Edge Local Differential Privacy for Dynamic Graphs

Sudipta Paul[1(✉)], Julián Salas[2], and Vicenç Torra[1]

[1] Department of Computing Science, Umeå Universitet, Umea, Sweden
{spaul,vtorra}@cs.umu.se
[2] Internet Interdisciplinary Institute, Universitat Oberta de Catalunya, Barcelona, Spain
jsalaspi@uoc.edu

Abstract. Huge amounts of data are generated and shared in social networks and other network topologies. This raises privacy concerns when such data is not protected from leaking sensitive or personal information. Network topologies are commonly modeled through static graphs. Nevertheless, dynamic graphs better capture the temporal evolution and properties of such networks. Several differentially private mechanisms have been proposed for static graph data mining, but at the moment there are no such algorithms for dynamic data protection and mining. So, we propose two locally ϵ-differentially private methods for dynamic graph protection based on edge addition and deletion through the application of the noise-graph mechanism. We apply these methods to real-life datasets and show promising results preserving graph statistics for applications in community detection in time-varying networks.

The main contributions of this work are: extending the definition of local differential privacy for edges to the dynamic graph domain, and showing that the community structure of the protected graphs is well preserved for suitable privacy parameters.

1 Introduction and Related Work

A huge amount of data is generated every day in networked systems such as social networks [4,5], biological networks, internet peer-to-peer networks [13], and other technological networks [3]. These data can be modelled using graph theory in which, the nodes represent the users or objects and the edges represent the relationship between two nodes in such networks. All networks undergo changes, with nodes or edges arriving or going away as the system develops. Therefore, static graph networks are not adequate to model these kinds of network structures.

It is known that naif anonymization of a graph can lead to disclosure because the adversaries can use information that they posses to infer private information from the structure of the graph. Several types of such attacks have been developed. See e.g., de-anonymization attack [18], degree attacks [14], 1-neighborhood

B. Arief et al. (Eds.): SocialSec 2023, LNCS 14097, pp. 224–238, 2023.
https://doi.org/10.1007/978-981-99-5177-2_13

attacks [23], and sub-graph attacks [11]. Proper privacy models have been developed for static graphs. These models can be broadly classified into those following k-anonymity and those following differential privacy. We focus here in the differential privacy model [8].

The definition of "adjacent graphs" is the key to extending differential privacy to social networks [12]. So far different definitions have been provided: node privacy [10], edge privacy [19], out-link privacy, and partition privacy [19,20]. The most commonly used are node and edge differential privacy. Node privacy, provides desirable privacy protection but is impractical to deliver high utility (precise network analysis). Edge privacy shields users from attackers trying to learn about precise relationships between them, and it has been more widely adopted since it offers effective privacy protection in many practical applications.

Data on dynamic networks can take many different forms, but the most popular form and the one we consider in this paper is a collection of successively obtained, typically (but not necessarily) equally spaced snapshots of the network topology [22]. We restrict ourselves to networks based on a constant set of nodes. That is, nodes do not change but only edges do. This is not a limitation, because as we consider a finite set of snapshots all known a priori, the set of nodes appearing in at least one snapshot is known beforehand.

Keeping all of these in mind, here we propose dynamic graph privacy models and two novel edge-differentially private mechanisms for dynamic graphs. The closest related work, are the differentially private algorithms for counting-based problems in [9]. However, their algorithms are based on sensitivity and hence the edge randomization cannot be carried out locally, as in the present work.

1.1 Contributions and Paper Structure

In this work, our contributions are:

- the extension of the definition of local differential privacy for edges to dynamic graphs;
- the privacy mechanisms for providing graphs compliant with edge-local differential privacy for dynamic graphs. This is achieved by applying the noise-graph mechanism;
- an empirical analysis of such privacy mechanisms. We show that the community structure in dynamic graphs can be preserved while still protecting the edges with local differential privacy.

The remainder of the paper is arranged in the following manner. We conclude this section presenting basic definitions related to graph protection (Subsect. 1.2). In Sect. 2 we propose two differentially private algorithms for dynamic graph protection. In Sect. 3 we implement the algorithms and describe how they work on real datasets. Lastly, we draw a conclusion and give a sketch of the future work on the basis of all these discussions. This is in Sect. 4.

1.2 Basic Definitions

For graph randomization, we consider adding noise-graphs as in [16], that is, a simplification from the original definition in [21], assuming that the original graph and the noise graph have the same sets of nodes.

We denote by $G(V, E)$ the graph with the set of nodes V and set of edges E.

Definition 1. *Let $G_1(V, E_1)$ and $G_2(V, E_2)$ be two graphs with the same nodes V; then the addition of G_1 and G_2 is the graph $G = (V, E)$ where:*

$$E = (E_1 \setminus E_2) \cup (E_2 \setminus E_1).$$ Unió d'arestes

We denote G as

$$G = G_1 \oplus G_2.$$

We will add noise using the *Gilbert model*, which is denoted by $\mathcal{G}(n, p)$. That is, there are n nodes and each edge is chosen with probability p. The Gilbert and the Erdös-Rényi random graph models are the most common and general in the literature. It has been proved that they are asymptotically equivalent in [1]. So, to add noise to a graph G, we will draw a random graph G' from the Gilbert model (i.e., $G' \in \mathcal{G}(n, p)$) and add it to G, to obtain $\tilde{G} = G \oplus G'$.

