User Consented Federated Recommender System Against Personalized Attribute Inference Attack

Qi Hu
Department of CSE, HKUST
Hong Kong SAR, China
qhuaf@connect.ust.hk

Yangqiu Song
Department of CSE, HKUST
Hong Kong SAR, China
yqsong@cse.ust.hk

ABSTRACT

Recommender systems can be privacy-sensitive. To protect users' private historical interactions, federated learning has been proposed in distributed learning for user representations. Using federated recommender (FedRec) systems, users can train a shared recommendation model on local devices and prevent raw data transmissions and collections. However, the recommendation model learned by a common FedRec may still be vulnerable to private information leakage risks, particularly attribute inference attacks, which means that the attacker can easily infer users' personal attributes from the learned model. Additionally, traditional FedRecs seldom consider the diverse privacy preference of users, leading to difficulties in balancing the recommendation utility and privacy preservation. Consequently, FedRecs may suffer from unnecessary recommendation performance loss due to over-protection and private information leakage simultaneously. In this work, we propose a novel *user-consented federated recommendation system* (UC-FedRec) to flexibly satisfy the different privacy needs of users by paying a minimum recommendation accuracy price. UC-FedRec allows users to self-define their privacy preferences to meet various demands and makes recommendations with user consent. Experiments conducted on different real-world datasets demonstrate that our framework is more efficient and flexible compared to baselines. Our code is available at https://github.com/HKUST-KnowComp/UC-FedRec.

CCS CONCEPTS

• **Security and privacy**; • **Information systems** → **Collaborative filtering**;

KEYWORDS

Federated Learning, Privacy-preserving, Recommender Systems, Collaborative Filtering.
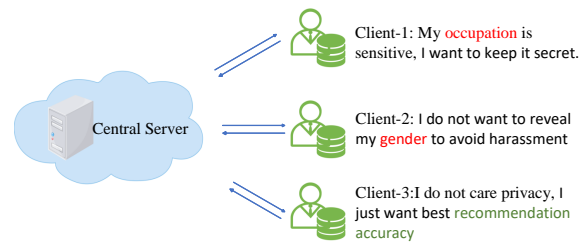
Figure 1: Overview of the user consented FedRec framework. Different users have various privacy or utility preferences.

1 INTRODUCTION

Recommender systems have gained considerable popularity in predicting users' interests in online services, such as e-commerce and social media [40]. Recently, deep learning based recommendation algorithms, which use interactions of users and items, and/or attributes with a parameterized network to predict users' preferences, have proven to be effective [21, 53]. However, the historical user and item interaction data and some users' attribute data are highly privacy-sensitive. With the growing attention to privacy preservation and the changes in privacy regulations such as the General Data Protection Regulation (GDPR), balancing privacy protection and recommendation accuracy is becoming increasingly critical [41, 51].

To protect users' raw data, federated learning has been proposed, which only exchanges the gradients between users and the server [26, 32]. Based on federated learning, federated recommender (FedRec) systems have been developed to address privacy issues by decentralizing the training process. In such systems, raw user-item interaction data is decoupled from the model training [13, 35]. Despite avoiding raw data collection and transmission, several issues still need to be addressed with FedRec systems. First, the personalization of users learned by FedRec, based on their past behaviors, poses the risk of user-level private information leakage [2, 18, 52]. Research has shown that a high-quality personalized federated learning model is vulnerable to attribute inference attacks to reveal participants' personal information [7, 45]. We use attackers to predict users' attributes from some FedRecs [34, 35] trained on MovieLens [19], the results in Table 1 show that those FedRecs face attribute inference attacks. Second, there is a tradeoff between privacy protection and recommendation utility. However, privacy demands differ among participants, and existing FedRec systems provide the same privacy protection for all users without considering their specific demands. This results in users with strong privacy demands are not satisfied, whereas users with weak privacy demands pay an unnecessary recommendation performance price.

Table 1: Attribute inference attacks on FedRecs

FedRecs	Utility		Privacy		
	NDCG	Recall	Gender	Age	Occupation
LDP-Rec	0.694	0.433	0.685	0.296	0.136
Fedfast-BPR	0.72	0.46	0.745	0.353	0.15
Random Attacker			0.5	0.141	0.05

Third, every user's privacy demands are not static, and traditional FedRec systems mainly focus on privacy protection in the training stage, making it difficult for users to modify privacy settings [34, 36]. Consequently, the balance between utility and privacy is not flexible for users in traditional FedRec systems.

