A Privacy-Preserving Subgraph-Level Federated Graph Neural Network via Differential Privacy

Yeqing Qiu[1,2], Chenyu Huang[1], Jianzong Wang[1](✉), Zhangcheng Huang[1], and Jing Xiao[1]

[1] Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China
jzwang@188.com, xiaojing661@pingan.com.cn
[2] Beijing Jiaotong University, Beijing, China
yeqing@bjtu.edu.cn

Abstract. Currently, the federated graph neural network (GNN) has attracted a lot of attention due to its wide applications in reality without violating the privacy regulations. Among all the privacy-preserving technologies, the differential privacy (DP) is the most promising one due to its effectiveness and light computational overhead. However, the DP-based federated GNN has not been well investigated, especially in the sub-graph-level setting, such as the scenario of recommendation system. The biggest challenge is how to guarantee the privacy and solve the non independent and identically distributed (non-IID) data in federated GNN simultaneously. In this paper, we propose DP-FedRec, a DP-based federated GNN to fill the gap. Private Set Intersection (PSI) is leveraged to extend the local graph for each client, and thus solve the non-IID problem. Most importantly, DP is applied not only on the weights but also on the edges of the intersection graph from PSI to fully protect the privacy of clients. The evaluation demonstrates DP-FedRec achieves better performance with the graph extension and DP only introduces little computations overhead.

Keywords: Recommendation system · Federated learning · Subgraph-level federated learning · Graph neural network · Differential privacy

1 Introduction

Graph neural network (GNN) has been applied to multiple scenarios such as molecule prediction [5,18], social network analysis [2,17], recommendation systems [8] and knowledge graph [20]. However, GNN approaches mainly rely on the centralized data, which is different from the real-world scenario where the source data may be stored at different organizations. For example, e-commerce platforms that sell different types of items have separate purchase and rating records of their users and items. In order to explore potential new users and provide better recommendation services to existing users, E-commerce platforms

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 165–177, 2022.
https://doi.org/10.1007/978-3-031-10989-8_14

would build a better model jointly learned from multiple data resources. In the meantime, the user privacy should be protected for ethical concerns and compliance with government regulations.

As a result, approaches are presented to combine the well-known privacy-preserving framework, federated learning (FL), and GNN. Different technologies such as differential privacy (DP) [16,22,23,26], homomorphic encryption [22], secret sharing [26] are widely applied to dealing with risks of privacy leakage. Among the techniques mentioned above, DP is the most promising one due to its light computational overhead and high fidelity. DP perturbs the data with a small noise without lowering the accuracy of the entire model, *i.e.*, if the input signal changes, the distribution of the output only changes a little.

Currently, real-world scenarios of privacy-preserve graph learning mainly concentrates on three settings [7]: graph-level setting [26], sub-graph-level setting [14,22,23,25] and node-level setting [3]. Among these settings, sub-graph-level is the most attractive since it is a good fit to the most important/common application scenario such as recommendation system and knowledge graph. For example, in recommendation systems, every data holder will only own the part of graph that contains the relationship between user and item. The biggest challenge in this setting is preserving the privacy and solving the Non-IID problem in federated GNN simultaneously. However, these work either assume one party owns the global topology [7,26], which violate the basic assumption in general scenario where no one is allowed to own the whole typology, or do not consider the information from the neighbors [14,22], which do not solve the Non-IID problem and thus lead to low accuracy of the model. Therefore, these approaches cannot be directly applied in the general sub-graph level scenario.

In this paper, we propose a novel DP-based GNN that aims at the sub-graph level setting in Sect. 3. To solve the Non-IID problem, the FedRec that utilizes the K-hop extension to expand the sub-graph of each client is introduced. The privacy of the communication between clients is preserved via the Private Set Intersection (PSI). Furthermore, we propose DP-FedRec that leverages DP in FedRec. The core idea is to apply well-designed noises to both adjacency edges and weights of client's sub-graph. Specifically, the Laplacian noise is applied on the edges via Lapgraph algorithm and apply the Laplacian noise on the weights. The analysis and evaluation in Sect. 4 and 5 show the K-hop extension achieves better performance than previous schemes and the DP introduces limited computational overhead. We summarize the main contributions as follows:

- We propose a state-of-art learning paradigm on sub-graph setting based on DP, which is able to be applied to many link prediction tasks.
- We utilize K-hop extension for exchanged feature and adjacency information and preserve the privacy of both the feature and edge information via DP.
- We evaluate our algorithms on two recommendation datasets, and demonstrate the effectiveness of our approach.

