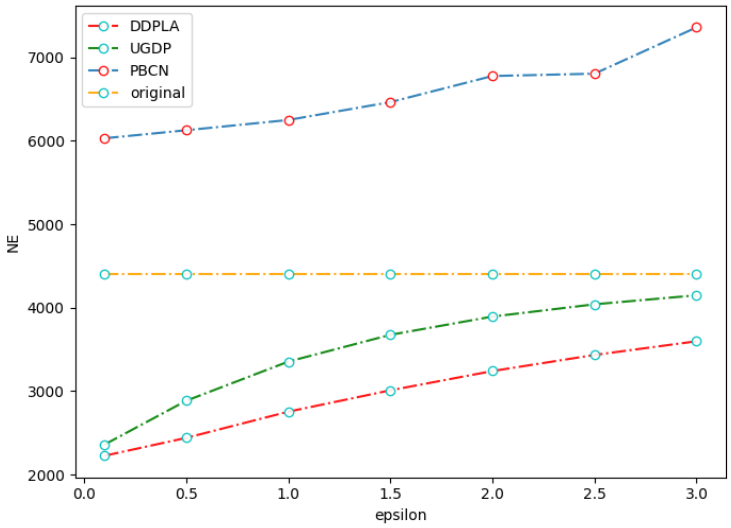

Figure 2. Comparison of NE of different algorithms on Musae-Github dataset

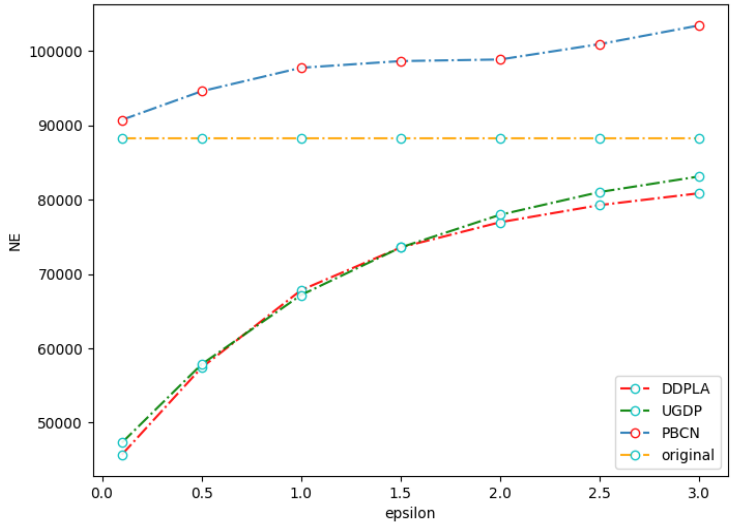

Figure 3. Comparison of NE of different algorithms on Ego-Facebook dataset

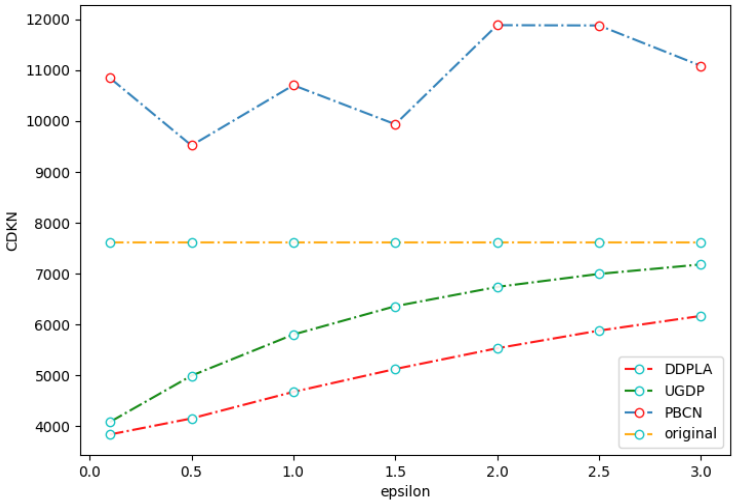

Figure 4. Comparison of CDKN of different algorithms on Musae-Github dataset

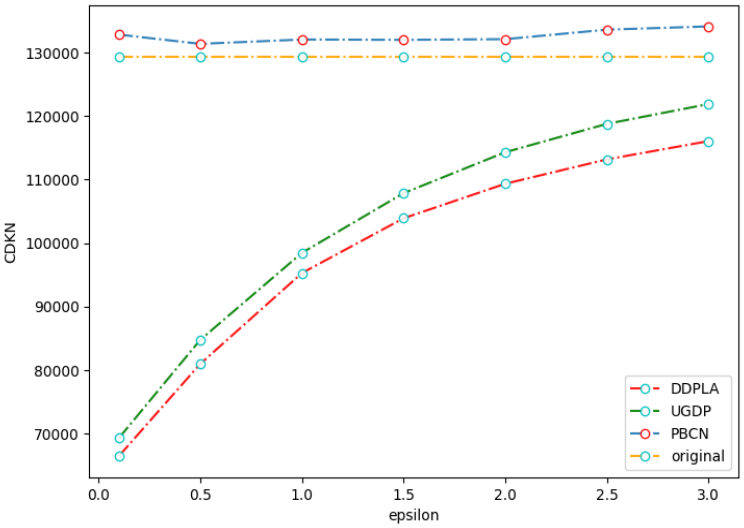

Figure 5. Comparison of CDKN of different algorithms on Ego-Facebook dataset

5 Conclusion

Social networks contain a lot of privacy information, if these privacy information are not protected, there will be a serious risk of privacy leakage. However, in the existing methods, more protection for key nodes is rarely considered when processing the network graph. At the same time, because the allocated privacy budget is fixed, the accuracy loss of ordinary nodes will be the same as that of important nodes, affecting the data utility. In order to solve this problem, this paper proposes a dynamic differential privacy algorithm for social networks based on local community (DDPLA). The algorithm can divide the social network into different communities, dynamically generate privacy budgets for different communities, and then generate uncertainty graphs, so less privacy budget is allocated for important nodes and slightly larger privacy budget is allocated for ordinary nodes, so as to reasonably control the amount of noise, achieve special protection for important nodes and reduce the loss of data of ordinary nodes. Finally, the simulation results show that the DDPLA algorithm has a better protection effect, and with the increase of the amount of data and the addition of ordinary nodes, the effectiveness of data will also improve. However, this algorithm also has some shortcomings. For example, if the amount of data in a graph is too large, the time complexity of the community partition algorithm proposed in this paper is large, so there is still room for improvement.

Data Availability

The data used to support the findings of this study can be accessed from http://snap.stanford.edu/.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62072319, the Key Research and Development Program of Science and Technology Department of Sichuan Province (No. 2020YFS0575), Sichuan University and Yibin Municipal People's Government University and City strategic cooperation special fund project (Grant No. 2020CDYB-29), the Luzhou Science and Technology Innovation R&D Program under Grant 2021CDLZ-11, the Chengdu Technology Innovation R&D Program under Grant 2021-YF05-02000-SN.

