The Journal of Supercomputing (2024) 80:7399–7432
https://doi.org/10.1007/s11227-023-05751-w

Entropy-driven differential privacy protection scheme based on social graphlet attributes

Jing Zhang[1,2] · Zuan-yang Zeng[1,2] · Kun-liang Si[1,2] · Xiu-cai Ye[3]

Accepted: 19 October 2023 / Published online: 7 November 2023

Abstract

The exponential growth of social networks has resulted in the generation of vast amounts of graph data containing sensitive information. However, the exposure of such data could lead to disastrous consequences. Current graph data protection algorithms lack sufficient research on the attribute characteristics of social users, which results in a failure to safeguard potentially vulnerable private data effectively. To address these issues, an entropy-driven differential privacy protection scheme based on social graphlet attributes (EDP-SGA) is proposed. Firstly, a matrix-based algorithm is proposed for constructing an attribute intimacy matrix, which can quantify the strength of links among social users' attributes. Secondly, an influence algorithm based on user node attributes and information entropy is proposed, which can divide social networks into communities and select seed nodes. Thirdly, a privacy-preserving social network data publishing algorithm is proposed, which can combine graph modification techniques and differential privacy to convert sensitive graph data into an uncertain graph. Finally, experimental results demonstrate that the EDP-SGA can keep the balance between the privacy and the utility of social graph data.

Keywords Information entropy · Differential privacy · Graphlet · Graph modification · Privacy protection · Uncertain graphs

✉ Jing Zhang
jing165455@126.com

Zuan-yang Zeng
zengzuanyang@gmail.com

Kun-liang Si
kunliangsi@outlook.com

Xiu-cai Ye
yexiucai@cs.tsukuba.ac.jp

[1] School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, Fujian, China

[2] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China

[3] Department of Computer Science, University of Tsukuba, Tsukuba 3058573, Japan

1 Introduction

The emergence of the internet has sparked exponential growth in social network services (SNS), leading to the development of various social networking applications. The network of collaborative relationships among people formed under social application software is called social network [1]. Social network is a social structure formed by the collection of social individuals and individual connection relationships based on information network. A large amount of graph data generated in social networks can visualize social changes and overall trends [2, 3]. At present, social graph data has been widely used in various fields of life, such as social recommendation systems, marketing influence maximization, social behavior research, etc. Social network graph data have enormous research value. Furthermore, some specific subgraph structures appear more frequently in real networks than in random networks, known as graphlet [4, 5]. Scholars have studied the evolution of social networks from a topological perspective and found that graphlet are one of the basic topological structures of networks and play an important role in network evolution [6].

Nevertheless, frequent privacy breaches, such as the leakage of personal data from 533 million Facebook users across 106 countries and over 235 million Twitter accounts had their personal information leaked and posted on an online hacker forum, are now common [7–9]. The leaked data include sensitive user attribute information like phone numbers, resumes, home addresses, names, and company addresses, as well as the relationships between user nodes, such as friend profiles, likes, reposts, etc. Through the analysis of these data, the sensitive information about individual users and their relationships with others may be inferred by adversaries. This can lead to informed speculation regarding a user's preferences, political attitudes, education level, and other details. Additionally, adversaries may exploit a user's friend list to extend their reach and cause significant harm [10]. Although social networks provide social services, the confidential information they possess is exposed [11, 12]. Ensuring secure data publishing while safeguarding user-sensitive information and contact relationships, has now become an urgent and important social issue that requires attention.

The social network can be formalized as a network topology graph, where nodes and edges represent social network users and connections, respectively. Social network graph not only includes sensitive user information and attributes but also includes vital relationships among user nodes. Moreover, various sensitive attributes can be associated by user nodes and have dissimilar degrees of sensitivity in their social associations [13]. The connections among user nodes are built on both strong and weak ties, and play a critical part in transmitting information within social networks [14–16]. These two types of relationship structures constitute necessary components of social networks are established by research [17, 18]. When adversaries gain access to parts of a social network graph structure, they can utilize this knowledge to pinpoint critical groups based on the strength of relationships between users and obtain valuable details. Thus, it is imperative to partition and safeguard both the link and the weak ties in social networks.

Various techniques have been proposed to protect graph data privacy in different situations [19–21], such as graph modification techniques, clustering methods, *K*-anonymity mechanism, etc. However, most of these solutions focus on protecting the topological structure between individual nodes in a social graph, which are limited to edge relationships between single nodes. These approaches lack a deeper understanding of node attribute information, overlook the links between users with different attributes, and thus fail to reflect the real interpersonal relationships among user nodes in a social network accurately. Therefore, an entropy-driven differential privacy protection scheme based on social graphlet attributes (EDP-SGA) is proposed.

Various techniques have been proposed to protect graph data privacy in different situations [19–21], such as graph modification techniques, clustering methods, *K*-anonymization mechanisms, etc. However, most of these solutions focus on protecting the topology between individual nodes in a social graph, limited to the edge relationships between individual nodes. Moreover, some techniques may be vulnerable to background knowledge attacks. Background knowledge includes specific information held by adversaries, which can be exploited for privacy-related attacks on published social network data. This information can be obtained through crawling or well-known web browser history stealing attacks or by participating actively in social networking sites or by exploring overlapping members of several social networking sites [22].

Nowadays, the imperative importance of privacy protection is pronounced increasingly. Trusted third party (TTP) plays a crucial role in data exchange and privacy preservation across various domains. Trusted third party is an independent and impartial entity with the primary responsibility of ensuring user privacy and data security during data processing and information transmission. Social Network Service Provided (SNSP) collects user information and performs initial anonymization before sending it to TTP. The TTP is responsible for privacy protection of both social user information and the social graph data. The EDP-SGA scheme proposed in this paper is based on privacy protection of social networks by trusted third party, which can defend against background knowledge attacks while protecting the social network graph structure. The system model of EDP-SGA is illustrated in Fig. 1. The principal contributions of this paper are as follows:

(1) An attribute intimacy matrix construction algorithm is proposed to quantify the intimacy between attributes of nodes in a social network. The strength of node links can be measured by the intimacy of user nodes' attribute features, thereby uncovering potential privacy leakage risks in the social network.
(2) An influence algorithm based on user node attributes and information entropy is proposed to measure the influence of nodes in the social network. Based on the attribute intimacy matrix of the social network and after dividing it into communities, seed nodes are selected. Seed nodes are special nodes with maximum influence in the social network, as starting points for protecting the privacy of the social network graph.

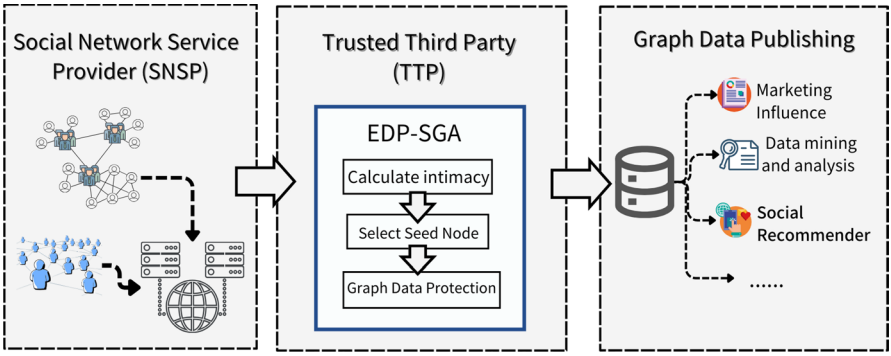


Fig. 1 System model of EDP-SGA consists of three parts: Social Network Service Provider (SNSP), Trusted Third Party (TTP) and Graph Data Publishing. **a** SNSP refers to organizations or platforms that offer social networking services to users. Examples include social media platforms like Facebook, Twitter, and others. **b** TTP is an impartial entity that verifies and secures transactions, ensuring the integrity of interactions between two parties. It sends social network data to the EDP-SGA scheme, ensuring privacy protection. **c** Graph Data Publishing can be applied to multiple domains such as social recommendation, data mining and analytics and marketing impact maximization, etc.

(3) A differential privacy-based graph data security publishing algorithm is proposed. Firstly, the graph structure of seed nodes is modified by using graph modification technology. Secondly, different privacy budgets are allocated to different strong or weak links based on the attribute intimacy by using differential privacy techniques. Finally, an uncertain graph is generated to publish social network graph data securely.
(4) EDP-SGA is evaluated across different scales of real social network data experimentally, and its effectiveness in protecting the privacy of social network graph data while maintaining good data usability is analyzed.

The rest of the article is organized as follows: The related work is given in Sect. 2; the Preliminaries is given in Sect. 3; the entropy-driven differential privacy protection scheme based on social graphlet attributes is designed in Sect. 4; Experimental results are shown in in Sect. 5; and finally, the conclusion is given in Sect. 6.

2 Related work

Graph data have become important increasingly in privacy protection and research across various fields, including social networks. Companies like Amazon and Netflix use graph data to enhance their recommendation services, while analyzing this data can provide valuable insight into social issues like communication patterns and information dissemination [23, 24]. Large-scale sharing of graph data can bring significant benefits to society, but can also lead to the disclosure of personal information. Therefore, many solutions related to social network graph data privacy protection are proposed. In this section, the privacy protection methods based on social

network graphs will be introduced in three aspects, which are graph modification, differential privacy, and uncertain graph.

2.1 Graph modification-based privacy method

The privacy protection method based on graph modification aims to achieve privacy protection by altering the local structure of the original graph. Graph modification is primarily accomplished through three techniques: random edge or node editing, random walk, and *K*-anonymous mechanism [19]. These methods modify the graph in different ways to break the link between sensitive information and the original graph.

Mittal et al. [25] proposed a method, which randomly walks from j, a neighbor of any node i, to u through the random walk, and replaces the original edge (i, j) with (i, u). *K*-anonymity is a classic limiting perturbation mechanism in graph modification, the core idea of *K*-anonymity is to ensure that after modifying the original graph, the probability of the attacker identifying a certain edge or node is no more than $1/k$ [26]. Xue et al. [27] proposed a probabilistic attack algorithm based on the random walk, which randomly flips the existence of each edge with probability p. Huang et al. [28] proposed a new privacy model (k, t)-privacy that combines the *K*-automorphism model for graph structure with the t-closeness privacy model for node label generalization. Mortazavi et al. [29] proposed the GRAM, an efficient (k, l) graph anonymization method based on edge addition. Tang et al. [30] proposed a $k - Vretr$ method to protect user query privacy and location privacy by combining the K-anonymity mechanism with the Voronoi diagram and quadratic residual hypothesis model. Ren et al. [31] proposed a novel graph privacy preservation mechanism, namely *kt*-safe graph. This approach can efficiently anonymize the graph by letting its n-hop neighbors contain the same or similar information.

However, modifying the topological structure of the original graph will destroy the original graph structure and cannot guarantee the availability of data. The method based on random walk cannot determine a reasonable walk step length, and the long walk length leads to low efficiency of the algorithm. If it is too short, the privacy of the graph data cannot be guaranteed. As for *K*-anonymous mechanism, relevant constraints need to be formulated, resulting in the inability to balance the efficiency of the algorithm and the availability of data, and it cannot resist background knowledge attacks such as link attacks and differential attacks.

2.2 Differential privacy-based privacy method

To solve the problem that *K*-anonymous mechanism cannot resist attacks such as background knowledge. Differential privacy protection model is proposed by Dwork [32], which defines the intensity of privacy protection on the basis of strict mathematics. Differential privacy is to perturb the data, so that whether a data exists in the data set does not affect the output [33].