Now, we can define the general noise-graph mechanism [15] that we will use.

Definition 2 (Noise-graph mechanism). *For any graph G with n nodes, and two probabilities p_0 and p_1, we define the following noise-graph mechanism:*

$$\mathcal{A}_{p_0,p_1}(G) = G \oplus G_0 \oplus G_1, \tag{1}$$

where G_0 and G_1 are such that:

$$G_0 = G' \setminus G \text{ for } G' \in \mathcal{G}(n, 1 - p_0)$$
$$G_1 = G'' \cap G \text{ for } G'' \in \mathcal{G}(n, 1 - p_1).$$

Definition 3 (Stochastic matrix associated to the noise graph). *The probabilities of randomization of an edge or a non-edge in a graph G after applying the noise-graph mechanism $\mathcal{A}_{p_0,p_1}$ are represented by the following stochastic matrix:*

$$P = P(\mathcal{A}_{p_0,p_1}) = \begin{pmatrix} p_0 & 1 - p_0 \\ 1 - p_1 & p_1 \end{pmatrix} \tag{2}$$

2 Dynamic Graphs

Considering that the relations in a dynamic network may remain or disappear over time, a basic model that accounts for the ratios of appearence or disappearance of edges in a graph over a period of time was proposed in [22].

Formally, the network is observed at an initial state G_0 at time $t = 0$, and for every snapshot G_t each node pair not connected by an edge at the previous

snapshot gains an edge with probability α, or not with probability $1-\alpha$. Similarly each existing edge disappears with probability β or not with probability $1-\beta$, from one snapshot to the next.

We denote the set of all dynamic graphs with n nodes V and T timestamps as $\mathcal{G} = \{(G_0, G_1 \dots G_T) : G_t = G_t(V, E_t) \text{ for } t = 0, \dots T\}$.

Thus, we can formally define a dynamic network graph model G iteratively by considering the initial state G_0 and applying the noise-graph mechanism $\mathcal{A}_{p_0,p_1}$ to G_{i-1} to obtain G_i, where $p_0 = 1-\alpha$ and $p_1 = 1-\beta$, for $i = 1, \dots T$.

Definition 4 (Dynamic-network-graph-model). *The dynamic network graph model consists of an initial state G_0 and states G_t, for $t = 1, \dots T$, defined by:*

$$G_t = \mathcal{A}_{1-\alpha,1-\beta}(G_{t-1})$$

We will denote it as: $G(G_0, T, \alpha, \beta)$.

In the other way around, if we have a series of snapshots $G_0, G_1, \dots, G_T$ of a graph that evolves with time, and we know that it follows the basic model of dynamic graphs, then, we can estimate α and β from these snapshots. The expressions to compute the parameters from the adjacency matrices are given in [22].

2.1 Differential Privacy for Dynamic Graphs

We adapt the definition of local differential privacy from [6] to be applied specifically to edges in a graph. Edge differential privacy is about the presence or absence of any edge, and local differential privacy is related to local randomization of each of the outputs. We combine both definitions for dynamic graphs to consider the edges in any of the graphs (snapshots) of the dynamic graph.

Definition 5 (Local differential privacy). *[6] A randomized algorithm π, satisfies ε-local differential privacy if for all inputs x, x' and all outputs $y \in Range(\pi)$:*

$$P(\pi(x) = y) \leq e^{\varepsilon} P(\pi(x') = y) \tag{3}$$

We denote by $\mathbb{1}_{uv(t)}$ the indicator function of edge uv in G_t, that is $\mathbb{1}_{uv(t)} = 1$ if $uv \in E_t$, and $\mathbb{1}_{uv(t)} = 0$ otherwise. Similarly, $\mathbb{1}_{\mathcal{A}(uv(t))}$ is the indicator function of edge uv in $\mathcal{A}(G_t)$, the randomized graph.

Definition 6 (Edge-local differential privacy for dynamic graphs). *An edge randomization algorithm $\mathcal{A} : \mathcal{G} \to \mathcal{G}$, satisfies ε-edge local differential privacy if for every pair of nodes $u, v \in V$, any timestamp $t \in \{1, \dots T\}$ and $x, x', y \in \{0, 1\}$:*

$$P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = x) \leq e^{\varepsilon} P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = x'), \tag{4}$$

we say that $\mathcal{A}$ is ε-edge locally differentially private (ε-eLDP).

Observe that when we consider local differential privacy at edge level, it implies the same probability of presence or absence of every edge in the protected graph independently of whether the edge was present or not in the original graph.