References

[1] L. Sheng, X. Guang, X. Ma, The Spread of Rumors and Positive Energy in Social Network, *Journal of Internet Technology*, Vol. 19, No. 5, pp. 1515-1524, September, 2018.

[2] U. Can, B. Alatas, A new direction in social network analysis: Online social network analysis problems and applications, *Physica A: Statistical Mechanics and its Applications*, Vol. 535, No. 1, Article No. 122372 December, 2019.

[3] F. Yu, M. Chen, B. Yu, W. Li, L. Ma, H. Gao, Privacy preservation based on clustering perturbation algorithm for social network, *Multimedia Tools and Applications*, Vol. 77, No. 9, pp. 11241-11258, May, 2018.

[4] P. Liu, H. J. Wang, S. Lin, X. X. Li, Social Network Anonymization via Local-perturbing Approach, *Journal of Internet Technology*, Vol. 19, No. 1 , pp. 247-256, January, 2018.

[5] J. Hu, J. Yan, Z. Q. Wu, H. Liu, Y. H. Zhou, 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, July, 2019.

[6] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, R. Wang, Privacy-preserving approach PBCN in social network with differential privacy, *IEEE Transactions on Network and Service Management*, Vol. 17, No. 2, pp. 931-945, June, 2020.

[7] K. Dong, Z. Liu, Y. Xu, Z. Li, Differentially private big data publication via structural inference and community detection, *2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*, Exeter, England, 2017, pp. 226-233.

[8] A. Onan, V. Bal, B. Y. Bayam, The use of data mining for strategic management: a case study on mining association rules in student information system, *Croatian Journal of Education: Hrvatski časopis za odgoj i obrazovanje*, Vol. 18, No. 1, pp. 41-70, January, 2016.

[9] A. Onan, S. Korukoğlu, H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, *Expert Systems with Applications*, Vol. 57, pp. 232-247, September, 2016.

[10] A. Onan, A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer, *Expert Systems with Applications*, Vol. 42, No. 20, pp. 6844-6852, November, 2015.

[11] X. Ding, X. Zhang, Z. Bao, H. Jin. Privacy-preserving triangle counting in large graphs, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Torino, Italy, 2018, pp. 1283-1292.

[12] W. Y. Day, N. Li, M. Lyu, Publishing graph degree distribution with node differential privacy, *Proceedings of the 2016 International Conference on Management of Data*, San Francisco, California, USA, 2016, pp. 123-138.