Nguyen et al. [34] proposed an edge difference privacy method based on the adjacency matrix of the graph. By setting thresholds for edge noise, the algorithm

protects the privacy of the graph while preserving the edge density of the original graph. Li et al. [35] proposed an edging difference privacy algorithm for publishing edge weight sequences. By sorting edge weights and grouping them according to the array, Laplacian noise is added. Nguyen et al. [36] proposed a new privacy definition, called subgraph-differential privacy (subgraph-DP), for graph data publishing based on the conventional differential privacy definition. Adhikari et al. [37] controlled the size and quality of the edge set through different privacy budgets, and publish the edge set in the graph by using the exponential mechanism and sampling method proposed by Roohi et al. [38]. Ning et al. [39] designed a privacy protection algorithm for the weighted graph, and use the differential privacy protection model to protect the edge weight and structure of graph. Qu et al. [40] proposed a privacy protection method based on differential privacy uncertainty, called HPDU, which takes into account both edge and node degree privacy. Jian et al. [41] proposed two methods for publishing graphs under node-DP. One is the node-level perturbation algorithm which modifies the input graph by randomly inserting and removing nodes. The other one is the edge-level perturbation algorithm which randomly removes edges and inserts nodes.

Despite the differential privacy can mitigate background knowledge attacks effectively, if the assumptions about the background knowledge are restrictive overly, it may result in randomized outcome. This is due to the differential privacy methods introduce significant amounts of noise to achieve high levels of privacy, which can reduce the usability of data and even render graph data unusable greatly. Therefore, the main challenge for most differential privacy methods is how to ensure some data availability while reducing the addition of noise.

2.3 Uncertainty graph-based privacy method

Uncertain graph is emerged as a novel approach to safeguarding privacy. This method entails injecting different probabilities into the edges of an original graph prior to its release, thereby generating the uncertain graph that ensures privacy protection. By assigning probability values to the graph edges, this method safeguards privacy effectively while minimizing alterations to the original data. Consequently, uncertain graph offers a higher degree of data utility than methods involving complete edge removal or addition. As such, uncertain graphs offer superior privacy protection guarantee [42, 43].

Boldi et al. [44] proposed the $(k–\epsilon)$ obfuscation algorithm, which injects uncertainty into the social graph to achieve fuzzy processing. This method can also resist the attack of node identity on the premise of minimizing the distortion of graph structure. Yan et al. [45] proposed an uncertain graph method based on the theory of triadic closure, which involves adding edges to nodes. The method then injects different probabilities into these edges, thereby transforming the network into an uncertain graph. Yan et al. [46] proposed an improvement on the ternary closure method, selecting the top 10% nodes with the highest centrality in the social network as seed nodes and adding edges to them. Xu et al. [47] proposed an Uncertain Graph scheme based on Node Similarity (UG-NS), which can not only preserve user privacy in

social networks but also maintain high data utility. Wu et al. [43] proposed a privacy protection algorithm based on differential privacy (UGDP), which combines the differential privacy technology and the graph modification. UGDP adds edges to the original graph according to the theory of triadic closure, and uses the differential privacy to inject Laplace noise into the edge of the triangle, and finally generates the uncertain graph for data release. However, the ternary closure algorithm of UGDP cannot resist the background knowledge attack, the privacy of the social graph is still at risk of being leaked even after being disturbed [48]. Therefore, Zheng et al. [48] proposed a differential privacy algorithm of uncertain graph based on ternary closure(TCDP), which adds edge between two nodes to form a triangle according to the theory of triadic closure. The edges that form a triangle are noised and the remaining edges are assigned a value of 1. Finally, the uncertain graph is generated.

However, when the privacy protection level of the uncertain graph algorithm is high, the effectiveness of data availability is the worst, unable to balance privacy and data availability [49]. Moreover, existing uncertain graph methods modify the original topology without considering the diverse attribute relationships among user nodes in social networks. Consequently, there is a dearth of profound exploration into the attribute links between social network users, resulting in inadequate protection of sensitive information of users within social networks.

To summarize, graph modification-based privacy protection schemes are unable to guarantee data privacy and are vulnerable to background knowledge attacks. Differential privacy-based schemes struggle to balance the availability and privacy security of social network graph data, as high privacy protection can result in excessive noise addition that reduces the availability of graph data greatly. Uncertain graph protection schemes lack a thorough exploration of attribute information between social network users, making it impossible to fully protect their privacy and security. In response to these limitations, an entropy-driven differential privacy protection scheme based on social graph graphlet attributes (EDP-SGA) is proposed in this paper. A social network attribute intimacy matrix is constructed to partition the network into communities and identify high-influence seed nodes. The graph modification and differential privacy technology are utilized to protect the privacy of the graph structure of seed nodes. Lastly, the social graph is transformed into an uncertain graph for secure publishing.

3 Problem description and preliminaries

3.1 Motivation scenario

Social network graph data contains not only user-sensitive data but also connection relationships among users. With the web crawler, public datasets, and other ways, the adversary can infer user node attribute information, inter-user connection probabilities and identify special graphlet by social network part structure. The adversary could identify seed nodes by combining background knowledge and graph structure information, launch node entity identity re-identification attacks on anonymized

graphs, and further infer entities' semantic attributes, connectivity relationships, and other privacy information. Most graph data protection methods lack deeper mining of the network and have difficulty in balancing data privacy and availability.

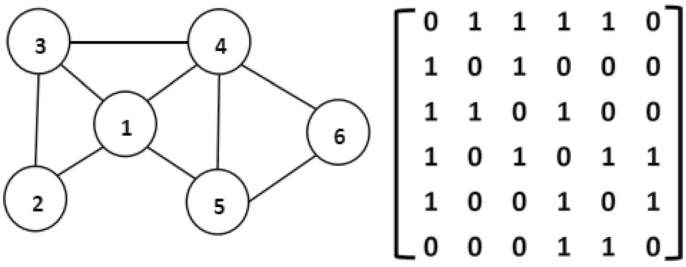
The EDP-SGA proposed in this paper is built in the social network application scenario, which aims to defend against background knowledge attacks and prevent the leakage of multi-attribute and structural information. EDP-SGA provides a new social graph protection scheme that meets the privacy requirements while preserving the statistical characteristics of the original data. The privacy objective of the EDP-SGA scheme proposed in this paper is to safeguard the attribute relationships, network structure of social network users, user information and social relationships, etc.

3.2 Problem definition

3.2.1 Social network

Definition 1 *(Social Network)* [50]: Let the social network graph be denoted as $G(V, E, S)$, where $V = \{v_1, v_2, \ldots v_n\}$ represents the set of user nodes in the network; E represents the set of edges in the social network G; S represents the set of attributes for social network users, where $v_i, v_j \in V$ denotes two users in the network, $e_{i,j}$ denote the friendship relationship between users v_i and v_j relationship $(e_{i,j}, e_{j,i} \in E)$, $|V|$ and $|E|$ denote the total number of nodes and edges in the network, respectively. A social user v_i $(v_i \in V)$, whose attribute set is set as $S_i \in S(1 \le i \le |V|)$, where $|S_i|$ denotes the total number of user attributes, where the attribute $atr_j \in S_i\,(1 \le j \le |S_i|)$ is for its corresponding attribute category $AC_j \in AC(1 \le j \le |AC|)$, where $|AC|$ denotes the total number of all the sets of categories in the social network G.

For example, AC={(AC_1=Music), (AC_2=Sport), (AC_3=Job)}; V={(v_1=Bob), (v_2 =Jane)}; atr_1={(AC_1=Country Music), (AC_2=Basketball), (AC_3=Engineer)}; atr_2 ={(AC_1=Jazz), (AC_2=Badminton), (AC_3=Teacher)}. Social network graph and corresponding adjacency matrix as shown in Fig. 2.



(a) Social network graph (b) Adjacency matrix

Fig. 2 Social network graph and corresponding adjacency matrix. A social graph can be represented by an adjacency matrix. In this matrix, there are edges among nodes 1, 2, and 5; while, node 1 is not connected to node 6. Therefore, node 1 can be represented as $A_1 = [0, 1, 1, 1, 1, 0]$

Definition 2 *(Attribute Similarity)* [51]: Attribute similarity reflects the intimacy between two nodes. The more identical attributes they share, the closer their social relationship is. The attribute similarity is defined as Eq. (1).

$$Sim(i,j) = \frac{s_i \cap s_j}{s_i \cup s_j} \tag{1}$$

where s_i and s_j represent the number of attributes for nodes i and j, respectively. The higher the similarity between s_i and s_j, the larger the value of $Sim(i, j)$, which ranges from 0 to 1.

Definition 3 *(Uncertain Graph)* [52]: Let $G(V, E)$ be a social graph, and $p : V_p \rightarrow [0, 1]$ be a mapping that represents the probability of presence of each edge, where $V_p = \{v_i, v_j \mid 1 \leq i < j \leq n \mid\}$ represents all vertex pairs in the vertex set V. The uncertain graph of G is denoted as $G' = (V, p)$.

Definition 4 *(Information Entropy)* [53]: Information entropy is a concept proposed by Shannon based on thermodynamics, which uses probability and statistical methods to quantify the degree of disorder in a system. Therefore, information entropy can be identified in the process of decay, and the larger the entropy value, the more valuable the information is, indicating a wider range and longer duration of information propagation. The information entropy in a network is expressed as Eq. (2).

$$E = -\sum_{i=1}^{n} I_i ln I_i \tag{2}$$

where $I_i = \frac{\sum_{i \in \Gamma i} w_{ij}}{Q_j}$ presents the importance of node v_i. In graph theory, information entropy reflects the local importance of nodes in the network. Based on whether the edges in the network are directed and weighted, the network can be divided into four types. In a weighted undirected network, as a measure of node importance, the edge weights are converted into node strengths [54]. The information entropy in the weighted undirected network is denoted as Eq. (3).

$$H_i = \sum_{j \in \Gamma v_i} \frac{\sum_{i \in \Gamma j} w_{ij}}{Q_j} \left(\frac{\sum_{j \in \Gamma i} w_{ij}}{Q_j} \log_2 \frac{\sum_{j \in \Gamma i} w_{ij}}{Q_j} \right) \tag{3}$$

where Γv_i represents the set of neighboring nodes of node v_i, and the adjacency strength value for node v_j is defined as $Q_j = \sum_{w \in \Gamma j} \sum_{i \in \Gamma j} w_{ij}$, where w_{ij} represents the weight of the edge between node v_i and node v_j.

3.2.2 Differential privacy protection models

Definition 5 *(ε-Differential Privacy)* [33]: Two adjacent datasets D and D' differ by at most one record. Let Z be a randomized query algorithm on D and D', and let

Range(*Z*) be the range of *Z*'s output. If the output $O \in Range(Z)$ satisfies Eq. (4), then *Z* satisfies ε-differential privacy.

$$Pr[Z(D) \in O] \leq e^{\varepsilon} \times Pr[Z(D') \in O] \tag{4}$$

Definition 6 *(Laplace Mechanism)* [33]: The Laplace mechanism mainly adds noise satisfying the Laplace distribution to the query result $f(D)$ through algorithm *Z*. For any function $f : D \to R^d$, if algorithm *Z* satisfies Eq. (5), then *Z* satisfies ε-differential privacy protection.

$$Z(D) = f(D) + (Lap_1(\triangle f/\varepsilon), \ldots, Lap_d(\triangle f/\varepsilon)) \tag{5}$$

where $Lap_i(\triangle f/\varepsilon)(1 \leq i \leq d)$ are independent Laplace variables, and the noise size is proportional to the function's global sensitivity and inversely proportional to ε. In other words, the larger the noise, the higher the global sensitivity of the function. The Laplace mechanism is mainly used for numerical outputs.