2 Preliminaries

2.1 Problem Formulation

In this work, denote $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$ and $\mathcal{P} = \{p_1, p_2, \cdots, p_m\}$ as user and item respectively. The purchasing interaction of user and item relationship is represented by a bipartite graph $\mathcal{G} \in \mathbb{R}^{n \times m}$, in which the value of edges refers to the points the user rate the item. Since each client will only have a part of global graph, for client i, the user-item bipartite graph is denoted as $G^i = (V^i, E^i)$. In detail, the set of vertexes and edges are denoted as V^i, E^i respectively. The task is to predict the ratings of users and items based on user-item graph. Thus, client i will train a local model in round r, the parameters of which are denoted as θ_i^l. The global model parameter that aggregate from each client is θ^r. Additionally, define $\mathsf{dist}(x, y, \mathcal{G})$ as the shortest path of vertex x and y in graph $\mathcal{G}$. Define $\mathsf{dist}(v, \mathcal{S}, \mathcal{G}) = \min_{x \in \mathcal{S}} \mathsf{dist}(v, x, \mathcal{G})$. The notation is summarized in Table 1.

Table 1. Notations used in DP-FedRec.

l	Number of clients
n, m	Number of users and items in graph $\mathcal{G}$
u_i, p_i	User i, item i
G^i	User-item graph of client i
V^i, E^i	Vertex set and edge set in G^i
$\bar{G}^i$	Extended user-item graph of client i
$\bar{V}^i, \bar{E}^i$	Extended vertex set and edge set in $\bar{G}^i$
K	Parameter of K-hop extension
r	Communication round
θ^r	Parameters of global model in round r
θ_i^r	Parameters of client i's local model in round r
$\nabla\theta_i$	Gradient of parameters of local model

2.2 Local Differential Privacy

Local differential privacy guarantees the privacy of the user in the process of collecting information. Specifically, before the user uploads the data to an untrusted third party, a certain amount of noises is added to the uploaded data. This guarantees that the data collectors can hardly infer the specific information of any user, but are able to learn the statistical properties of the data by increasing the amount of data.

Different from the previous unweighted graph [23], the user-item graph in recommendation system is a weighted graph. Therefore, in order to protect information of user-item graph, the definition of DP in undirected weighted graph data is obtained by combining both unweighted undirected graph information and weight information. Consistent with prior work [23], we adapt the idea of edge differential privacy based on adjacency matrix.

Definition 1 (Neighbor Relation). *Two matrices are called neighbors if there is only one different node. Specifically, the graph corresponding to the two matrices can be obtained by adding or deleting an edge or modifying value of an edge.*

Definition 2 (ϵ-Weighted Edge Local Differential Privacy). *A mechanism M is called to satisfy ϵ-Weighted Edge Local Differential Privacy if for all neighbor matrix pairs X and X', and for any possible output $t \in Range(M)$:*

$$P[M(X) = t] \leq e^{\epsilon} P[M(X') = t] \tag{1}$$

2.3 Federated Graph Neural Network

Graph neural network is widely used in recent recommendation systems [24]. In this paper, we leverage the graph convectional network (GCN) [9] under the message passing neural network framework (MPNN). MPNN is a supervised learning framework which extracts information from the user-item graph by aggregating adjacency information into the latent space, and then generates the prediction from the latent space.

Furthermore, we extend GNN to the federated scenario which is the same as in [7]. Specifically, it is a sub-graph setting where each entity/company has a part of data/graph, such as users and rating information, and a model is jointly trained on the entire data for better prediction accuracy. Therefore, there are multiple clients and one centralized server. For communication round r in the training stage, client i will get the model parameter θ_i^r by training the local model for e epochs on the sub-graph G_r^i. The server will aggregates the parameters $\theta_r = \frac{\sum \theta_i^r}{l}$ and distribute them to all local clients, and each client updates its local model parameters as θ_i^r.