Thus, Definition 6 can be obtained from Definition 5, considering that the inputs x, x' represent whether any edge uv is present in the snapshot-graph G_t or not, and output y is the presence or absence of the same edge in the randomized graph.

2.2 Protection Mechanisms for Dynamic Graphs

Considering that a dynamic graph can be modelled with the dynamic-network-graph-model from Definition 4, it is natural to use it as a first approach to protect dynamic graphs. Thus, we show that the dynamic random graph model $G(G_0, T, p_0, p_1)$ is edge-differentially private for specific parameters p_0 and p_1.

Additionally, we define the parallel protection mechanism that adds noise to each snapshot of the dynamic graph, and therefore it may have a better utility.

Definition 7 (Dynamic-network-mechanism). *Let $G = (G_0, G_1 \dots G_T)$ be a dynamic graph. We define the protected dynamic graph $G' = (G'_0, G'_1 \dots G'_T)$ by letting:*

$$G'_0 = \mathcal{A}_{p_0,p_1}(G_0) \text{ and } G_i = \mathcal{A}^{i+1}_{p_0,p_1}(G_0).$$

That is, the dynamic-network-mechanism is:

$$\mathcal{D}_{p_0,p_1}(G) = G(G'_0, T, 1-p_0, 1-p_1),$$

Remark 1 (Randomization probabilities matrix). The probabilities of randomization for the dynamic network mechanism $\mathcal{D}_{p_0,p_1}(G)$ are calculated by the $(t = 1, \dots, T+1)$ powers of the stochastic matrix P in (2), we denote them as as:

$$P^t = \begin{pmatrix} p_0 & 1-p_0 \\ 1-p_1 & p_1 \end{pmatrix}^t = \begin{pmatrix} p_{00}[t] & p_{01}[t] \\ p_{10}[t] & p_{11}[t] \end{pmatrix} \tag{5}$$

Note that $p_{xy}[t]$ corresponds to: $P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = x)$, with $x, y \in \{0, 1\}$.

Theorem 1. *The mechanism $\mathcal{D}_{p_0,p_1}$ is ε-eLDP if*

$$e^{\varepsilon} \geq \max_{t=1,\dots T+1} \left\{ \frac{p_{10}[t]}{p_{00}[t]}, \frac{p_{11}[t]}{p_{01}[t]}, \frac{p_{00}[t]}{p_{10}[t]}, \frac{p_{01}[t]}{p_{11}[t]} \right\} \tag{6}$$

See proof on page 13.

Lemma 1. *Assume that the following inequality holds:*

$$e^{\varepsilon} \geq \max \left\{ \frac{p_{10}}{p_{00}}, \frac{p_{11}}{p_{01}}, \frac{p_{00}}{p_{10}}, \frac{p_{01}}{p_{11}} \right\} \tag{7}$$

Then (6) holds, that is:

$$e^{\varepsilon} \geq \max_{t=1,\dots T+1} \left\{ \frac{p_{10}[t]}{p_{00}[t]}, \frac{p_{11}[t]}{p_{01}[t]}, \frac{p_{00}[t]}{p_{10}[t]}, \frac{p_{01}[t]}{p_{11}[t]} \right\}$$

See proof on page 14.

Corollary 1. *The mechanism $\mathcal{D}_{p_0,p_1}$ is ε-eLDP if*

$$e^{\varepsilon} \geq \max\left\{\frac{p_{10}}{p_{00}}, \frac{p_{11}}{p_{01}}, \frac{p_{00}}{p_{10}}, \frac{p_{01}}{p_{11}}\right\} \tag{8}$$

Theorem 1 from [17] provides a complete characterization of the values for the probabilities (p_{00}, p_{11}) for which this equation holds, hence (p_{00}, p_{11}) can be parameterized depending on the ε required for protection.

Let us now consider an alternative protection mechanism. We call it the parallel protection of a dynamic graph. We define it as follows.

Definition 8 (Parallel protection mechanism). *Let $G = G_0, G_1, \ldots, G_T$ be a dynamic graph. Let $\mathcal{A}_{p_0,p_1}$ denote the noise-graph mechanism. Then, we define the parallel protection of the dynamic graph with parameters p_0 and p_1 as the protection process that provides $\tilde{G} = \tilde{G}_0, \tilde{G}_1, \ldots, \tilde{G}_T$ with $\tilde{G}_i = \mathcal{A}_{p_0,p_1}(G_i)$ for $i = 0, \ldots, T$.*

We denote the parallel protection of a dynamic graph G with parameters p_0 and p_1 as $\mathcal{A}^{||}_{p_0,p_1}(G)$.

Equivalently to Corollary 1, the following can be proven.

Proposition 1. *The parallel protection mechanism $\mathcal{A}^{||}_{p_0,p_1}$ satisfies ε-local differential privacy when*

$$e^{\varepsilon} \geq \max\left\{\frac{p_{10}}{p_{00}}, \frac{p_{11}}{p_{01}}, \frac{p_{00}}{p_{10}}, \frac{p_{01}}{p_{11}}\right\}$$

See proof on page 15.