Definition 7 *(Composition of Differential Privacy Sequences)* [33]: If algorithm *Z* can be decomposed into multiple processes $Z_1, Z_2 \ldots Z_n$ all of which satisfy ε_i-differential privacy ($1 \leq i \leq n$), and act on the same dataset *D*, then algorithm *Z* satisfies ε–differential privacy, where $\varepsilon = \sum_{i=1}^{n} \varepsilon_i$.

3.2.3 Privacy attack model and definitions

Definition 8 *(Threat Model of Graph Privacy Attacks)* [44]: Adversary obtains the local structure of the network, which includes node degree, set of neighboring nodes, and user node attribute information through crawlers, publicly available datasets, and other means. The adversary can use the known information as an auxiliary graph to launch graph data privacy attacks on the social network. For example, adversary launches a graph structure re-identification attack based on node degree, neighbor node sets and utilizes user attribute information to launch a background knowledge attack on the network.

Definition 9 *(Graphlet)* [55]: Graphlet of *G* is denoted as $g_i = (V', E')$, $V' \subseteq V$ and $E' \subseteq E$. Graphlet is a subset of the vertices in graph G as well as all edges whose endpoints are both in this subset. Graphlet is composed of only a few nodes, primarily consisting of 3–4 nodes. Figure 3 shows all 3-node and 4-node graphlet structures. Graphlet statistical significance is evaluated by comparing the *Z*-score of the

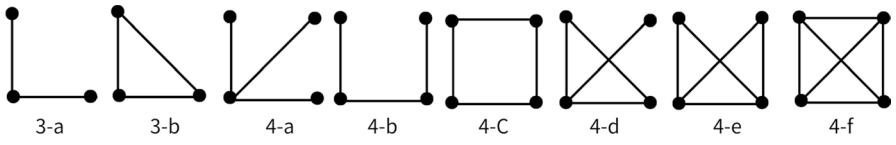


Fig. 3 3-Node and 4-node *graphlets*

computed subgraph with the randomized network. The Z-score is defined as shown in Eq. (6).

$$Z_i = \frac{(N_{readi} - \langle N_{randi} \rangle)}{std(\sigma_{randi})} \tag{6}$$

where N_{readi} denotes the number of occurrences of graphlet g_i in the real network, while $\langle N_{randi} \rangle$ and $std(\sigma_{randi})$ denote the average number of occurrences and the standard deviation of graphlet g_i in the set of random networks, respectively. A higher Z-score indicates the greater importance of the graphlet in the network. Among them, $4-f$ has the highest Z-score.

Definition 10 *(Graph Utility Measurement)* [44, 56, 57]: The graph data utility is for characterizing changes in the graph data structure, connection tightness, degree distribution, and so on. $d_v = \{d_1, d_2, d_3 \ldots d_n\}$ denotes the sequence of degrees of nodes in the graph, the number of edges denotes as $NE = \frac{1}{2}\sum_{v \in V} d_v$, the average degree of nodes denotes $AD = \frac{1}{n}\sum_{v \in V} d_v$, the variance of degrees of nodes denotes $DV = \frac{1}{n}\sum_{v \in V} (d_v - AD)^2$. Besides, Degree Distribution Entropy (DDE) is introduced as a metric to quantify network heterogeneity when the uncertainty in the distribution probability of node degrees with a specified edge number is considered. DDE is defined as $DDE = -\sum_{d=0}^{N-1} p(d) log p(d)$.

Definition 11 *(Edge Entropy)* [45]: The edge entropy in information entropy can measure the degree of privacy protection in the uncertain graph. The greater the edge entropy, the greater the uncertainty in the uncertain graph, which means stronger privacy protection for the graph. The definition of edge entropy is denoted as Eq. (7).

$$Ent_e = \sum_{e \in G\prime} -p(e_i) \times log_2 p(e_i) \tag{7}$$

where $p(e_i)$ represents the probability of the presence of the edge.

4 Entropy-driven differential privacy protection scheme based on social graphlet attributes

In order to protect the privacy and security of social network graph data and prevent adversaries from launching background knowledge attacks on social networks, it is crucial to protect not only the structure of the social network but also the attribute feature information of social users. As a solution, an entropy-driven differential privacy protection scheme based on social graph graphlet attributes (EDP-SGA) is proposed in this paper, which can protect the privacy of social graph data.

Step 1. An attribute intimacy matrix construction algorithm is offered, which calculates the attribute intimacy between nodes based on the similarity of their attribute values and constructs the attribute affinity matrix of the social network.

Step 2. An influence algorithm based on user node attributes and information entropy is proposed. The social network is partitioned into community structures based on the attribute intimacy matrix, and a seed node set that maximizes network influence using information entropy is selected.

Step 3. A privacy-preserving social network data publishing algorithm is proposed, which can mine the important social graph graphlet of the seed nodes. To protect the privacy of the graph data structure, the edge intimacy of the graph graphlet structure is perturbed using graph modification techniques combined with differential privacy technology. Finally, the algorithm transforms the graph into an uncertain graph for publishing.

4.1 Attribute intimacy matrix construction algorithm

Social network users have various social and characteristic attributes, which are interrelated with each other. Therefore, having the same attribute relationship between users affects the intimacy significantly among them. The attribute intimacy matrix construction algorithm aims to quantify the strength of relationships between nodes. Through the calculation of the similarity of node attributes, the attribute intimacy between nodes can be determined. This intimacy reflects the strength of the links among nodes and allows for the identification of potential privacy risks in their connections.

In Fig. 5, node A has three neighboring nodes: node B, C, and D.According to the node attribute list in Fig. 4b, the total number of attributes shared by node A and node B is $s_A + s_B = 8$, and they have 3 common attributes, $s_A \cap s_B = 3$. So, by using Eq. (1) for attribute similarity, the attribute intimacy between node A and node B can be calculated as $Sim(A, B) = \frac{|s_A \cap s_B|}{|s_A \cup s_B|} = 3/8$. Similarly, the attribute intimacy between node A and C can be obtained as $Sim(A, C) = 1/4$, and between node A and D as $Sim(A, D) = 1/4$. As there is no edge connecting node A with E, F, G, and H in Fig. 3a, the attribute intimacy can be calculated as $Sim(A, E) = Sim(A, F) = Sim(A, G) = Sim(A, H) = 0$. By following these steps, the attribute intimacy of all nodes can be calculated and the attribute intimacy matrix of the social network can be constructed. The attribute Intimacy Matrix Construction Algorithm is provided in Algorithm 1.

4.2 Influence algorithm based on user attributes and information entropy

Seed nodes in social networks are vital for identifying the key nodes involved in information propagation, making them a crucial starting point for safeguarding the structure of social networks. These nodes not only contain sensitive attribute information but also have the potential to maximize influence on other nodes within the network. Hence, an influence algorithm based on user node attributes and information entropy is presented in this section.

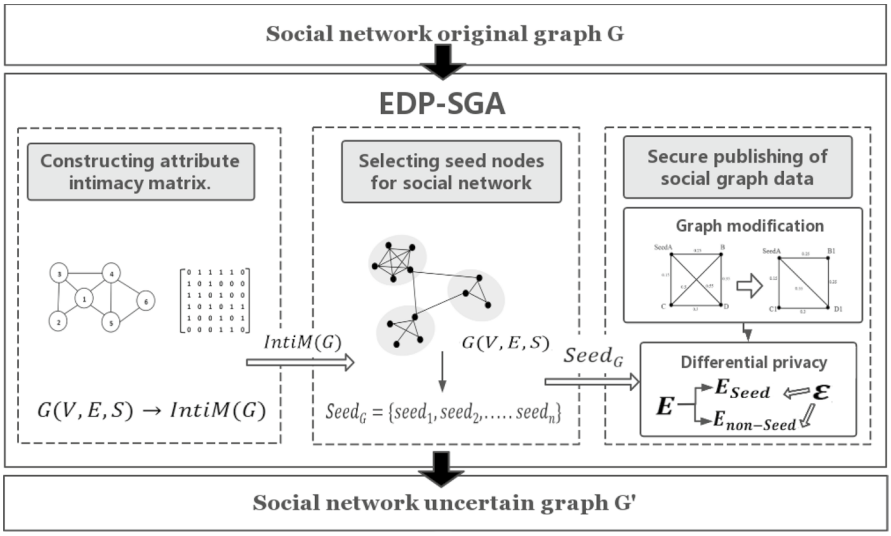
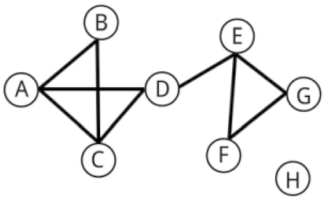


Fig. 4 The flowchart of the EDP-SGA. In EDP-SGA, the original social network graph is transformed into an adjacency matrix. Then, the graph is partitioned into communities, and important seed nodes are selected from the social network. Next, the important structural graphlets of the seed nodes are protected for privacy. Finally, the output is an uncertain graph of the social network after being processed by EDP-SGA



(a) Social network graph

$atr_A = [1,0,1,0,0,1,1,0]$
$atr_B = [1,1,1,0,0,1,0,0]$
$atr_C = [0,1,1,0,0,1,0,1]$
$atr_D = [0,0,1,0,0,0,0,1]$
$atr_E = [1,0,1,0,1,0,0,1]$
$atr_F = [1,0,0,0,0,0,0,1]$
$atr_G = [1,0,1,0,0,0,1,1]$
$atr_H = [0,1,1,0,0,0,0,0]$

(b) Attributes of nodes

Fig. 5 Social Network Graph $G(V, E, S)$. **a** the social network structure with 8 nodes. **b** the attributes of nodes in the social network. '1' denotes that the node has this attribute feature, while '0' indicates that the node does not have this attribute feature

The first step involves using a structural clustering algorithm for networks (SCAN) to partition the network into non-overlapping community structures, while also detecting bridge nodes and isolated nodes [58]. Bridge nodes represent connections between different communities that can facilitate information propagation across communities, leading to faster diffusion and wider spread. Isolated nodes are independent of other community structures and have no connection to other nodes. In the second step, the nodes with maximum influence within each community and all bridge nodes are added to the candidate seed set $Seed_G$. Finally, based on the

Input: Social graph $G(V, E, S)$
Output: Attribute intimacy matrix $IntiM$

A $\leftarrow G(V, E, S)$
Intialize $IntiM$
for i in V **do**
    Calculate node degree $deg[i]$ and attributes $Attr\ [S_i]$
end for
if $deg[i] \neq 0$ **then**
    Calculate the neighbor node set $neigh_i$ of i
    for i in $neigh_i$ and $A[i][j] \neq 0$ **do**
        $$IntiM[i][j] = \frac{\mid Atrr[i] \cap Attr[j] \mid}{\mid Atrr[i] \cup Attr[j] \mid}$$
Calculate the intimacy matrix
    end for
end if
Return $IntiM$
Return A

Algorithm 1 Attribute Intimacy Matrix Construction Algorithm.