2.4 Private Set Intersection

Private Set Intersection (PSI) is a cryptographic protocol in multiparty computation. It allows two clients to get the intersection set of the data without revealing any information outside the intersected data. There are many different implementations of PSI, DP-FedRec instantiates the PSI the same as [10]. It leverages the programmable pseudo random function (OPPRF) which is fast and efficient.

3 Approach

3.1 Overview

The basic federated GNN framework does not use the graph information from others and could cause non-IID problem in the training data. We will first present FedRec which extends the sub-graph of each client without leaking the information of the edges. Then we will introduce DP-FedRec that combines FedRec and DP and jointly considers the privacy of both weights and connectivity of the edges simultaneously.

Specifically, DP-FedRec jointly trains a model via four steps as shown in Fig. 1: (i) All the clients add noises on the graph data including both weights and edges; (ii) The clients extend the local graph via K-hop extension; (iii) Each client trains the local model on the extended graph and submits the parameters to the server; (iv) The centralized server aggregates the parameters and distributes the updated parameters to all the clients. The process will continue until certain number of rounds is reached.

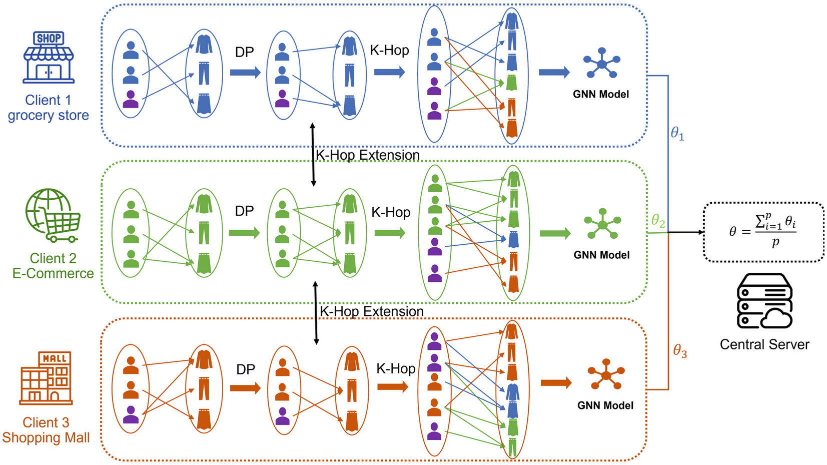

Fig. 1. Overall framework of DP-FedRec. Each bipartite graph refers to purchasing relationship between user and item in each platform and client. The purple ones are the users in the intersection set of clients' sub-graphs.

3.2 User-Item Graph K-Hop Expansion

To overcome the non-IID problem, FedRec privately exchanges the edges information between clients. The main idea is to expand the edges from the intersected users in different sub-graphs. In two-client setting, for example, the intersected users are the users appear in both sub-graphs. We integrate PSI to the K-hop extension, which avoid leaking the user-item information that is not in the intersection set.

Without loss of generality, suppose there are two clients, client i and client j, who exchange the edges information via K-hop extension and generate the extended sub-graph $\bar{G}^i$ and $\bar{G}^j$. The vertex set and edge set of user-item graph G^i are V^i and E^i respectively, and client j also records the $G^j = (V^j, E^j)$. Firstly, client i and client j will execute PSI protocol to get the intersection of V^i and V^j, denoted as $V^{i,j}$, *i.e.*, $V^{i,j} = \mathsf{PSI}(V^i, V^j)$ (Line 4 in Algorithm 1). Then, the K-hop extension is performed by extending the edges and vertexes on $V^{i,j}$. The extended vertex set $\bar{V}^{i,j}$ will cover the vertexes in V^j within K hops from $V^{i,j}$ (Line 5 in Algorithm 1). Similarly, the extended edge set $\bar{V}^{i,j}$ will cover the edges that both vertexes are in $\bar{V}^{i,j}$ (Line 6 in Algorithm 1). Next, the client

i and client j will exchange the extended vertex set and extended edge set (Line 7,10 in Algorithm 1). Lastly, the client i combines the extended vertexes and edges with G^i to form the new graph $\bar{G}^{(i)}$ (Line 11–14 in Algorithm 1). Through the exchange of edge information, the local model learns new information, and thus improves the accuracy of the global model.

However, the PSI only preserves the privacy of the edges the other clients do not own. It's not able to protect the privacy of the edges in the intersection set. Thus, we leverage DP to FedRec to extend FedRec to DP-FedRec via DP.