[13] A. Onan, Hybrid supervised clustering based ensemble scheme for text classification, *Kybernetes*, Vol. 46, No. 2, pp. 330-348, February, 2017.

[14] A. Onan, Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification, *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 5, pp. 2098-2117, May, 2022.

[15] X. Li, J. Yang, Z. Sun, J. Zhang, Differential privacy for edge weights in social networks, *Security and Communication Networks*, Vol. 2017, pp. 1-10, March, 2017.
[16] M. A. Toçoğlu, A. Onan, Sentiment analysis on students' evaluation of higher educational institutions, In: C. Kahraman, S. C. Onar, B. Oztaysi, I. Sari, S. Cebi, A. Tolga, (Eds) *International Conference on Intelligent and Fuzzy System*s, Springer, Cham, 2020, pp. 1693-1700.
[17] M. Hajiabadi, H. Zare, H. Bobarshad, IEDC: An integrated approach for overlapping and non-overlapping community detection, *Knowledge-Based Systems*, Vol. 123, pp. 188-199, May, 2017.
[18] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, *Knowledge and Information Systems*, Vol. 42, No. 1, pp. 181-213, January, 2015.
[19] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, In: S. Halevi, T. Rabin, (Eds), *Theory of cryptography conference*, Springer, Berlin, Heidelberg, 2006, pp. 265-284.
[20] Z. Ji, Z. C. Lipton, C. Elkan, Differential privacy and machine learning: a survey and review, *arXiv preprint*, December, 2014. https://arxiv.org/abs/1412.7584?context=cs
[21] P. Liu, Y. X. Xu, Q. Jiang, Y. Tang, Y. Guo, L. E. Wang, X. X. Li, Local differential privacy for social network publishing, *Neurocomputing*, Vol. 391, pp. 273-279, May, 2020.
[22] F. D. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, Providence, Rhode Island, USA, 2009, pp. 19-30.
[23] F. McSherry, K. Talwar, Mechanism design via differential privacy, *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, Providence, RI, United States, 2007, pp. 94-103.
[24] X. Zheng, Z. Cai, G. Luo, L. Tian, X. Bai, Privacy-preserved community discovery in online social networks, *Future Generation Computer Systems*, Vol. 93, pp. 1002-1009, April, 2019.
[25] J. Yu, H. Xue, B. Liu, Y. Wang, S. Zhu, M. Ding, GAN-Based Differential Private Image Privacy Protection Framework for the Internet of Multimedia Things, *Sensors*, Vol. 21, No. 1, Article No. 58, January, 2021.
[26] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 1, No. 1, pp. 3-es March, 2007.
[27] G. Natesan, J. Liu, An adaptive learning model for k-anonymity location privacy protection, *2015 IEEE 39th Annual Computer Software and Applications Conference*, Taichung, Taiwan, 2015, pp. 10-16.
[28] L. Karimi, B. Palanisamy, J. Joshi, A dynamic privacy aware access control model for location based services, *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, Pittsburgh, PA, USA, 2016, pp. 554-557.
[29] S. Q. Lai, X. L. Yuan, S. F. Sun, J. K. Liu, Y. H. Liu, D. X. Liu, GraphSE^2: An Encrypted Graph Database for Privacy-Preserving Social Search, *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19), Association for Computing Machinery*, Auckland, New Zealand, 2019, pp. 41-54.
[30] D. Chen, B. Wang, J. Zhou, Y. W. Tang, Multi-Scale Local Community Detection Algorithm Based on Structure Similarity, *Journal of the University of Information Engineering*, Vol. 16, No. 1, pp. 90-97, January, 2015.
[31] Stanford University. Available online: http://snap.stanford.edu/data/.

Biographies

Yuanpeng Long, Master of International Business, graduated from the Southwestern University of Finance And Economics in 2014. Doctoral candidate at Southwestern University of Finance and Economics. His research interests include Urban governance and Intelligent decision.

Xianyi Zhou is studying for his master's degree in the School of Computer Science from Sichuan University under the guidance of Professor Liangyin Chen. His research interests include cyberspace security and internet of things.

Yang Li received the B.S. degree in physics and the Ph.D. degree in radio physics from Peking university, Beijing, China, in 2007 and 2012, respectively. He is currently a Senior Engineer with the Department of Science and Technology on Communication Security Laboratory, Chengdu, China.

Xuena Zhang, Master of Communication engineering, Lecturer. Graduated from the ShenYang LiGong University in 2009. Worked in School of Electronic Engineering in the Chengdu Technological University. Her research interests include electronic information engineering, communication and information system. More than 2 papers and a book are published in 2020.

Bin Xing, Chief scientist of National Engineering Laboratory of industrial big data application technology, used to be an international cooperation engineer of DASSULT Group. The head of big-data analysis and processing department and the data analysis scientist of ATOS France.