Input: Social graph $G(V, E, S)$, attribute intimacy matrix $IntiM$, number of seed nodes N
Output: Seed node set $Seed_G$

Intialize $S, Seed_G, Seed_N$
if $\mid V \mid \neq 0$ **then**
    Community structure C, bridge nodes $C_B \leftarrow SCAN(G)$
end if
for c_i in C **do**
    for j in c_i **do**
        $d_j = \sum_{v \in N(j)} IntiM\,[j]\,[v]$
        $SimE_j = H_j$ //Calculate the intimacy entropy of nodes by Eq. (3)
        $Seed_G.add(SimE_{jmax})$
    end for
end for
Return $Seed_{topN}$

Algorithm 2 Influence algorithm based on user node attributes and information entropy

social network attribute intimacy matrix combined with information entropy, the influence of nodes in the candidate seed set $Seed_G$ is calculated, and the final seed node set $Seed_{topk}$ is selected. The specific process of selecting social network seed nodes is shown in Algorithm 2.

An example is given by Fig. 5 to describe the process of selecting seed nodes in social networks. First, the SCAN algorithm is used to partition the social graph in Fig. 4a into communities. The resulting community structures are: $C_1 = \{A, B, C\}$, $C_2 = \{E, F, G\}$, bridge node set $C_B = \{D\}$, and isolated node set $C_g = \{H\}$. The number of seed nodes N is set to 2, and the attribute intimacy between nodes is equivalent to the edge weight. Then, we can calculate the attribute entropy of each node within community C_1 and C_2 using Eq. (3). The attribute entropy of nodes within C_1 are as follows: $H_A = 0.58, H_B = 0.38, H_C = 0.56$. The attribute entropy

of nodes within C_2 is as follows: $H_E = 0.59, H_F = 0.46, H_G = 0.44$. Similarly, the attribute entropy value of bridge node D is $H_D = 0.61$, and the attribute entropy of isolated node H is $H_H = 0$. Next, we select node A with the highest attribute entropy value from C_1, and nodes E and D from C_1 and bridge node set C_B, respectively, to form the candidate seed set. Finally, we select the two nodes with the highest attribute entropy value, D and E, from the candidate seed set as the final seed nodes.

4.3 Privacy-preserving social network data publishing algorithm

Social network graphlet has enormous research value, and graphlet is the basic topological structure of social networks. Therefore, graphlet plays an important role in social network graph data. The EDP-SGA algorithm mainly consists of two phases in the privacy-preserving social network data publishing algorithm:

Phase 1. Graph Structure Perturbation: Set the sets $Graphlet_f$ and $Graphlet_c$ to store the seed node $4-f$ graphlet and $4-c$ graphlet, respectively. Perform edge additions on the social network graphlet structure based on the theory of triadic closure.

Phase 2. Graph Data publishing: To improve the privacy protection effect and ensure the secure publishing of graph data, convert the perturbed social graph attribute intimacy matrix into an uncertain graph. Inject uncertainty into the attribute intimacy between nodes to enhance privacy protection. The specific algorithmic process is shown in Algorithm 3.

In Algorithm 3, lines 1–11 aim to mine the graph graphlets of the seed nodes in the social network and create sets $Graphlet_f$ and$Graphlet_c$ to store 4-f graphlet and 4-c graphlet collections. Lines 12–20 modify the graphlet structure by adding edges to 4-c graphlets based on the ternary closure principle and perturbing their attribute intimacy values. Lines 21–27 perturb the edges formed by the seed nodes' graphlet structure by partitioning them into different sets according to their attribute intimacy values. $\theta = \frac{IntiM_{max} + IntiM_{min}}{2}$ is the threshold set for this purpose. The privacy budget is allocated as ε_1:ε_2:ε_3:ε_4=1:4:3:2. Different levels of noise are added based on the attribute intimacy value of each edge to prevent excessive noise from degrading data utility. Finally, lines 28–31 convert the perturbed attribute intimacy matrix into an uncertain graph with $p(i,j)(p \in [0,1])$.

In Algorithm 3, the modification of graph graphlets involves both edge addition and deletion. Figure 6 illustrates the protective process of graphlet deletion. Firstly, the 4-f graphlet structure of Seed-A is selected. Based on the intimacy values calculated according to Algorithm 1, the edge $E_{SeedA,D}$ with the highest intimacy value is removed. Subsequently, differential privacy protection is applied to the modified graphlet structure by injecting Laplace noise to perturb the intimacy values. Finally, the graph is transformed into an uncertain graph based on the perturbed intimacy values.

However, some constraints need to be added to the graph structure modification process to prevent a decrease in data utility. Specifically, the edges added or deleted cannot share common edges with the already modified edge sets. To protect sensitive node attributes and resist background knowledge attacks, the attribute intimacy

Input: Social graph $G(V, E, S)$, attribute intimacy matrix $IntiM$, seed node set $Seed_{topk}$, privacy budget ε,θ
Output: Uncertain graph G'

Intialize $Graphlet_f, Graphlet_c$
for i in $Seed_{topk}$ **do**
  Calculate the neighbor node set $neigh_i$ of i
  for v in $neigh_i$ **do**
    $Graphlet_f.add\,(g_{4f})$
    $EgdeGl_f \leftarrow$ all edge sets in $Graphlet_f$
    $Gl_f.add(\ IntiM[EgdeGl_f])$
  end for
  for a in $V_{non-Seed_{topk}}$ **do**
    if $dis\,(i,a) = 2, E\,(i,a) = \varnothing$ and $E(neigh_{v_1}, neigh_{v_2}) = \varnothing$ **then**
      $Graphlet_c.add\,(g_{4c})$
      $AddEgde.add\,([i,a])$
    end if
  end for
end for
for i in $Graphlet_c$ **do**
  $g \leftarrow$ the maximum value in G_f
  del $e \leftarrow$ the maximum value in g
  del $g_{4f} \leftarrow$ share common edges with e
  $Graphlet_f.remove\,(i)$
end for
for e in $AddEgde$ **do**
  $E(G) = E(G) + e$
  $IntiM[e] = g + laplace\,(1/\varepsilon_1)$
end for
for e in $EgdeGl_f$ **do**
  if $Gl_f\,(e) > \theta$ and $V_e \cap V_{seed_{topk}} \neq \varnothing$ **then**
    $IntiM\,(e) = Gl_f\,(e) + laplace\,(1/\varepsilon_2)$
  else if $Gl_f\,(e) > \theta$ **then**
    $IntiM\,(e) = Gl_f\,(e) + laplace\,(1/\varepsilon_3)$
  else
    $IntiM\,(e) = Gl_f\,(e) + laplace\,(1/\varepsilon_4)$
  end if
end for
for i, j in $IntiM$ **do**
  $p\,(i,j) \leftarrow Pr\,[IntiM\,[i]\,[j]]$
  Add the probability $p\,(i,j)$ to $E\,[i]\,[j]$
end for
Return $G' = (V, P)$

Algorithm 3 Influence algorithm based on user node attributes and information entropy

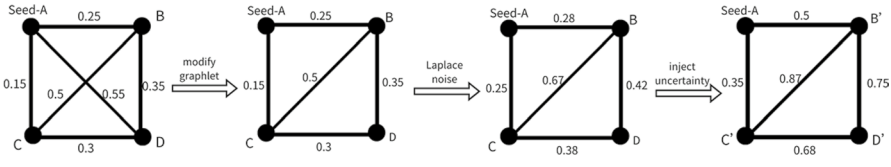


Fig. 6 Graphlet protection process

values of the seed nodes' edge sets are partitioned based on their sensitivity levels, and different levels of Laplace noise are added using differential privacy techniques to protect social network graph data comprehensively.

4.4 Complexity analysis and privacy statement

4.4.1 Complexity analysis

Theorem 1 *The time complexity of the EDP-SGA is* $O(n^2)$.

Proof The computation complexity of the EDP-SGA scheme is analyzed according to its process described by the aforementioned pseudo-codes of Algs. 1–3. Here, n, m, nei_n, N, C, c_n and g are used to denote the number of nodes, edges, neighbor nodes, seed nodes, community, nodes in the community and graphlet, respectively. Algorithm 1 calculates the intimacy between social users with the complexity of $O(n^2)$. Algorithm 2 involves selecting the seed set in the social network, with the complexity of $O(C \cdot c_n)$. Algorithm 3 is responsible for selecting the important graphlet structure for privacy protection based on the seed nodes, it has the complexity of $O(N \cdot nei_n + g)$. To sum up, the time complexity is $O(n^2)$. □

4.4.2 Privacy statement

Theorem 2 *The EDP-SGA satisfies* ε*-differential privacy*.

Proof The EDP-SGA scheme consists of three algorithms, where the privacy-preserving social network data publishing algorithm applies differential privacy techniques to perturb the edges of the graph graphlets. Assuming that graphs $G(V, E)$ and $G'(V', E')$ are attribute intimacy neighboring graphs in Algorithm 3, it can be concluded from Definition 5 that $V = V'$ and $E \oplus E' = 1$. Assuming that the edges of graphs G and G' differ from each other by e, then we have $E = E \cup e$, and the maximum difference between graphs $G(V, E)$ and $G'(V', E')$ in graphs $G(V, E)$ and $G'(V', E')$ caused by the edge e. The maximum difference between E and E' is 2, and by Definition 5, it can be obtained as $\Delta f = max_{G_1 G_2} \|f(G) - f(G')\|_1 = 2$. According to Definition 6, in the Laplace mechanism, the Laplace noise added by the query function f is $Lap(\Delta f/\varepsilon)$, where $\Delta f = 2$. □

In this paper, we set the privacy budget allocation ratio as $\varepsilon_1{:}\varepsilon_2{:}\varepsilon_3{:}\varepsilon_4 = 1{:}4{:}3{:}2$. privacy-preserving social network data publishing algorithm divides the privacy budget into four parts, as shown in lines 22–24 of Algorithm 3, adding noise as $Lap(\Delta f/\varepsilon_1)$ to the intimacy of the newly added edges. In lines 26–32 of Algorithm 3, the intimacy progression on the edges of the graph element is divided into sets of edges with different sensitivities based on the threshold θ. Noise $Lap(\Delta f/\varepsilon_2)$ is added to edges with high sensitivities, followed by $Lap(\Delta f/\varepsilon_3)$,

and $Lap(\Delta f/\varepsilon_4)$ to the set of edges with less sensitivities. From Definition 7, it follows that the algorithm satisfies $(\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4)$-differential privacy preservation and the privacy-preserving social network data publishing algorithm has an overall privacy budget $\varepsilon=\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4$ so that EDP-SGA satisfies ε-differential privacy preservation.

The level of privacy protection provided by the algorithm depends on the privacy budget, denoted as ε. A larger privacy budget means a narrower range of noise values and weaker privacy protection, while a smaller privacy budget means a wider range of noise values and stronger privacy protection.

5 Experiments

5.1 Experimental data

Facebook [59]: The SNAP Facebook dataset contains interpersonal relationships and multiple attributes of users, which can be applied to influence analysis, privacy protection, user behavior prediction, and other fields. This social network dataset consists of a social relationship dataset (.edege) and a node attribute dataset (.feature), where the attribute dataset contains a large number of attribute dimensions. These attributes have been anonymized to protect personal information. The dataset contains 10 anonymized networks, 6 of which are selected for this paper.

LastFM [60]: A social network dataset comprises LastFM users and was obtained from the public API in March 2020. The nodes represent LastFM users hailing from various Asian countries, and the edges represent mutual follower relationships among them. The node attributes are generated based on the artists that users have liked. In this paper, 500 attribute values are extracted from node attributes for calculation.

Table 1 shows the properties of the eight datasets.