3.3 Privacy-Preserve User-Item Graph Sharing

Since the user-item graph contains sensitive information involving user privacy, the direct interaction of the user-item graph between clients will be strictly restricted due to privacy regulations. DP-FedRec applies different DP algorithms in both topology as well as the weights information to preserve the privacy of both.

For the topology of the graph, DP-FedRec adds noises to a weighted undirected graph using the LAPGRAPH algorithm. For simplicity, we denote the user-item connection matrix of the graph as M, where 0 indicates that there is no scoring relationship between the corresponding user and the corresponding item, and the vice versa is 1. DP-FedRec first calculates the sparsity degree $T = n_1$ where n_1 is the number of 1. Next, DP-FedRec adds Laplacian noise with a mean value of 0 and an intensity of λ_1 to each matrix element, and at the same time uses a part of the privacy budget (usually 1%) to protect the sparsity degree T from noise. We denote the sparse degree after adding noises is T'. Finally, DP-FedRec leaves the top T' large elements of the matrix after adding noise as 1, and others as 0.

For the protection of the edge weights information of the user-item graph, a Laplace noise with a mean value of 0 and an intensity of λ_2 is added directly to the new graph formed by the above algorithms.

4 Analysis

4.1 Privacy Analysis

The privacy of DP-FedRec is protected by the following aspects: (i) The vertexes outside of the intersection set during K-hop extension. The privacy of this part is guaranteed by PSI. (ii) The vertexes within the intersection set during K-hop extension. The privacy of this part is guaranteed by DP. The protection of privacy is divided into protection of the topology structure and protection of the weights of the edges. For preservation of topology, the Laplace noise is added to its adjacency matrix using the Lapgraph algorithm so that the information is perturbed. For protection of edge weights, the values of edge weights are disturbed by adding noises directly to the edge weights. It has been demonstrated in [23] that when the noise added satisfies $Lap(0, \frac{\Delta f_1}{\epsilon_1})$, the Lapgraph algorithm satisfies $\epsilon_1 - DP$. At the same time, due to the characteristics of Laplace mechanism [4], the noise added to the edge weights satisfies $Lap(0, \frac{\Delta f_2}{\epsilon_2})$, which is $\epsilon_2 - DP$. By Composition theorem [4], DP-FedRec satisfies $\epsilon_1 + \epsilon_2$-DP.

Algorithm 1. K-hop extension of client i

```
Input: $K$, the parameter of K-hop; the graph $G^i$
Output: the extended graph $\bar{G}^i$
procedure K-hop Extension($K, i$)
    $\bar{V}^i = V^i$, $\bar{E}^i = E^i$
    for $u_j \in \mathcal{U} \backslash \{u_i\}$ do
        $V^{i,j} = \mathsf{PSI}(V^i, V^j)$
        $\bar{V}^{i,j} = \bar{V}^{i,j} \cup \{v|\mathsf{dist}(v, V^{i,j}, G^i) \leq k \wedge v \in V^i\}$
        $\bar{E}^{i,j} = \bar{E}^{i,j} \cup \{< x, y > |x, y \in \bar{V}^{(i,j)}\}$
        Send $(\bar{V}^{i,j}, \bar{E}^{i,j})$ to client $j$
    end for
    for $u_j \in \mathcal{U} \backslash \{u_i\}$ do
        Receive $(\bar{V}^{j,i}, \bar{E}^{j,i})$ from client $j$
        $\bar{V}^i = \bar{V}^i \cup \bar{V}^{j,i}$
        $\bar{E}^i = \bar{E}^i \cup \bar{E}^{j,i}$
    end for
    return $\bar{G}^i = (\bar{V}^i, \bar{E}^i)$
end procedure
```

4.2 Performance Analysis

First, consider the time complexity of DP-FedRec for one client. Since a certain amount of noise needs to be added to each element of the adjacency matrix, the time for single addition of noise is $\mathcal{O}(|V^i|^2)$. Then analysis is performed on the communication complexity of DP-FedRec for client i. Since DP-FedRec requires interaction between two clients, communication cost of such interaction between clients is $\mathcal{O}(l^2)$. Each interaction contains PSI, K-hop extension, and adding noise towards expanded graph data. Correspondingly, the time complexity of PSI is $\mathcal{O}(|V^i|)$, the time complexity of K-hop extension is $O(|V^i|)$. The time for single addition of noise, as analyzed above, is $\mathcal{O}(|V^i|^2)$. So the communication cost of DP-FedRec is $\mathcal{O}(l^2 \cdot |V^i|^2)$.