3 Application to Community Detection Algorithms

This section consists of an experimental analysis of previous theoretical claims for an application of the proposed privacy algorithms. We base our utility analysis on *community detection algorithms* through *normalised mutual information*.

3.1 Experiment Description

We used two real-life datasets to evaluate the application of the proposed privacy protection algorithms. They are: CAIDA-AS relationship and DBLP datasets. We provide their basic statistics in Table 1.

One dataset is the *CAIDA-AS relationship dataset* [2] – Autonomous Systems (AS), which roughly corresponds to Internet Service Providers (ISP) and their relationships. We consider the p2p links, that are those that connect two ISPs who have agreed to exchange traffic on a quid pro quo basis. From this data we took the 1-month snapshot graphs for each of the 12 months in 2015.

The computer science bibliography *DBLP* provides its whole dataset of bibliographical entries in XML format, under the terms of the Open Data Commons Attribution License (ODC-BY 1.0). We use the coauthorship graph from [7], which has 1,482,029 unique authors and 10,615,809 timestamped co-authorship edges between authors. We preprocess the DBLP dataset, considering only the authors that published a paper each of the years between 2005 and 2013.

Table 1. Preprocessed datasets statistics

Dataset	No. of nodes	No. of Edges	Avg. Snapshot Density
CAIDA-AS	5,715	403,761	0.0010
DBLP	25,439	450,878	0.00007

The experiments are divided into five parts, that we summarize as follows:

1. We divide the data into snapshots such that the same vertices appear in every snapshot. In the case of DBLP this is the set of authors that have published at least a paper each year of the period from 2005 to 2013.
2. We fix the value of p_0, p_1 as in Table 2. We choose the smaller values of p_1 for smaller ε, otherwise the data will have a huge amount of edges. For larger ε values we may choose larger p_1 which also yields better utility. Note that the same ε can be obtained from several pair of values p_0, p_1, cf. [17].
3. We protect the data with our two proposed protection algorithms: the dynamic-network and the parallel mechanisms. We apply them five times each to obtain the average and confidence intervals of the utility measures.
4. We detect the community structure on each of the snapshot graphs, and compare it to the original community structure without privacy protection, we report the average and 95% confidence intervals on the figures.
5. We measure the density of each snapshot graph and compare them.

Table 2. Values of p_0 and p_1 to obtain the ε in the experiments.

ε	2	4	6	8	10	12	14	16	18	20
p_0	0.986602	0.998187	0.999755	0.999967	0.999955	0.999994	0.999999			
p_1	0.099				0.999					

Utility Measures: We use *Community Detection Algorithms*, to assess the partitioning or clustering of nodes as well as their propensity to stick together or disintegrate. Communities make it possible to map a network at a wide scale because they operate as meta-nodes in the network, that are used to facilitate analysis. The prediction of missing connections and the detection of fake links

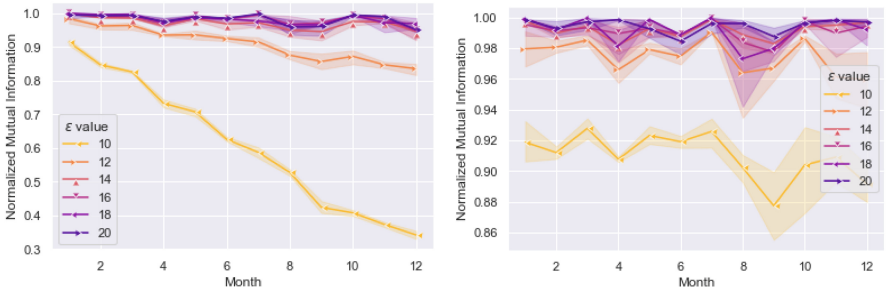
in the network are the two most significant applications of community detection in network research.

NMI, or *Normalised Mutual Information*, is a metric used to assess how well community discovery methods execute network partitioning. Due to its broad meaning and ability to compare two partitions even when there are different numbers of clusters, it is frequently taken into consideration.

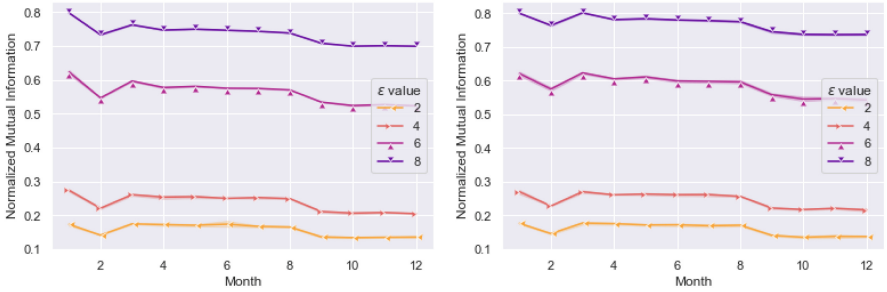
Finally, the *Graph Density* is defined to be the ratio of the number of edges with respect to the maximum possible edges.