Table 1 Facebook dataset list

Social network	Node number	Edge number	Attribution number
Facebook_0	347	5038	224
Facebook_107	1045	53,499	574
Facebook_686	170	3312	63
Facebook_1912	755	60,050	480
Facebook_3980	59	292	42
Facebook_3437	547	9626	262
Facebook	4039	88,234	574
LastFM	7624	27,806	500

5.2 Evaluation metrics

5.2.1 Privacy evaluation metric

The privacy protection scheme proposed in this paper transforms the original graph into an uncertain graph by assigning different probabilities to the edges, making it uncertain highly. The edge entropy in information entropy can measure the degree of privacy protection in the uncertain graph. The greater the edge entropy, the greater the uncertainty in the uncertain graph, which means stronger privacy protection for the graph [43]. The definition of edge entropy is denoted as Eq. (8).

Edge Entropy:

$$Ent_e = \sum_{e \in G\prime} -p(e_i) \times log_2 p(e_i) \tag{8}$$

5.2.2 Data utility metric

The sequence of node degrees $d_v = \{d_1, d_2, d_3 \dots d_n\}$ is some random variables. The degree of a node in an uncertain graph is represented by its expected degree, defined as the sum of the probabilities of the edges connected to any node v within the set V. The sequence of node degrees $d_v = \{d_1, d_2, d_3 \dots d_n\}$ is some random variables, which is given in Eq. (9).

$$d_v = \sum p(i,j) \tag{9}$$

In the uncertain graph, Number of Edges (NE') is denoted as Eq. (10). Average Degree of Nodes (AD')is denoted as Eq. (11). DV computed in the certain graph is in the same way as DV' computed in the uncertain graph, which are denoted as Eq. (13).

Number of Edges (NE):

$$NE' = \sum_{e \in E'} p(e) \tag{10}$$

Average Degree of Nodes(AD):

$$AD' = \frac{2}{n} \sum_{e \in E'} p(e) \tag{11}$$

Degree Variance of Nodes (DV):

$$DV' = \frac{1}{n} \sum_{v \in V} (d_v - AD)^2 \tag{12}$$

Degree Distribution Entropy (DDE') is proposed to measure the utility of graph data [47, 57]. DDE' degree distribution entropy in the uncertain graph, which is denoted as Eq. (13).

Degree Distribution Entropy (*DDE*):

$$DDE' = -\sum_{d=0}^{N-1} p(d) log p(d) \tag{13}$$

5.3 Experimental analysis

To verify the effectiveness of the EDP-SGA scheme, experiments are conducted on the datasets listed in Table 1 to compare EDP-SGA with the Uncertain graph method based on the ternary closure, an uncertain graph approach based on important nodes method, and the Transitive Closure-based Differential Privacy (TCDP) algorithm.

Uncertain graph method based on the ternary closure algorithm (UGTC) utilizes the transitive closure principle to randomly select nodes for edge addition and assigns different probabilities to the triangle edges formed after edge addition, making the resulting graph uncertain [45].

An uncertain graph approach based on important nodes (UGIN) selects the top 10% nodes based on their centrality value as seed nodes, then uses the transitive closure principle to add edges to the neighboring nodes that have not formed triangles in the seed nodes, and finally assigns probability values to the formed triangles to create an uncertain graph [46].

Differential privacy algorithm of uncertain graph based on ternary closure (TCDP) improves on the transitive closure method by adding noise to the edges of the triangle before assigning probability values, using differential privacy techniques [48]. The resulting uncertain graph can resist attacks based on background knowledge, enhancing data privacy protection.

In order to verify the privacy protection effect of uncertain graphs, Eq. (8) is used to measure the algorithm's privacy protection level on social network graphs in terms of edge entropy. According to Eq. (8), the higher the edge entropy, the higher the uncertainty of the graph and the better the protection effect.

5.3.1 Data privacy analysis

Tables 2, 3, 4 and 5 shows the changes in edge entropy for the EDP-SGA, TCDP, UGTC, and UGIN, respectively. Both EDP-SGA and TCDP algorithms use differential privacy technology to add Laplace noise to uncertain graphs, so their edge entropy values are also affected by the number of nodes and privacy budget ε. The results presented in Tables 2 and 3 indicate that as the privacy budget ε increases, the edge entropy of both EDP-SGA and TCDP algorithms decreases when the privacy budget ε is smaller. However, for privacy budgets within the range of $\varepsilon \in [0.1, 1]$, the edge entropy value of EDP-SGA is 4–19 times higher than that of the TCDP

Table 2 Variation of edge entropy in EDP-SGA

ε				
$\mid V \mid$	$\varepsilon = 0.1$	$\varepsilon = 0.3$	$\varepsilon = 0.5$	$\varepsilon = 1$
59	41.38	33.63	30.14	28.83
170	646.03	616.32	605.81	550.11
347	924.86	912.16	880.58	775.85
547	1690.66	1709.89	1661.26	1498.17
792	5073.59	5004.50	4791.10	4381.40
1045	10,400	10320.53	9981.11	8936.21
4039	99022.53	90832.23	89562.43	78932.32
7624	164365.43	144365.43	124365.43	93293.23

Table 3 Variation of edge entropy in TCDP

ε				
$\mid V \mid$	$\varepsilon = 0.1$	$\varepsilon = 0.3$	$\varepsilon = 0.5$	$\varepsilon = 1$
59	10.94	8.47	12.54	9.98
170	80.18	75.93	60.59	43.64
347	136.15	125.92	116.72	79.09
547	244.53	219.85	196.16	129.86
792	362.50	311.10	298.97	288.71
1045	732.99	628.67	511.66	487.65
4039	7073.87	6073.87	3791.97	2876.0
7624	12859.68	8205.79	6098.85	5275.19

Table 4 Variation of edge entropy in UGTC

$(m * c)$				
$\mid V \mid$	$c = 0.1$	$c = 0.3$	$c = 0.5$	$c = 1$
59	4.37	6.59	17.91	28.15
170	15.40	20.44	53.73	79.48
347	42.35	54.11	70.16	117.78
547	71.31	91.06	124.89	211.90
792	101.58	126.43	197.24	266.34
1045	157.43	283.67	329.62	427.28
4039	1132.9	2132.995	3516.07	6110.04
7624	1589.73	2676.94	2919.73	4700.90

algorithm. This disparity can be attributed to the random selection of nodes and injection of Laplace noise into the edges of triangles by the TCDP algorithm. While this approach can prevent background knowledge attacks, its single privacy budget cannot provide a controllable level of privacy. In contrast, the proposed EDP-SGA algorithm selects important seed nodes based on user attributes and information

Table 5 Variation of edge entropy in UGIN

$(m * c)$				
$\mid V \mid$	$c = 0.1$	$c = 0.3$	$c = 0.5$	$c = 1$
59	5.08	6.79	9.36	11.01
170	32.22	32.25	35.14	39.22
347	61.52	63.77	64.21	67.91
547	124.04	122.18	120.02	117.45
792	152.75	153.79	158.95	166.88
1045	189.16	209.70	215.43	218.19
4039	296.9	466.9	564.46	896.9
7624	786.43	9676.94	1676.94	28287.0

entropy, modifies their graph structure, and then leverages differential privacy technology to inject different privacy budgets into the edge sets connected to these seed nodes. This approach enables more effective allocation of the privacy budget, improving the overall privacy of social networks.

Tables 4 and 5 present the changes in edge entropy resulting from the application of two graph construction algorithms, namely, the UGTC algorithm and the UGIN algorithm. Both algorithms rely on the triad closure principle to add edges to the original graph while injecting probability values into the newly formed edges in order to preserve the privacy of the graph. Furthermore, the variables m and c represent the number of added edges and an adjusting factor, respectively. The total number of edges added to the graph is given by $m \times c$. The data presented in Tables 4 and 5 clearly de3monstrate that as the value of c increases, i.e., more edges are added, the edge entropy of the generated uncertain graph also increases.

In conclusion, the analysis of experimental data from Tables 2, , 4 and 5 reveals that as the number of nodes increases, the edge entropy of all four algorithms also increases. Furthermore, it is observed that EDP-SGA has a significantly higher edge entropy value than the other three algorithms.

5.3.2 Data utility analysis

Figure 7 shows the variation of NE for the EDP-SGA algorithm and the TCDP algorithm with different privacy budgets, as well as for the UGTC algorithm and the UGIN algorithm with adjustment factors c = 1 and C = 0.1. Figure 7 indicates that the NE of EDP-SGA and TCDP algorithms increases with the number of nodes, and also changes with the size of the privacy budget. The noise introduced by perturbation is larger, the range of edge probabilities in the perturbed graph is wider.

Figure 7a–f show that when the number of nodes is less than 500, the *NE* of TDCP changes the most and decreases with the increase in the privacy budget. When the privacy budget is greater than 0.1, the *NE* of UGIN has the greatest difference from the original image. Figure 7g, h show that when the number of nodes is around 547, the *NE* difference of UGTC is the highest. In Fig. 7i, j, the *NE* difference of UGIN when the number of nodes is 755. In Fig. 7k–p, when the

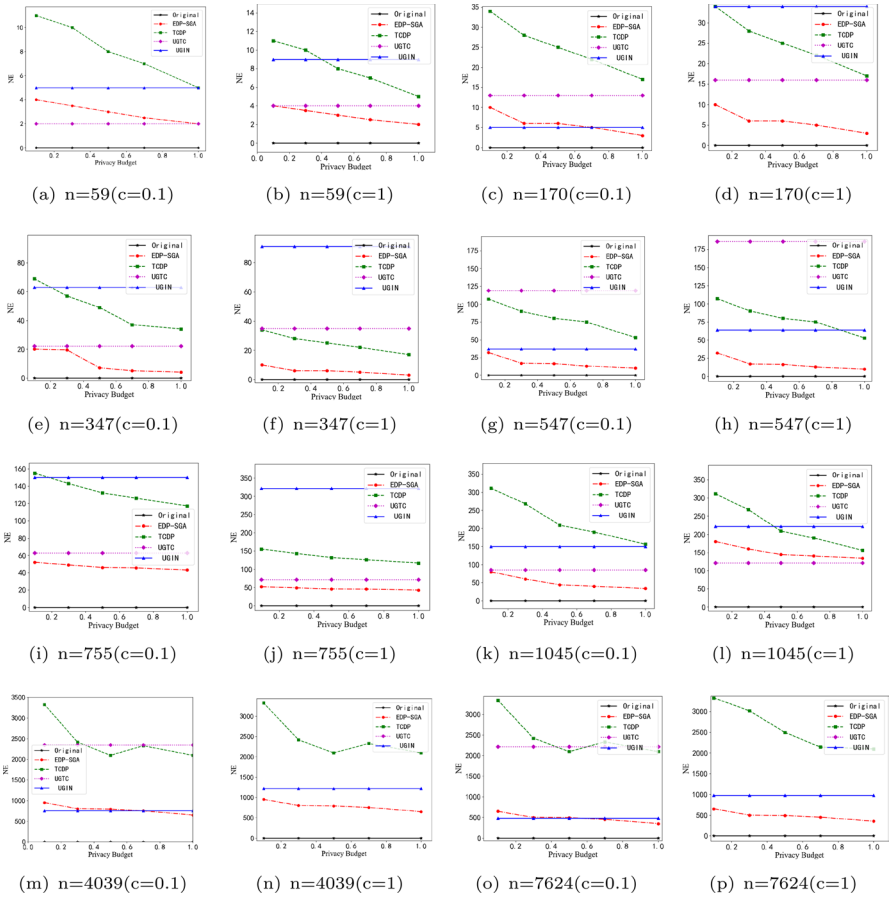


Fig. 7 *NE* comparison

number of nodes is greater than 1000, the maximum difference between the *NE* difference of TCDP and the original image exceeds 3000, followed by UGTC and UGIN, while the EDP-SGA proposed in this article has the smallest difference from the original image. However, when the number of nodes is around 1000, the maximum *NE* difference between the TCDP and the original graph reaches about 300, followed by differences of 84 for the UGTC and 83 for the UGIN, while the EDP-SGA proposed in this paper has a difference of only 80 from the original graph. In Fig. 7k, l, when the number of nodes more than 4000, the maximum difference in *NE* between the TCDP and the original graph is more than 3300, followed by differences of 3041 for the UGTC and 83 for the UGIN, while the EDP-SGA has a difference of only 80 from the original graph. Figure 7 show that the *NE* differences between the TCDP algorithm and the original graph are the largest, followed by the UNTC and UGIN algorithms, while the EDP-SGA has the smallest difference from the original graph.