5 Evaluation

5.1 Evaluation Setup

Implementation. We implement both FedRec and DP-FedRec via Python based code of FedGraphNN [7]. We conduct the evaluation on a computation instance equipped with 2.1 GHz 64 Intel(R) Xeon(R) Gold 6130 CPU, 512 GB memory and 8 Tesla V100 GPU with 12 GB.

Dataset. We conduct evaluation on two datasets: Epinions [19] and MovieLens [6]. The Epinions dataset contains consumers' ratings on items from the Epinions website. The MovieLens dataset contains the users' rating of different movies from the MovieLens website. For both datasets, we divide the graph to different clients via the item category, *i.e.*, the items with same category and their relevant users will be assigned to the same client. In particular, due to the large number of points in the epinions dataset and the limitation of memory, we only select 12 categories for experiments. Table 2 shows the average number of users, items and edges after separation.

Table 2. Dataset description. The K is the parameter of K-hop, n is the number of users, the m is the number of items and #edges is the number of edges. For centralized, the number of user, items and edges is the total number.

Dataset	Number of clients	K	Average n for each client	Average n for each client	Average #edges for each client
Epinions	Centralized	/	21296	163874	870838
	8	2	21052	117641	754303
		10	21172	163588	870266
	12	2	21007	105897	720281
		10	21165	163539	870227
MovieLens 1M	Centralized	/	6040	3706	1000209
	8	1	5894	3704	995298
		5	6040	3706	1000209
	12	1	5409	3699	972759
		5	6040	3706	1000209

Setting of Experiments. Our evaluation goal is to prove two claim: the K-hop extension improves the accuracy of the federated GNN and the leverage of DP in DP-FedRec do not reduce the accuracy too much. The experiments is conducts under two client settings: 8 and 12 clients. Five experiments were performed in each setting: (i) Centralized training, the central server owns the full graph for training; (ii) FedGraphNN with FedAvg, the structure proposed in [7], which is also the baseline we compared. For simplicity we denote it as FedGraphNN in the remaining sections; (iii) FedRec, where we only perform K-hop extension without adding noise to the interactive content. The purpose of this experiment is to demonstrate that the K-hop extension helps to increase the accuracy of link prediction; (iv) DP-FedGraphNN, we add Laplace(0, 1) noise to the edge weights of the user-item graph based on FedGraphNN as a baseline to compare with DP-FedRec. (v) DP-FedRec with different K, which is realized by adding noise to the interactive content on the basis of the third group of FedRec. For evaluation metrics, we adopt mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE) to evaluate the accuracy of edge weights prediction and record the average time it takes to add noise to a single client in each experiment.

5.2 Performance of K-Hop Extension

Table 3 and Table 4 show the performance of centralized server, FedRec and FedGraphNN. It indicates that FedRec performed better than FedGraphNN in all metrics. The result proves the K-hop extension does really help to handle the non-IID problem in federated learning and thus improve the link prediction accuracy.

The effect of DP-FedRec is also better than FedGraphNN, which proves that the K-hop extension algorithm based on local differential privacy improves the performance of the model while protecting privacy. Meanwhile The K-Hop extension is very robust even with adding noise to the graph data.

Table 3. Performance of different systems with 8 clients. Noising time refers to the time of adding noise per client.

Dataset/8 clients	Model type	System	MAE	MSE	RMSE	Noising time (s)
Epinions	W/O DP	Centralized	0.8377	1.2464	1.1164	/
		FedGraphNN	0.8719	1.3424	1.1559	
		FedRec	0.8643	1.3303	1.1505	
	W/ DP	DP-FedGraphNN	0.8724	1.3484	1.1584	/
		DP-FedRec (K=2)	0.8689	1.3415	1.1560	328
		DP-FedRec (K=10)	0.8658	1.3278	1.1523	517
MovieLens1M	W/O DP	Centralized	0.8812	1.1782	1.0855	/
		FedGraphNN	0.8832	1.1850	1.0884	
		FedRec	0.8793	1.1786	1.0884	
	W/ DP	DP-FedGraphNN	0.8912	1.2057	1.0980	/
		DP-FedRec (K=1)	0.8820	1.1798	1.0862	4
		DP-FedRec (K=5)	0.8813	1.1783	1.0875	4

5.3 Performance of Differential Privacy

From the results, the performance of DP-FedRec does not decrease much than FedRec. However, after adding noise to FedGraphNN, accuracy drops a lot.