To meet the personalized privacy demands of various users, we propose a new framework called User-Consented Federated Recommender System (UC-FedRec), where users can flexibly safeguard their sensitive personal information and decide on the trade-off between individual privacy and recommendation accuracy. The UC-FedRec framework aims to satisfy users' diverse privacy demands while paying minimum recommendation accuracy costs. As illustrated in Figure 1, clients have various privacy demands and utility preferences. Thus, the framework has a dual objective: 1) to meet privacy needs and protect private clients from inference attacks, and 2) to avoid other clients paying the unnecessary cost and retaining high-quality recommendation results. To achieve this goal, we leverage the different privacy preferences of participants to train a set of sensitive attribute information filters. These filters can be optionally applied to the local recommendation models on client sides according to their privacy demands during the local training stage, so the model can flexibly eliminate the sensitive attribute information that users care about.

We summarize our main contributions as follows:

- To the best of our knowledge, our work is the first user-consented FedRec framework. By applying sensitive attribute filters, users can flexibly protect their personal sensitive private information with proper configuration.
- We introduce an adversarial framework in FedRec and propose a personalized privacy-aware algorithm to meet users' diverse privacy and utility preferences. Our framework can meet the different privacy needs of users while minimizing the recommendation accuracy cost.
- We quantify the privacy leakage problem in FedRec and evaluate the proposed framework in privacy protection and utility on two real-world datasets. The results indicate that the framework can flexibly deal with various and changing user privacy preferences.

2 RELATED WORK

In this section, we briefly summarize the related work. Our work is closely related to collaborative filtering and privacy-preserving systems.

2.1 Collaborative Filtering

Collaborative filtering (CF) is one of the most popular approaches in recommender systems and has been widely used in real-world systems. Having the assumption that people with similar historical interactions will have similar preferences, CF models, such as those

proposed in [5, 15, 21] parameterize users and items and their interactions by vectorized representations. Based on the representations, CF models predict the interactions by vector computations like inner product [27]. To improve the recommendation accuracy and solve the cold-start problem, existing works focus on improving the quality of the representations. For example, some studies made use of side information such as user/item relations [17, 48] and external knowledge graph [43]. Some works [21, 49] utilized deep learning models to extract interaction features in user-item interactions. Some GNN-based collaborative filtering techniques [5, 10, 44] were proposed to exploit the high-order connectivity from user-item interactions. Most collaborative filtering techniques are centralized, requiring users to share their private historical interactions with the server. With the increasing concern about the privacy problem, decentralized collaborative models [1, 8] were proposed to avoid sensitive data sharing. However, the models learned by the recommender system are personalized and are unbalanced for sensitive variables [6, 9, 12] and federated learning faces the problem same as the central system.

2.2 Privacy-Preserving Systems

To overcome the privacy leakage problem, various frameworks are proposed. One direction targets the private information in learned models. For example, some studies use perturbation and differential privacy to prevent an adversary from inferring a targeted user's private information [3, 24, 30, 54]. Some works propose to use adversarial learning to protect users' private personal attributes [4, 20]. Some propose to disentangle the private information from utility tasks [22, 23, 33]. However, these methods provide privacy preservation in central learning and cannot be applied in distributed learning where personal data is kept on local devices. To overcome the central data collection problem, federated learning was proposed, allowing data owners to collaboratively build a shared privacy-preserving decentralized model in distributed. [26, 32, 50]. Many works are proposed to learn federated recommender systems with various privacy guarantees [1, 35, 46]. Differential privacy (DP) [13, 46] is commonly applied in the context of distributed computing. It adds random noise to true data records such that two arbitrary records have close probabilities to generate the same noisy data record. It provides data anonymous that will not reveal individual information in sensitive information collection and analysis [39]. However, DP in federated learning is designed for the model weights or update of information transmission and is not suitable for the problem of inference attacks and privacy leakage in the learned federated recommender model.

3 USER CONSENTED FEDERATED RECOMMENDATION

In this section, we present the user-consented FedRec. It meets users' privacy demands by utilizing different privacy preferences to eliminate specific sensitive information accordingly.