Figure 7 demonstrates that the EDP-SGA algorithm has the smallest *NE* difference from the original graph, making it closer to the original data structure. The reason why the *NE* of the EDP-SGA is closest to that of the original graph is that after adding edges to the seed nodes during graph modification, the corresponding edges are deleted to prevent significant changes in the degree of the seed nodes. Whereas, the other three algorithms follow the theory of triadic closure to protect privacy in social network graphs, which can preserve the structure of social network graph data to some extent. However, during the protection process, random nodes are selected for edge additions, resulting in significant uncertainty in the level of privacy protection for the final uncertain graph. Moreover, as the number of nodes increases, the number of added edges also increases, leading to a significant decrease in the utility of the graph data. Thus, the EDP-SGA proposed in this paper has higher data utility.

Figure 8 displays the changes in *AD* (Average Degree) for EDP-SGA, TCDP, UGTC, and UGIN. These algorithms compute the overall change in node degree by assigning probability values to the edges disturbed in the graph. As depicted in

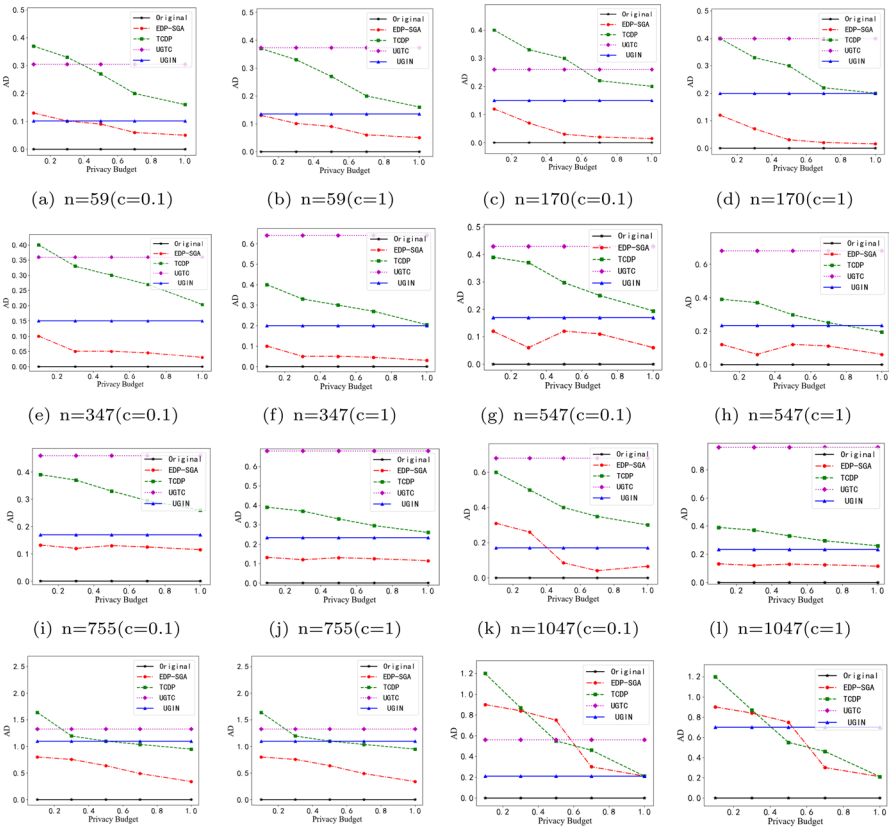


Fig. 8 *AD* comparison

Fig. 8, the *AD* values of all four algorithms increase with an increase in the number of nodes.

Figure 8a–d illustrate the variations in *AD* when the node number is below 200. It's apparent that TDCP and UGTC exhibit the most substantial disparities in *AD* compared to the original graph, with differences ranging from slightly above 0.2 to just under 0.4. In contrast, EDP-SGA and UGIN demonstrate the most minor deviations in *AD* from the original graph, staying within the 0.2 range. Figure 8e–l show the fluctuations in *AD* for node number spanning from 200 to 1047. It's noteworthy that UGTC showcases the most significant *AD* differences, with disparities ranging between 0.35 and 0.7, followed by TDCP and UGIN. In contrast, the proposed EDP-SGA maintains *AD* differences within the 0.2 threshold compared to the original graph. Figure 8m, n outline the changes in *AD* for node counts ranging from 200 to 1000. When the privacy budget is below 0.3, TCDP exhibits the largest *AD* disparities. However, with a privacy budget exceeding 0.3, UGTC records the most substantial *AD* differences, followed by UGIN. Meanwhile, TCDP and EDP-SGA display diminishing *AD* values as the privacy budget increases. Figure 8o–p demonstrate the variations in *AD* as the node count reaches 7624. With a privacy budget less than 0.4, TCDP showcases the most prominent *AD* disparities, approximately around 1.2. ED follows with discrepancies of approximately 0.9. When the privacy budget is 1, TCDP and EDG-SGA's *AD* differences gradually approach those of the original graph.

When the node number is less than 500, Fig. 9a–f present the difference between the *DV* of TCDP and the original graph reaches a maximum of 14, followed by 6.8 of UGTC, while the proposed EDF is closest to the original graph *DV*. In Fig. 9g–l, the difference between the *DV* of TCDP and the original graph is the largest when the node number ranges from 200 to 1000. When the node number is 547 and the privacy budget is greater than 0.3, the *DV* difference between UGTC and the original graph is 12.24, followed by 10.84 of TCDP. In Fig. 9m–p, when the node number is more than 4000, the maximum difference between the *DV* of TCDP and UGTC and the original graph is more than 70, followed by the difference range of UGIN in (10,40). In conclusion, Fig. 9 shows that the *DV* of the proposed EDP-SGA is closest to the *DV* of the original graph.

Figure 9 illustrates the changes in *DV* for the EDP-SGA, TCDP, UGTC, and UGIN algorithms as node degree increases. As the degree of nodes in the original graph is modified, the *DV* also increases accordingly.

The experimental comparison presented in Fig. 9 demonstrates that the proposed EDP-SGA approach has the smallest increase in *DV* value and the least deviation from the original graph's *DV*, compared to the other three algorithms. This result indicates that EDP-SGA maintains a better social network structure when dealing with uncertain graphs.

Degree distribution entropy (*DDE*) is used as a measure of uncertainty or diversity of node degree distribution in a network. Node degree is the number of connections between a node and other nodes, and the degree distribution indicates the frequency distribution of degrees of different nodes. Higher entropy of the degree distribution indicates a more diverse degree distribution of the network, with greater variation in the degrees of individual nodes. In order to measure the graph data

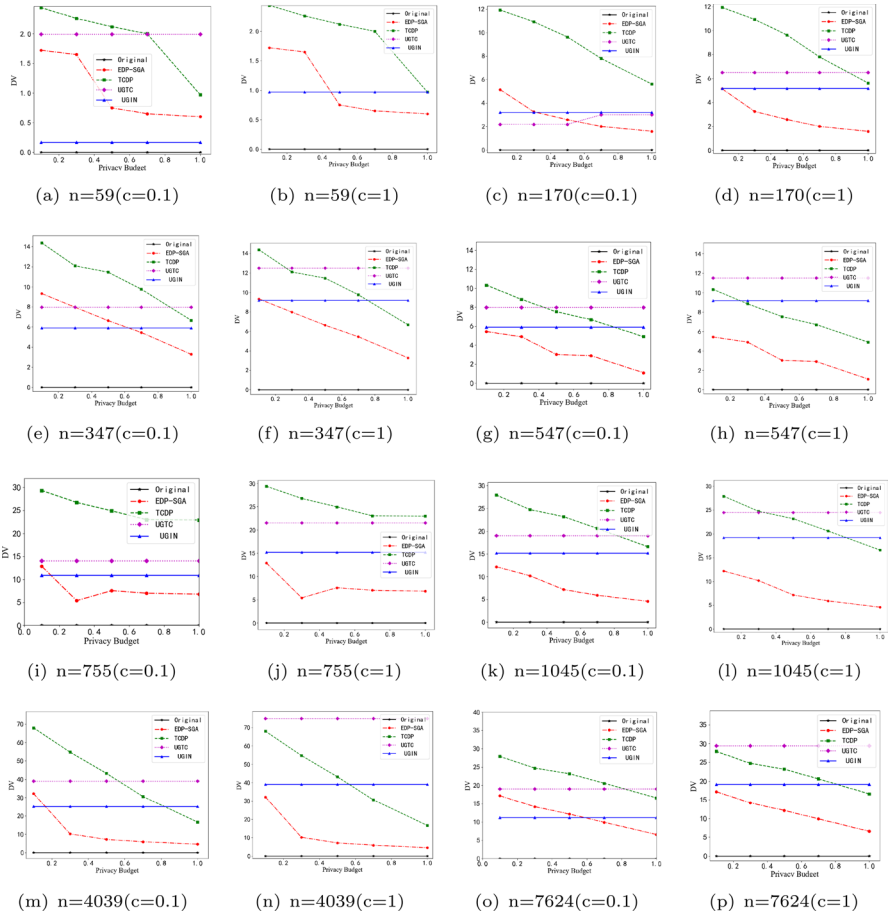
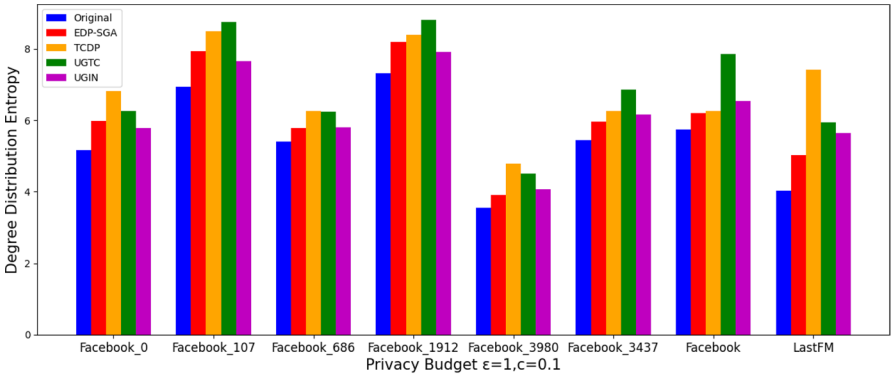


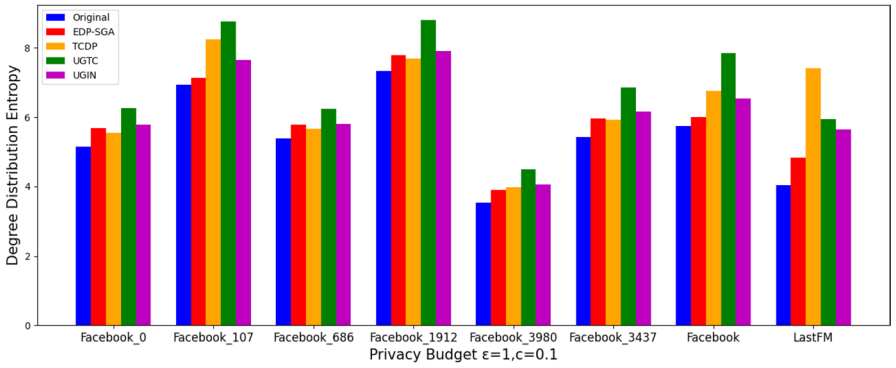
Fig. 9 *DV* comparison

utility more comprehensively, the degree distribution entropy is therefore introduced as a graph data utility indicator.