Table 4. Performance of different systems with 12 clients. Noising time refers to the time of adding noise per client.

Dataset/12 clients	Model type	System	MAE	MSE	RMSE	Noising time (s)
Epinions	W/O DP	Centralized	0.8377	1.2464	1.1164	/
		FedGraphNN	0.8674	1.3279	1.1502	
		FedRec $(K = 10)$	0.8635	1.3270	1.1496	
	W/ DP	DP-FedGraphNN	0.8716	1.3298	1.1513	/
		DP-FedRec $(K = 2)$	0.8625	1.3261	1.1493	314
		DP-FedRec $(K = 10)$	0.8585	1.3258	1.1493	501
MovieLens1M	W/O DP	Centralized	0.8812	1.1782	1.0855	/
		FedGraphNN	0.8931	1.2454	1.1152	
		FedRec $(K = 5)$	0.8874	1.1850	1.0883	
	W/ DP	DP-FedGraphNN	0.8989	1.2669	1.1247	/
		DP-FedRec $(K = 1)$	0.8936	1.2257	1.1063	3
		DP-FedRec $(K = 5)$	0.8907	1.1991	1.0948	4

When the number of clients in the Epinions dataset is 12, the effect of DP-FedRec is better than that of FedRec, which to a certain extent shows that the noise added in the experiment not only protects the privacy of the data from being leaked, but also ensures the data availability is not compromised, reflecting the balance between data privacy and availability.

We also recorded the average time for each client to add noise. According to our analysis, the time for adding noise is positively correlated with the number of points in the graph, *i.e.*, the number of points increases, the time it takes to add noise will increase, while the increase in the number of edges does not significantly increases the time it takes to add noise.

In the Epinions dataset and MovieLens1M dataset, the number of points of Epinions is significantly larger than that of MovieLens1M, and the number of edges of MovieLens1M is significantly larger than the number of points of Epinions. Summarizing the average time to add noise per client in the experiments, we found that the time required to add noise to the Epinions dataset is much greater than that required for Movielens which is consistent with our analysis.

6 Related Work

6.1 Federated Recommendation System

Federated Learning is being applied in lots of field [11,21]. And as the laws and regulations of data and privacy become stricter, recommendation systems based on federated learning with privacy-preserving features have become a hot research trend. FCF [1], a classic federated recommendation system, is the first collaborative filtering framework based on the federated learning paradigm. They build a joint model by using user implicit feedback. [14] is a privacy-preserving method which leverages the behavior data of massive users and meanwhile don't require centralized storage to protect user privacy to train news recommendation model with accuracy. FedFast [13] achieves high accuracy for each user during the federated learning training phase as quickly as possible. In each training round, They sample from a set of participating clients and apply an active aggregation method that propagates the updated model to the other clients.

6.2 Differential Privacy Graph Neural Network

Several literature leverage DP to preserve the privacy of GNN. Solitude [12] is a privacy-preserving learning framework based on GNN, with formal privacy guarantees based on edge local differential privacy. The crux of Solitude is a set of new delicate mechanisms that calibrate the introduced noise in the decentralized graph collected from the users. LDPGen [15] is a multi-phase technique that incrementally clusters users based on their connections to different partitions of the whole population. LDPGen carefully injects noise to ensure local differential privacy whenever a user reports information. There are only few works that combine the DP with the GNN federated learning. [22] applies differential privacy techniques to the local gradients of GNN model to protect user privacy in federated learning setting. But it need a third-party server to store embedding of users besides training server. So FedGNN is a two-server model. In [26], They propose (VFGNN), a federated GNN learning model for privacy-preserving node classification task under data vertically partitioned setting. They leave the private data related computations on data holders, and delegate the rest of computations to a semi-honest server. However, their work has an strong assumption that every data holders have the same nodes, which is far different from real scenario.