3.1 Preliminary

Following general settings of recommender systems [21, 25, 44], we denote a recommender system R containing a set of users and items represented by $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ and $\mathcal{I} = \{i_1, i_2, \cdots, i_{|\mathcal{I}|}\}$

respectively. Denote the interaction matrix between users and items as $Y \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{I}|}$, where the value of 1 indicates that there is an interaction between corresponding user u and item i while the value 0 means that user u is not interested in the item i or user is not aware of the existence of item i. Due to privacy concerns, federated learning is adopted. Users' data is stored in local devices to protect privacy, each user u holds its own historical interaction records Y_u. The recommendation model can be summarized as estimating user u's preference on any item i by learned latent user representations $h_u = f_\theta(u) \in \mathbb{R}^d$ and item representations $h_i = f_\theta(i) \in \mathbb{R}^d$, where d denotes the representation size, so that:

$$\hat{y}_{u,i} = s_\psi(h_u, h_i), \tag{1}$$

where the scoring function $s_\psi(\cdot)$ can be dot product, multi-layer perceptions, etc., and $\hat{y}_{u,t}$ denotes the preference score for user u on unobserved item i which is usually presented in probability [47]. Representation function f_θ commonly has two parts: embedding layer which maps user/item to vectors e_u/e_i and propagation layer which catches collaborative signals. The recommendation objective can be formalized as:

$$\min_{\theta, \psi} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathcal{L}(s_\psi(f_\theta(u), f_\theta(i)), Y), \tag{2}$$

where $\mathcal{L}$ can be commonly used loss functions in recommender systems (e.g., BPR loss for implicit feedback [44]). To solve the cold-start problem and improve recommendation accuracy, it is common for recommender systems to investigate a set of user attributes (e.g., occupation, location) $\mathcal{T}$. However, privacy can be a different definition for users [38]. Though the information is kept on local devices, some users may find some attributes are sensitive and want to eliminate the information in FedRec. We denote user u's personal sensitive attributes as $\mathcal{T}_u \subseteq \mathcal{T}$.

3.2 Problem Definition

Given a FedRec where users have different privacy preferences, each user u has its own private attribute set $\mathcal{T}_u$, and non-sensitive attributes $\mathcal{T} \setminus \mathcal{T}_u$ are revealed but are kept in local devices. The challenge is that if partial participants' attributes are leaked, the attacker can easily infer the unknown sensitive attributes that private users prefer not to reveal from the FedRecs. This is mainly because collaborative filtering models can catch the signals from similar behavior users. UC-FedRec aims to eliminate the information of sensitive attributes in the training stage with minimum recommendation utility cost so that the sensitive attribute leakage risks will be reduced.

3.3 Basic Framework

Next, we introduce the basic structure of our proposed framework. UC-FedRec is a supplement of common FedRecs and uses arbitrary embedding-based FedRec as the base model. Compared to traditional FedRec, the critical module of UC-FedRec is the compositional sensitive attribute filters (distribution estimators). It can leverage the different privacy needs to learn sensitive attribute filters to protect private information from inference attacks. As shown in Figure 2, it mainly consists of a central server and a set of user clients. On the client side, the filters are flexibly applied or trained by all the users according to their privacy preferences. User u's privacy

is protected by the $\mathcal{T}_u$ feature filters. These filters can eliminate the sensitive information in the representations. When the user regards some attributes as non-private, it takes the responsibility of training $\mathcal{T} \setminus \mathcal{T}_u$ feature distribution estimators. Each user learns user/item embeddings from its interaction data and feature extractors from its non-private attributes, and uploads filtered gradients of embeddings and non-private feature extractors. On the server side, the central server is responsible for aggregating the gradients and distributing the updated global models to clients.