Figure 10 shows the variation of degree distribution entropy of the four algorithms on different datasets. Figure 10a shows the variation of *DDE* at privacy budget = 0.1 and adjustment factor c = 1. It can be seen that on datasets with more than 500 nodes less than 7000 (Facebook_107, Facebook_1912, Facebook_3437 and Facebook) UNTC has the largest difference in *DDE* from the original graph, followed by TCDP. On the LastFM dataset, TCDP has the largest difference with the original graph for *DDE* and the smallest for EDP-SGA. In Fig. 11b, with a privacy budget of 0.1 and an adjustment factor c = 1, UGTC demonstrates the largest deviation from the original graph across most datasets, followed by UGIN, TDCP, and EDP-SGA. This disparity can be attributed to the common usage of the triadic closure principle in TCDP, UGIN, and UGTC, which involves adding edges. Consequently, this addition of edges leads to increased node degrees within the graph.



(a) ε=0.1, c=1



(b) ε=1, c=1

Fig. 10 Degree distribution entropy comparison

However, UGIN selects seed nodes to add edges based on the betweenness centrality, so the edges added will be less compared to UGTC. Additionally, Fig. 10 illustrates that as the privacy budget increases, the *DDE* differences between EDP-SGA and TDCP and the original graph decrease. In summary, the degree distribution entropy can be used to measure the utility of the graph. This result indicates that EDP-SGA maintains a better social network structure when dealing with uncertain graphs.

A novel approach to address the issue of privacy protection in social network graph data is presented in this paper. We introduce the graphlet structure and information entropy to design an entropy-driven differential privacy protection scheme based on social graphlet attributes (EDP-SGA). EDP-SGA is capable of providing robust privacy protection through graph modification and differential privacy techniques, thereby ensuring the confidentiality of sensitive user information. In the graph modification stage, the EDP-SGA scheme proposed in this paper is to find the special graphlet structure composed of a specific set of seed nodes for edge modification, which is a modification of the local structure of the social network.

The comparison algorithms rely on the concept of ternary closure to connect nodes with edges and form triangles, resulting in alterations to the overall structure of the social network. Figures 7, 8, 9 and 10 show that the average number of edges, average degree variance and degree distribution entropy of EDP-SGA is the closest to the original graph structure. Therefore, the EDP-SGA scheme provides greater data availability. During the publishing phase of graph data, differential privacy is used to perturb the intimacy of edges to resist graph structure attacks. Then, we convert the perturbed intimacy into probabilities to avoid inferring the relationship or intimacy between users from their attribute intimacy. Finally, the uncertainty graph is publishing. According to Tables 2, 3 and 4, the uncertainty of uncertainty graph of EDP-SGA is higher than other algorithms.

5.3.3 Performance evaluation

Table 6 presents a comparison between the EDP-SGA scheme and three other algorithms. Both EDP-SGA and TCDP employ differential privacy to perturb the graph, resulting in an uncertain graph as output. However, TCDP is based on the triadic closure principle, adding edges to the social network graph structure, while EDP-SGA seeks specific graphlet structures of seed nodes for graph modification. In addition to using NE, AD, and DV to measure the data availability of the uncertain graph, EDP-SGA also employs *DDE* to assess the utility of the graph data.

Figure 11 illustrates a comparison of the running times of the four algorithms on datasets with varying numbers of nodes. Figure 11 shows that the running time of all four algorithms is fast basically around one second when the number of nodes is less than 100. When the number of nodes is less than 500, the difference in the running time of the four algorithms is not very large, and all are within five seconds. However, when the number of nodes is more than 500, it can be clearly seen that the running time of UGTC is much greater than the other algorithms, while UGIN has the least runtime. TCDP has the longest running time when the number of nodes is greater than 1000 and the number of nodes reaches about 4000, followed by EDP-SGA, and the least time is UGIN. However, when the number of nodes is greater than 1000, each algorithm's running time is more than 2000s. The number of nodes is more than 7000, the algorithm EDP-SGA proposed in this paper has the shortest running time and the longest running time is TCDP followed by UGTC and UGIN. This is due to the fact that TCDP, UNIN, and UNTC algorithms all rely on the triadic closure principle to add edges to the graph,

Table 6 Comparison of different privacy protections

	EDP-SGA	TCDP	UGTC	UGIN
Time Complexity	$O(n^2)$	$O(n \times m)$	$O(n^3 logn)$	$O(n^2 + m)$
Edge Edit	Edge Add & Delete	Edge Add	Edge Add	Edge Add
Privacy Metric	Edge Entropy	Edge Entropy	Edge Entropy	Edge Entropy
Utility Metric	NE & AD & DV & DDE	NE & AD & DV	NE & AD	NE & AD
Privacy protection technology	Differential privacy & Graph modification	Differential privacy & Graph modification	Graph modification	Graph modification

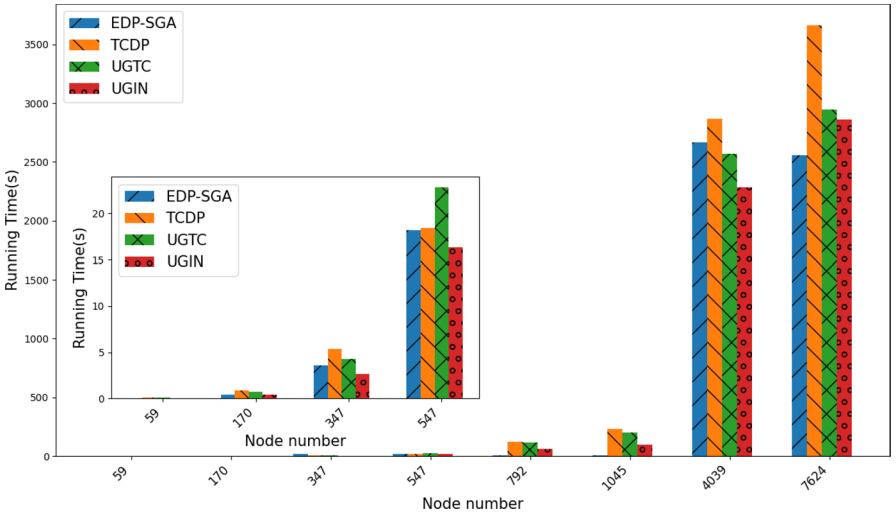


Fig. 11 Execution time comparison

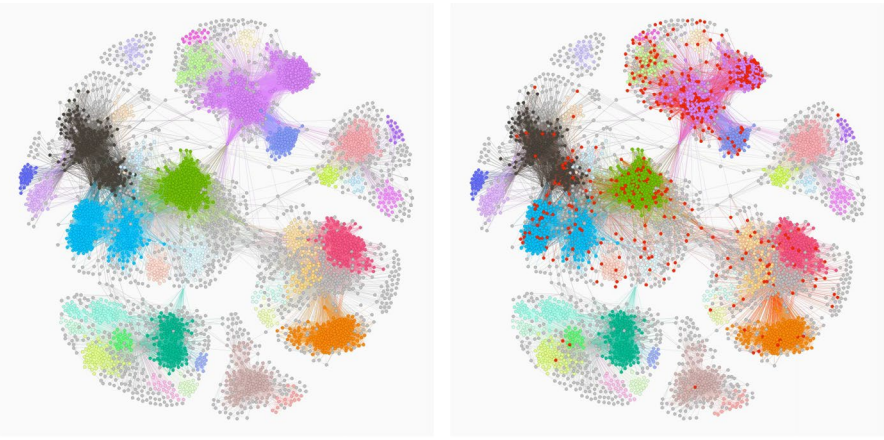
necessitating a traversal of the entire graph structure to identify suitable nodes for edge addition. In contrast, EDP-SGA identifies specific structures of seed nodes, eliminating the necessity for a complete graph structure traversal.

Figure 12 shows the visualization changes in social network graph data after protecting by the EDP-SGA scheme. Figure 12a, b present the visualizations of the Facebook dataset, which exhibit distinct community structure characteristics. In Fig. 12a, we observe the community structure after SCAN community partitioning, while Fig. 12b showcases the visualization of Facebook data after EDP-SGA processing. The red areas represent modified graphlet structures. It's worth noting that in densely connected community areas, the distribution of red points is denser, indicating a higher number of graphlets within these communities. Figure 12c, d display the visualization of change in the LastFM dataset's graph data. Compared to the Facebook dataset, the LastFM dataset has fewer edges, resulting in a lower graph structure and community density. In Fig. 12d, we can observe a relatively uniform distribution of red points with no significant dense areas.

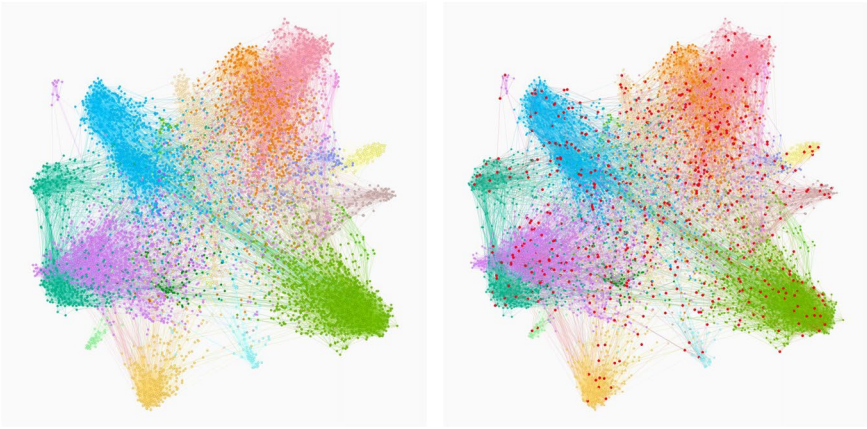
Overall, through the visualization comparisons in Fig. 12, it becomes evident that the EDP-SGA scheme can maintain the original data characteristics without substantial changes to the community structure. When combined with the results of the data utility experiments from Figs. 7, 8, 9, 10, 11 and 12, we can conclude that the EDP-SGA scheme ensures data privacy while preserving data efficiency.

6 Conclusion

As social network structures continue to evolve and become more complex, user nodes in real life often contain various social attributes and characteristic attributes that are interrelated. Furthermore, user attributes may be associated with multiple



(a) Graph data visualization of Facebook (b) Modified graph data visualization of Facebook



(c) Graph data visualization of LastFM (d) Modified graph data visualization of LastFM

Fig. 12 Visualization of graph data changes

sensitive pieces of information, making it challenging to protect their privacy when publishing social graphs. Different social groups also have varying degrees of sensitivity in their social relationships, and the degree of intimacy between nodes varies.