3.2 CAIDA Dataset

In this section, we compare the effects of the dynamic-network and the parallel mechanisms for small and large ε values on the NMI and density measures of the CAIDA-AS dataset in Figs. 1 and 2.



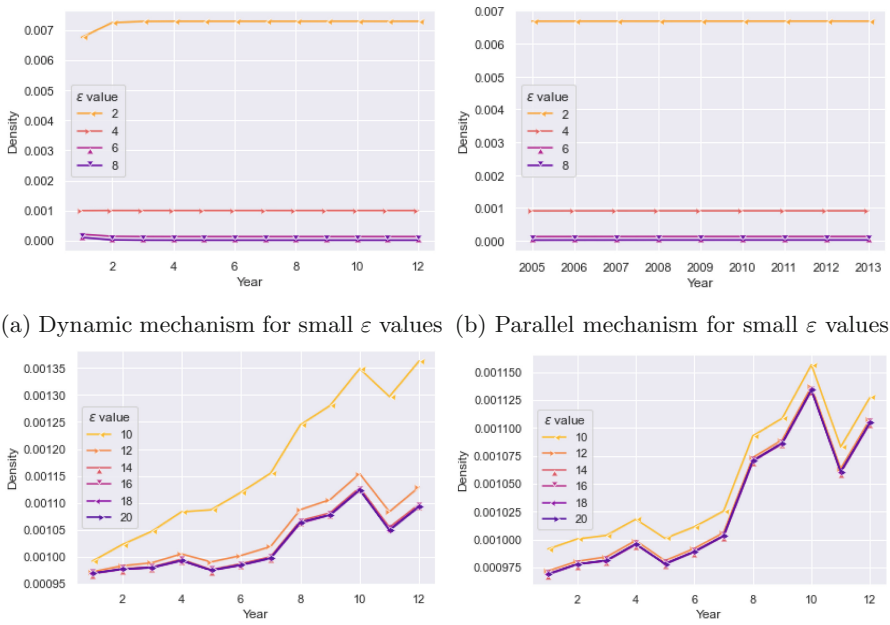
(a) Dynamic mechanism for large ε values (b) Parallel mechanism for large ε values



(c) Dynamic mechanism for small ε values (d) Parallel mechanism for small ε values

Fig. 1. Normalized mutual information between the communities detected on the CAIDA-AS data and the data protected with the dynamic and parallel mechanisms for several ε values.

In Fig. 1, we notice that for $\varepsilon = 20$ the NMI of both algorithms is almost 1, which means that the communities discovered over the protected data are almost the same as the original ones. Additionally, the NMI in the dynamic mechanism tends to decrease as the timestamp increases, whereas the parallel does not has this effect. For smaller values of ε, as the protection is stronger, the difference between the NMI values in both mechanisms is small. This may be explained with the larger densities obtained for small ε values in Fig. 2 (a) and (b). In contrast, in Fig. 2 (c) and (d), we note that for large ε values, the densities for each protected snapshot graphs are similar to the original density, and tend to it as the ε value increases.



(a) Dynamic mechanism for small ε values (b) Parallel mechanism for small ε values

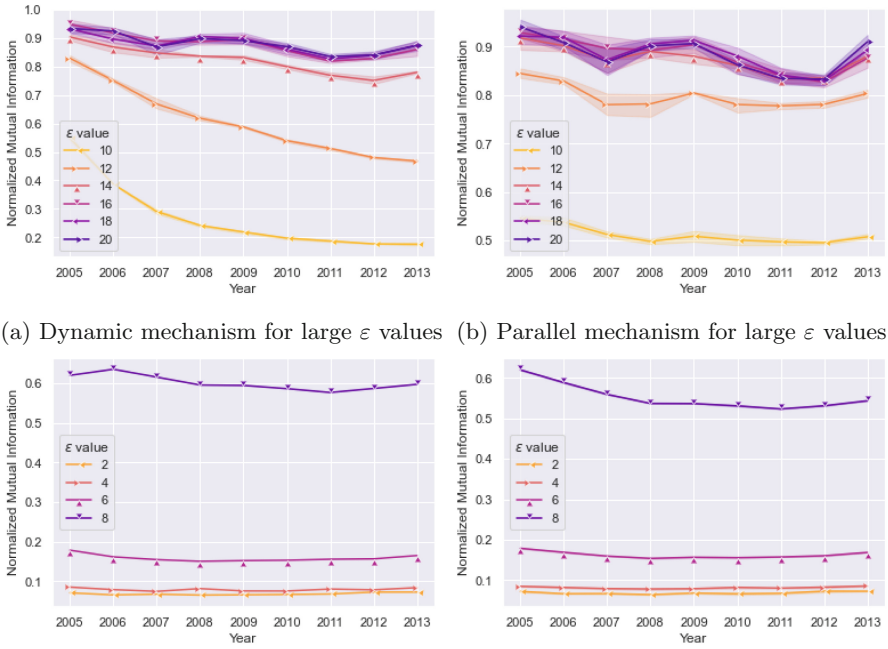
(c) Dynamic mechanism for large ε values (d) Parallel mechanism for large ε values

Fig. 2. Densities for the snapshot-graphs obtained by applying the dynamic and parallel mechanisms to CAIDA-AS.