3.4 Compositional Privacy Protection

3.4.1 ***Non-private attributes.*** In practice, FedRec faces the problem of sparsity and cold-start [53]. It is common for FedRec to utilize user personal information to improve recommendation quality. Some users are willing to choose a part of non-private attributes to reveal for a better experience. Suppose user u reals $\mathcal{T} \setminus \mathcal{T}_u$ features. We define the attribute probability distribution when given user u's representations as $p(y_{u,t}|h_u)$, where $y_{u,t}$ is the label of attribute t for user u. As the real distribution is unknown, and unfortunately the computation is expensive and in most cases intractable, it is necessary to introduce an auxiliary distribution $q_{\phi_t}(y_{u,t}|h_u)$ to approximate the posterior distribution. Users are responsible for the training of non-private attribute distribution estimators. We aim to have the $q_{\phi_t}(y_{u,t}|h_u)$ as close as possible to $p(y_{u,t}|h_u)$. Therefore, we use the Kullback–Leibler divergence (KL divergence) to measure the difference between two distributions [28] and minimize the distance:

$$\begin{aligned} &\min_{\theta, \phi_t} KL(p(y_{u,t}|h_u) \| q_{\phi_t}(y_{u,t}|h_u)) \\ =&\min_{\theta, \phi_t} \mathbb{E}_{p(y_{u,t}|h_u)} \log p(y_{u,t}|h_u) - \mathbb{E}_{p(y_{u,t}|h_u)} \log q_{\phi_t}(y_{u,t}|h_u) \\ \Leftrightarrow&\min_{\theta, \phi_t} -\mathbb{E}_{p(h_u, y_{u,t})} [\log q_{\phi_t}(y_{u,t}|h_u)] \quad \forall t \in \mathcal{T} \setminus \mathcal{T}_u. \end{aligned} \tag{3}$$

To solve the objective function, we parameterize the q_{ϕ_t} function using a node classifier g_{ϕ_t} defined on the user representations. Suppose a set of users $\mathcal{U}_t$ consider t as non-private attributes, then the objective function can be estimated as:

$$\min_{\theta, \phi_t} \sum_{u \in \mathcal{U}_t} CE(g_{\phi_t}(f_\theta(u)), y_{u,t}), \tag{4}$$

where $CE(\cdot)$ is the cross entropy loss.

3.4.2 ***Private attributes.*** The FedRec is at risk of leaking privacy, the fundamental reason is that the representations generated by the FedRec contain plenty of private information. Therefore, our goal is to eliminate the sensitive information in FedRec. The sensitive attributes are expected to be independent from the learned user u's representations, that is minimizing the mutual information $I(h_u; t)$ learned by FedRec, generalized to multiple attributes:

$$\min_\theta \sum_{t \in \mathcal{T}_u} I(h_u; t). \tag{5}$$

To solve the Equation (5), we leverage the upper bound $I_{vCLUB}(h_u; t)$ proposed in [11] and the minimizing attribute mutual information
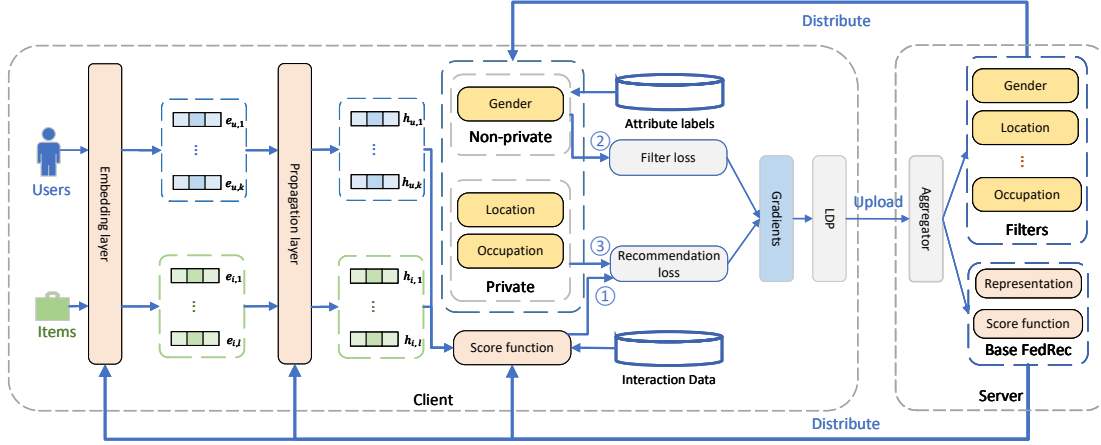
Figure 2: The overall architecture of UC-FedRec. The clients' training has three learning targets: ① to train the recommendation part as the primary goal; ② to train attribute distribution estimator on personal non-private attributes; ③ to utilize estimators to eliminate private information in the representation model. Finally, it uploads the gradients to the server. The server aggregate and distribute the gradients after receiving local clients' upload.