To address these challenges, this paper proposes an entropy-driven differential privacy protection scheme based on social graph graphlet attributes (EDP-SGA) and proposes three algorithms for privacy protection in social networks. Firstly, the attribute intimacy matrix construction algorithm is proposed to quantify the attribute intimacy between user nodes based on their feature attributes. Secondly, an influence algorithm based on user node attributes and information entropy is proposed to identify seed nodes containing significant amounts of information in the social network based on the attribute intimacy matrix and the concept of entropy. Finally, a graph data security publishing algorithm based on differential privacy is

proposed to modify the important graph graphlet of seed nodes using graph modification techniques to protect the topological structure of the social network. This is followed by adding noise to the edges between nodes using differential privacy protection technology and converting the network to an uncertain graph for publishing. To evaluate the EDP-SGA, a real Facebook network dataset is used, and the *NE*, *AD*, *DV*and *DDE* evaluation data structure indicators are utilized to compare the EDP-SGA with the TCDP algorithm, uncertain graph method based on the ternary closure algorithm, and an uncertain graph approach based on important nodes algorithm. Moreover, the performance of the EDP-SGA is evaluated by visualization and time performance comparison. The results indicate that the proposed EDP-SGA has high data privacy protection and data utility.

According to the analysis of the scenario, it can be found that not all computing centers are trusted. Possible future works are: (1) Because of the existence of untrustworthy centers. The distributed local differential privacy strategy will be discussed in graph data privacy-enhanced. (2) The social network graph data volume is extremely large, and the traditional algorithms can not be satisfied, will be extended to machine learning and deep learning direction.

Acknowledgements The authors would like to thank the National Natural Science Foundation of China (No. 61902069) and the Natural Science Foundation of Fujian Province of China (2021J011068).

Data availability Data will be made available on reasonable request.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Can U, Alatas B (2019) A new direction in social network analysis: online social network analysis problems and applications. Phys A Stat Mech Appl 535:122372
2. Li Y, Purcell M, Rakotoarivelo T, Smith D, Ranbaduge T, Ng KS (2023) Private graph data release: a survey. ACM Comput Surv 55(11):1–39
3. Kiranmayi M, Maheswari N (2021) A review on privacy preservation of social networks using graphs. J Appl Secur Res 16(2):190–223
4. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827
5. Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 601–610
6. Huang R, Chen Z, Zhai G, He J, Chu X (2022) A graph entropy measure from urelement to higher-order graphlets for network analysis. IEEE Trans Netw Sci Eng 10(2):631–644
7. Hong Y, Hu J, Zhao Y (2023) Would you go invisible on social media? An empirical study on the antecedents of users' lurking behavior. Technol Forecast Soc Change 187:122237
8. Gao Y, Li Y, Sun Y, Cai Z, Ma L, Pustišek M, Hu S (2022) IEEE access special section: privacy preservation for large-scale user data in social networks. IEEE Access 10:4374–4379
9. Cerruto F, Cirillo S, Desiato D, Gambardella SM, Polese G (2022) Social network data analysis to highlight privacy threats in sharing data. J Big Data 9(1):19
10. Rossi A, Arenas MP, Kocyigit E, Hani M (2022) Challenges of protecting confidentiality in social media data and their ethical import. In: 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS &PW), pp 554–561

11. Shejy G (2022) Data privacy and security in social networks. In: Principles of Social Networking: The New Horizon and Emerging Challenges, pp 387–411
12. Qian W, Shen Q, Wu P, Wu Z (2022) Research progress on privacy-preserving techniques in big data computing environment. Chin J Comput 45(4):669–701
13. Wei C, Ji S, Liu C, Chen W, Wang T (2020) AsgLDP: collecting and generating decentralized attributed graphs with local differential privacy. IEEE Trans Inf Forensics Secur 15:3239–3254
14. Weng L, Karsai M, Perra N, Menczer F, Flammini A (2018) Attention on weak ties in social and communication networks. In: Complex Spreading Phenomena in Social Systems, 213
15. Laitinen M, Fatemi M, Lundberg J (2020) Size matters: digital social networks and language change. Front Artif Intell 3:46
16. Rajkumar K, Saint-Jacques G, Bojinov I, Brynjolfsson E, Aral S (2022) A causal test of the strength of weak ties. Science 377(6612):1304–1310
17. Burke M, Kraut R (2013) Using Facebook after losing a job: differential benefits of strong and weak ties. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, pp 1419–1430
18. Doerfel ML, Moore PJ (2016) Digitizing strength of weak ties: understanding social network relationships through online discourse analysis. Ann Int Commun Assoc 40(1):127–148
19. Liu Y, Chen H, Liu Y, Zhao D, Li C (2022) State-of-the-art privacy attacks and defenses on graphs. Chin J Comput 4:702–734
20. Jha A, Dave M, Madan S (2017) Big data security and privacy: a review on issues, challenges and privacy preserving methods. Int J Comput Appl 975:8887
21. Ribeiro P, Paredes P, Silva ME, Aparicio D, Silva F (2021) A survey on subgraph counting: concepts, algorithms, and applications to network motifs and graphlets. ACM Comput Surv (CSUR) 54(2):1–36
22. Abawajy JH, Ninggal MIH, Herawan T (2016) Privacy preserving social network data publication. IEEE Commun Surv Tutorials 18(3):1974–1997
23. Antonakaki D, Fragopoulou P, Ioannidis S (2021) A survey of twitter research: data model, graph structure, sentiment analysis and attacks. Expert Syst Appl 164:114006
24. Ye W, Liu Z, Pan L (2021) Who are the celebrities? Identifying vital users on Sina Weibo microblogging network. Knowl Based Syst 231:107438
25. Mittal P, Papamanthou C, Song D (2012) Preserving link privacy in social network based systems. arXiv preprint arXiv:1208.6189
26. Ni C, Cang LS, Gope P, Min G (2022) Data anonymization evaluation for big data and IoT environment. Inf Sci 605:381–392
27. Xue M, Karras P, Chedy R, Kalnis P, Pung HK (2012) Delineating social network data anonymization via random edge perturbation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp 475–484
28. Huang K, Hu H, Zhou S, Guan J, Ye Q, Zhou X (2022) Privacy and efficiency guaranteed social subgraph matching. VLDB J 31:1–22
29. Mortazavi R, Erfani S (2020) GRAM: an efficient (k, l) graph anonymization method. Expert Syst Appl 153:113454
30. Tang C, Li P, Wang H, Wang C, Shen Z (2022) K-vretr privacy protection method for location-based services. J Chin Comput Syst 43(1):165–172
31. Ren W, Ghazinour K, Lian X (2022) kt-safety: graph release via k-anonymity and t-closeness. IEEE Trans Knowl Data Eng 35:9102
32. Dwork C, Naor M, Pitassi T, Rothblum GN (2010) Differential privacy under continual observation. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing, pp 715–724
33. Dwork C, Roth A et al (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9(3–4):211–407
34. Nguyen HH, Imine A, Rusinowitch M (2015) Differentially private publication of social graphs at linear cost. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pp 596–599
35. Li Y, Liu S, Li D, Wang J (2018) Release connection fingerprints in social networks using personalized diffierential privacy. Chin J Electron 27(5):1104–1110
36. Nguyen BP, Ngo H, Kim J, Kim J (2016) Publishing graph data with subgraph differential privacy. In: Information Security Applications: 16th International Workshop, WISA 2015, Jeju Island, Korea, August 20–22, 2015, Revised Selected Papers 16, pp 134–145

37. Adhikari MB, Suppakitpaisarn V, Paul A, Rangan CP (2020) Two-stage framework for accurate and differentially private network information publication. In: Computational Data and Social Networks: 9th International Conference, CSoNet 2020, Dallas, TX, USA, December 11–13, 2020, Proceedings 9, pp 267–279
38. Roohi L, Rubinstein BI, Teague V (2019) Differentially-private two-party egocentric betweenness centrality. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp 2233–2241
39. Ning B, Sun Y, Tao X, Li G (2021) Differential privacy protection on weighted graph in wireless networks. Ad Hoc Netw 110:102303
40. Qu L, Yang J, Wang Y (2023) Homogeneous network publishing privacy protection based on differential privacy uncertainty. Inf Sci 636:118925
41. Jian X, Wang Y, Chen L (2021) Publishing graphs under node differential privacy. IEEE Trans Knowl Data Eng 35:4164
42. Kassiano V, Gounaris A, Papadopoulos AN, Tsichlas K (2017) Mining uncertain graphs: an overview. In: Algorithmic Aspects of Cloud Computing: Second International Workshop, ALGOCLOUD 2016, Aarhus, Denmark, August 22, 2016, Revised Selected Papers, pp 87–116
43. Wu ZQ, Hu J, Tain YP, Shi WC, Yan J (2019) Privacy preserving algorithms of uncertain graphs in social networks. J Softw 30(4):1106–1120
44. Boldi P, Bonchi F, Gionis A, Tassa T (2012) Injecting uncertainty in graphs for identity obfuscation. arXiv preprint arXiv:1208.4145
45. Yan J, Zhang L, Shi W, Hu J, Wu Z (2017) Uncertain graph method based on triadic closure improving privacy preserving in social network. In: 2017 International Conference on Networking and Network Applications (NaNA), pp 190–195
46. Yan J, Zhang L, Tian Y, Wen G, Hu J (2018) An uncertain graph approach for preserving privacy in social networks based on important nodes. pp 107–111
47. Xu J, Zhang H, Xu L (2022) An uncertain graph privacy preserving scheme based on node similarity in social networks. In: 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), pp 108–114
48. Zheng J, Yang L (2022) Differential privacy algorithm of uncertain graph based on ternary closure. Jiangxi Metall 42(1):61–68
49. Hu J, Shi W, Yan J, Wu Z (2018) Research on privacy preserving method based on uncertain graph. Comput Technol Dev Comput Technol Dev 28(12):116–121
50. Hu J, Zhang J, Xu L, Lin L (2022) Research on influence of relationship between attribute and density affinity. J Chin Comput Syst 43(2):422–429
51. Li Z, Liu J, Wu K (2017) A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks. IEEE Trans Cybern 48(7):1963–1976
52. Tian Y, Yan J, Hu J, Wu Z (2018) A privacy preserving model in uncertain graph mining. In: 2018 International Conference on Networking and Network Applications (NaNA), pp 102–106
53. Rioul O (2021) This is it: a primer on Shannon's entropy and information. In: Information Theory: Poincaré Seminar 2018, pp 49–86
54. Li A, Pan Y (2016) Structural information and dynamical complexity of networks. IEEE Trans Inf Theory 62(6):3290–3339
55. Bhuiyan MA, Rahman M, Rahman M, Al Hasan M (2012) Guise: uniform sampling of graphlets for large graph analysis. In: 2012 IEEE 12th International Conference on Data Mining. IEEE, pp 91–100
56. Solé RV, Valverde S (2004) Information theory of complex networks: on evolution and architectural constraints. In: Ben-Naim E, Frauenfelder H, Toroczkai Z (eds) Complex Networks. Springer, Berlin, pp 189–207
57. Cai M, Cui Y, Stanley HE (2017) Analysis and evaluation of the entropy indices of a static network structure. Sci Rep 7(1):9340
58. Xu X, Yuruk N, Feng Z, Schweiger TA (2007) Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 824–833
59. Leskovec J, Mcauley J (2012) Learning to discover social circles in ego networks. In: Advances in Neural Information Processing Systems, vol 25

60. Rozemberczki B, Sarkar R (2020) Characteristic functions on graphs: birds of a feather, from statistical descriptors to parametric models. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM'20). ACM, pp 1325–1334