3.3 DBLP Dataset

We compare the effects of the dynamic-network and the parallel mechanisms for small and large ε values on the NMI and density measures of the DBLP dataset in Figs. 3 and 4. In Fig. 3, it is shown that for larger ε, the communities detected on the protected graph are similar to the original communities, since the NMI is around 0.9. Also, it can be noted that the parallel mechanism has better NMI than the dynamic-network mechanism for large ε values. Additionally,

the dynamic-network mechanism's NMI decreases in time, whereas the parallel does not. For smaller ε values the performance of both mechanisms is similar.

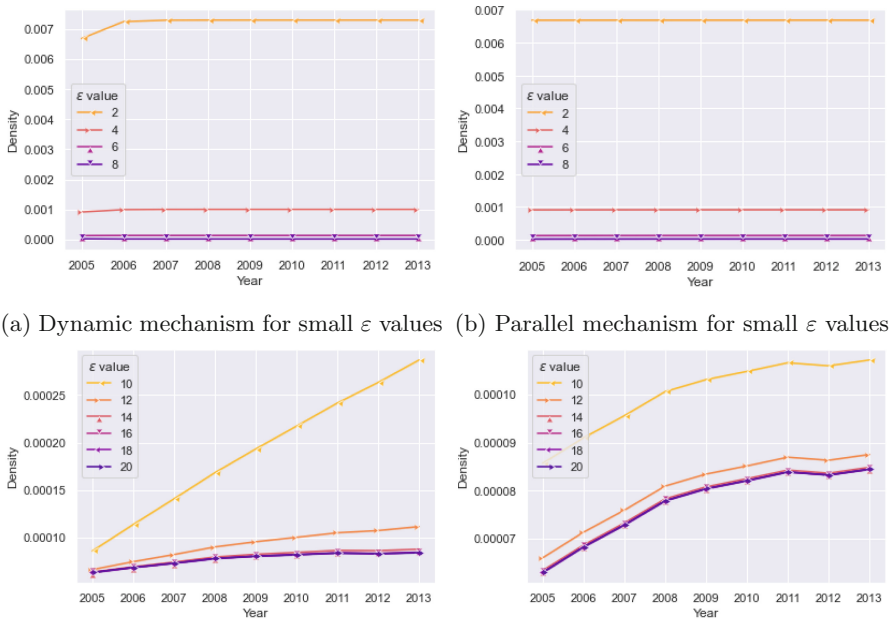


(a) Dynamic mechanism for large ε values (b) Parallel mechanism for large ε values

(c) Dynamic mechanism for small ε values (d) Parallel mechanism for small ε values

Fig. 3. Normalized mutual information between the communities detected on the DBLP data and the data protected with the dynamic and parallel mechanisms for several ε values.

In Fig. 4 (a) and (b), we show the effect of both mechanisms on the densities of the graph for small ε values. We notice that for $\varepsilon = 2$ the density of the protected snapshots is near to 0.007 which means that they have around 4,529,999 of edges, which is 100 times the original average snapshot density of 0.00007. In Fig. 4 (c) and (d), we show the effect of both mechanisms on the densities of the graph for large ε values. An increase in the density means that there have been created more noise-edges than there have been erased real-edges. Again, the parallel mechanism incurs a lower increase in density than the dynamic. Moreover, the increase in density for $\varepsilon = 10$ is more steep for the dynamic than for the rest of ε values.



(a) Dynamic mechanism for small ε values (b) Parallel mechanism for small ε values

(c) Dynamic mechanism for large ε values (d) Parallel mechanism for large ε values

Fig. 4. Densities for the snapshot-graphs obtained by applying the dynamic and parallel mechanisms to DBLP.

4 Conclusions and Future Scope

We proposed two protection methods for dynamic graphs: the dynamic-network and the parallel protection mechanisms. We extended the definition of local differential privacy for edges in dynamic graphs. We showed that both our proposed methods are ε-edge locally differentially private for specific values of randomization probabilities in the noise-graph mechanism p_0 and p_1. We performed an empirical analysis of such algorithms, to show that they keep the community structure of the dynamic graphs while protecting their edges with local differential privacy.

In this work, we only focus on edge privacy with fixed nodes which extend the ε-edge local differential privacy notion. But, there is still room to look over changing nodes and change of edges and nodes simultaneously. We also would like to extend this notion of privacy in the path of graph neural networks and federated learning. We plan to extend the empirical analysis to other graph utility metrics and other definitions of dynamic graphs.