7 Conclusion and Future Work

In this paper, we proposed DP-FedRec a privacy-preserving federated GNN framework for recommendation system. To overcome the challenge of the Non-IID problem under the privacy regulation, DP-FedRec integrates the PSI and the DP technique with the federated GNN. The PSI-based K-hop extension helps to extend the sub-graph of each client without leaking any non-intersection information to solve the non-IID problem. Moreover, DP preserves not only the privacy of weights but also the privacy of edges/topology in the intersection information to guarantee the privacy. We implemented the prototype of DP-FedRec and tests it on different datasets. Compared with other works, the evaluation shows DP-FedRec achieves high performance and only induces little computations overhead. In the future, we would like to investigate a universal DP for both weights and edges in graph data for better performance.

Acknowledgement. This paper is supported by the Key Research and Development Program of Guangdong Province under grant No. 2021B0101400003.

References

1. Ammad-Ud-Din, M., et al.: Federated collaborative filtering for privacy-preserving personalized recommendation system (2019)
2. Chen, J., Ma, T., Xiao, C.: FastGCN: fast learning with graph convolutional networks via importance sampling. In: 6th International Conference on Learning Representations, ICLR 2018 (2018)
3. Cheung, T.H., Dai, W., Li, S.: FedSGC: federated simple graph convolution for node classification. In: International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjuncation with IJCAI 2021, FTL-IJCAI 2021 (2021)
4. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2013)
5. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: 31st Conference on Neural Information Processing Systems, NeurIPS 2017 (2017)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TiiS) **5**(4), 1–19 (2015)
7. He, C., et al.: FedGraphNN: a federated learning benchmark system for graph neural networks. In: ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML) (2021)
8. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017 (2017)
10. Kolesnikov, V., Matania, N., Pinkas, B., Rosulek, M., Trieu, N.: Practical multi-party private set intersection from symmetric-key techniques. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (2017)
11. Kong, L., Tao, H., Wang, J., Huang, Z., Xiao, J.: Network coding for federated learning systems. In: Neural Information Processing - 27th International Conference, ICONIP 2020 (2020)
12. Lin, W., Li, B., Wang, C.: Towards private learning on decentralized graphs with local differential privacy. CoRR abs/2201.09398 (2022)
13. Muhammad, K., Wang, Q., O'Reilly-Morgan, D., Tragos, E., Lawlor, A.: FedFast: going beyond average for faster training of federated recommender systems. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2020 (2020)
14. Qi, T., Wu, F., Wu, C., Huang, Y., Xie, X.: Privacy-preserving news recommendation model learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (2020)
15. Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X., Ren, K.: Generating synthetic decentralized social graphs with local differential privacy. In: ACM SIGSAC Conference on Computer and Communications Security (2017)
16. Qiu, H., Qiu, M., Zhihui, L.U.: Selective encryption on ECG data in body sensor network based on supervised machine learning. Inf. Fusion **55**, 59–67 (2020)
17. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. Future Gener. Comput. Syst. **87**, 772–781 (2017)

18. Qiu, M., Zhang, L., Ming, Z., Chen, Z., Qin, X., Yang, L.T.: Security-aware optimization for ubiquitous computing systems with seat graph approach. J. Comput. Syst. Sci. **79**(5), 518–529 (2013)
19. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: International Semantic Web Conference (2003)
20. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
21. Wang, J., Huang, Z., Kong, L., Li, D., Xiao, J.: Modeling without sharing privacy: federated neural machine translation. In: International Conference on Web Information Systems Engineering (2021)
22. Wu, C., Wu, F., Cao, Y., Huang, Y., Xie, X.: FedGNN: federated graph neural network for privacy-preserving recommendation. In: International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021, FL-ICML 2021 (2021)
23. Wu, F., Long, Y., Zhang, C., Li, B.: Linkteller: recovering private edges from graph neural networks via influence analysis. In: Proceedings of the Symposium on Security and Privacy (2021)
24. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
25. Yang, C., Wang, H., Zhang, K., Chen, L., Sun, L.: Secure deep graph generation with link differential privacy. In: The 30th International Joint Conference on Artificial Intelligence, IJCAI 2021 (2021)
26. Zhou, J., et al.: Vertically federated graph neural network for privacy-preserving node classification. In: The 31st International Joint Conference on Artificial Intelligence, IJCAI 2022 (2022)