Acknowledgements. This research was partly supported by the Spanish Ministry of Science and Innovation under project PID2021-125962OB-C31 "SECURING" and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

A Proofs

Proof (Proof of Theorem 1). Let $G = G_0, G_1, \ldots G_T$ be a dynamic graph. Recall that $\mathcal{D}_{p_0,p_1}(G) = G(g_0, T, 1-p_0, 1-p_1)$, which outputs the initial state $g_0 = \mathcal{A}_{p_0,p_1}(G_0)$, and the further snapshots $g_1, \ldots, g_T$ such that $g_i = \mathcal{A}^{i+1}_{p_0,p_1}(G_0)$.

To prove that mechanism $\mathcal{D}_{p_0,p_1}$ is ε-eLDP we must show that:

$$\frac{P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = x)}{P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = x')} \leq e^{\varepsilon}$$

We assume that $x \neq x'$, otherwise the inequality holds.

Now, suppose that $x = 1$ and $x' = 0$, and that (6) holds.

Therefore, we must prove that, for $y = 0, 1$ and $t \geq 1$:

$$\frac{P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = 1)}{P(\mathbb{1}_{\mathcal{A}(uv(t))} = y \mid \mathbb{1}_{uv(t)} = 0)} \leq e^{\varepsilon} \tag{9}$$

Note that, these probabilities can be calculated using the stochastic matrix P^t in (5), and by Remark 1 they are the following for $y = 0$:

$$P(\mathbb{1}_{\mathcal{A}(uv(t))} = 0 \mid \mathbb{1}_{uv(t)} = 1) = p_{10}[t]$$
$$P(\mathbb{1}_{\mathcal{A}(uv(t))} = 0 \mid \mathbb{1}_{uv(t)} = 0) = p_{00}[t]$$

and the following for $y = 1$:

$$P(\mathbb{1}_{\mathcal{A}(uv(t))} = 1 \mid \mathbb{1}_{uv(t)} = 1) = p_{11}[t]$$
$$P(\mathbb{1}_{\mathcal{A}(uv(t))} = 1 \mid \mathbb{1}_{uv(t)} = 0) = p_{01}[t]$$

Thus, for $y = 0, 1$, the Eq. (9) becomes:

$$\frac{p_{10}[t]}{p_{00}[t]} \leq e^{\varepsilon} \text{ and } \frac{p_{11}[t]}{p_{01}[t]} \leq e^{\varepsilon}$$

The argument is similar when $x = 0$ and $x' = 1$. As all these probabilities are bounded by e^{ε} by (6), we finish the proof.

Proof (Proof of Lemma 1). Assume that (7) holds. We first show that (6) is true for $t = 2$. Note that:

$$\frac{p_{00}[2]}{p_{10}[2]} = \frac{p_{00}p_{00} + p_{01}p_{10}}{p_{10}p_{00} + p_{11}p_{10}}$$

Divide all by p_{10} and, by (7), to obtain:

$$\frac{(\frac{p_{00}}{p_{10}})p_{00} + p_{01}}{p_{00} + p_{11}} \leq \frac{e^{\varepsilon}p_{00} + p_{01}}{p_{00} + p_{11}}.$$

And,

$$\frac{e^{\varepsilon}p_{00} + p_{01}}{p_{00} + p_{11}} \leq e^{\varepsilon} \iff \frac{p_{01}}{p_{11}} \leq e^{\varepsilon}.$$

Which is true from (7).

Note that:

$$\frac{p_{10}[2]}{p_{00}[2]} = \frac{p_{10}p_{00} + p_{11}p_{10}}{p_{00}p_{00} + p_{01}p_{10}}$$

Again, divide all by p_{10} and, by (7), obtain:

$$\frac{p_{00} + p_{11}}{(\frac{p_{00}}{p_{10}})p_{00} + p_{01}} \leq \frac{p_{00} + p_{11}}{(\frac{1}{e^{\varepsilon}})p_{00} + p_{01}}.$$

Moreover,

$$\frac{p_{00} + p_{11}}{(\frac{1}{e^{\varepsilon}})p_{00} + p_{01}} \leq e^{\varepsilon} \iff \frac{(\frac{1}{e^{\varepsilon}})p_{00} + p_{01}}{p_{00} + p_{11}} \geq \frac{1}{e^{\varepsilon}} \iff \frac{p_{01}}{p_{11}} \geq \frac{1}{e^{\varepsilon}} \iff \frac{p_{11}}{p_{01}} \leq e^{\varepsilon},$$

which is true from (7). The proof is similar for $\frac{p_{11}[2]}{p_{01}[2]}$ and $\frac{p_{01}[2]}{p_{11}[2]}$. Finally, since (6) is true for $t = 2$, considering that it is true for $t = 1$, the proof for all t follows by iteratively letting the corresponding $p_{ij}[2] = p_{ij}$, and all the rest is the same.

Proof (Proof of Proposition 1). We need to consider two cases. In the first case, the edge uv is in $\mathcal{A}^{||}_{p_0,p_1}(G)$ and also in $\mathcal{A}^{||}_{p_0,p_1}(G')$. We consider that we have a graph G with an edge uv and the graph G' does not have this edge. Then, the protection mechanism will produce graphs $\tilde{G}_1$ and $\tilde{G}'$. With probability p_1 we have that the edge uv is still in $\tilde{G}_1$ and with probability $1 - p_0$ the edge uv has appeared in $\tilde{G}'$. In order that the condition for differential privacy holds we need

$$p_1/(1 - p_0) \leq e^{\varepsilon}.$$

Similarly, if the edge uv is in G' but not in G, we will have

$$(1 - p_0)/(p_1) \leq e^{\varepsilon}.$$

The second case is when we have that the edge uv is neither in $\mathcal{A}^{||}_{p_0,p_1}(G)$ not in $\mathcal{A}^{||}_{p_0,p_1}(G')$. Let us consider that the graph G does not have the edge uv but the graph G' has this edge. Then, the protection mechanism will add the edge uv to G with probability $1 - p_0$, and the edge uv will be kept in G' with probability p_1. So, we need that

$$(1 - p_0)/p_1 \leq e^{\varepsilon}.$$

Similarly, if the edge uv is in G but not in G', then

$$p_0/(1 - p_1) \leq e^{\varepsilon}.$$

References

1. Aiello, W., Chung, F., Lu, L.: A random graph model for power law graphs. Exp. Math. **10**(1), 53–66 (2001)

2. As rank. https://catalog.caida.org/dataset/as_rank. Accessed 25 Jan 2023
3. Asharov, G., et al.: Privacy-preserving interdomain routing at internet scale. Cryptology ePrint Archive (2017)
4. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 44–54 (2006)
5. Bergami, G., Bertini, F., Montesi, D.: On approximate nesting of multiple social network graphs: a preliminary study. In: Proceedings of the 23rd International Database Applications & Engineering Symposium, pp. 1–5 (2019)
6. Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., Wang, T.: Privacy at scale: local differential privacy in practice. In: Proceedings of the 2018 International Conference on Management of Data, SIGMOD 2018, pp. 1655–1658. Association for Computing Machinery, New York (2018)
7. Demaine, E., HajiaGhayi, M.T.: BigDND: big dynamic network data. https://projects.csail.mit.edu/dnd/DBLP/
8. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
9. Fichtenberger, H., Henzinger, M., Ost, W.: Differentially private algorithms for graphs under continual observation. In: 29th Annual European Symposium on Algorithms (ESA 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
10. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 169–178. IEEE (2009)
11. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. Proc. VLDB Endow. **1**(1), 102–114 (2008)
12. Jiang, H., Pei, J., Yu, D., Yu, J., Gong, B., Cheng, X.: Applications of differential privacy in social network analysis: a survey. IEEE Trans. Knowl. Data Eng. **35**(1), 108–127 (2021)
13. Lakshmanan, L.V., Ng, R.T., Ramesh, G.: To do or not to do: the dilemma of disclosing anonymized data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 61–72 (2005)
14. Pedarsani, P., Grossglauser, M.: On the privacy of anonymized networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1243 (2011)
15. Salas, J., González-Zelaya, V., Torra, V., Megías, D.: Differentially private graph publishing through noise-graph addition. In: Modeling Decisions for Artificial Intelligence: 20th International Conference, MDAI 2023, Umeå, Sweden, 19–22 June 2023, Proceedings, pp. 253–264 (2023)
16. Salas, J., Torra, V.: Differentially private graph publishing and randomized response for collaborative filtering. In: Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, 8–10 July 2020, pp. 415–422. ScitePress (2020)
17. Salas, J., Torra, V., Megías, D.: Towards measuring fairness for local differential privacy. In: Garcia-Alfaro, J., Navarro-Arribas, G., Dragoni, N. (eds.) DPM CBT 2022. LNCS, vol. 13619, pp. 19–34. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25734-6_2

18. Takbiri, N., Shao, X., Gao, L., Pishro-Nik, H.: Improving privacy in graphs through node addition. In: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 487–494. IEEE (2019)
19. Task, C., Clifton, C.: A guide to differential privacy theory in social network analysis. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 411–417. IEEE (2012)
20. Task, C., Clifton, C.: What should we protect? Defining differential privacy for social network analysis. In: Can, F., Özyer, T., Polat, F. (eds.) State of the Art Applications of Social Network Analysis. LNSN, pp. 139–161. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05912-9_7
21. Torra, V., Salas, J.: Graph perturbation as noise graph addition: a new perspective for graph anonymization. In: Pérez-Solà, C., Navarro-Arribas, G., Biryukov, A., Garcia-Alfaro, J. (eds.) DPM/CBT -2019. LNCS, vol. 11737, pp. 121–137. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31500-9_8
22. Zhang, X., Moore, C., Newman, M.E.: Random graph models for dynamic networks. Eur. Phys. J. B **90**(10), 1–14 (2017)
23. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 506–515. IEEE (2008)