problem is equivalent to minimize the least upper bound:

$$
\begin{aligned}
&\min_{\theta} \sum_{t \in \mathcal{T}_u} I(h_u; t) \\
\Leftrightarrow &\min_{\theta} \sum_{t \in \mathcal{T}_u} I_{v\text{CLUB}}(h_u; t).
\end{aligned}
\tag{6}
$$

To solve the Equation (6), we utilize a objective function proposed in [42]:

$$
\min_{\theta} \sum_{t \in \mathcal{T}_u} \max_{\phi_t} \mathbb{E}_{p(h_u,t)} \left[\log q_{\phi_t}(y_{u,t}|h_u) \right],
\tag{7}
$$

where the posterior distribution q_{ϕ_t} is estimated by parameterized neural networks g_{ϕ_t} which are trained by other users. Therefore, users do not update the estimator g_{ϕ_t} for private attributes. Specifically, we have:

$$
\begin{aligned}
&\min_{\theta} \sum_{t \in \mathcal{T}_u} \max_{\phi_t} \mathbb{E}_{p(u,t)} \left[\log q_{\phi_t}(y_{u,t}|h_u) \right] \\
\approx &\max_{\theta} \sum_{t \in \mathcal{T}_u} CE(g_{\phi_t}(h_u), y_{u,t}).
\end{aligned}
\tag{8}
$$

As the labels $y_{u,t}$ are kept secret, the objective function is not practical in the training, we transform the objective function to unsupervised training. To maximize the cross entropy loss, we aim to raise the uncertainty of the estimated distribution:

$$
\begin{aligned}
&\max_{\theta} \sum_{t \in \mathcal{T}_u} CE(g_{\phi_t}(h_u), y_{u,t}) \\
\approx &\min_{\theta} \sum_{t \in \mathcal{T}_u} KL(g_{\phi_t}(f_\theta(u)), \hat{y}_{u,t}),
\end{aligned}
\tag{9}
$$

where $\hat{y}_{u,t} \sim \mathbb{U}$, which is the discrete uniform distribution.

*3.4.3 **Objective function**.* Equations (2), (4), and (9) are three sub-targets in UC-FedRec which correspond to three motivations respectively: recommendation performance, attribute distribution estimation, and private information elimination. For each user, we formulate two local training processes to meet the private representation training objective.

We solve the Equation (4) for attribute distribution estimator training. We sample a set of users u from $\mathcal{U}_t$ and the corresponding attribute labels $y_{u,t}$, disentangle the training objective, and view the user representation $f_\theta(u)$ as the input. We have:

$$
\phi_t = \arg\min_{\phi_t} \sum_{u \in \mathcal{U}_t} CE(g_{\phi_t}(h_u), y_{u,t}).
\tag{10}
$$

To solve the Equation (2) and Equation (9). We sample a set of users from $\mathcal{U}$ and their historical interaction data. Combining two objectives as joint learning, the private representation training objective for clients is as follows:

$$
\begin{aligned}
\theta, \psi = \arg\min_{\theta} (\beta \min_{\psi} &\mathcal{L}(s_\psi(f_\theta(u), f_\theta(i)), \mathbf{Y}) \\
&+ (1-\beta) \sum_{t \in \mathcal{T}_u} KL(g_{\phi_t}(f_\theta(u)), \hat{y}_{u,t})),
\end{aligned}
\tag{11}
$$

where $\beta \in [0, 1]$ is the trade-off between recommendation utility and privacy protection. A larger β indicates stronger privacy protection while a *smaller* β has a better recommendation accuracy. Note that the weight of each attribute's privacy loss can be adjusted. Here we assume that users weigh all the private attributes the same for simplicity.

3.5 Training Algorithm

There are three types of neural network in UC-FedRec: recommendation representation network f_θ, recommendation score function

s_ψ are learned for recommendation, and attribute distribution estimator g_{ϕ_t} is for sensitive attribute information protection. We adopt joint learning to train three networks. The UC-FedRec training process can be divided into two scenarios: distributed learning and central learning.

3.5.1 **Distributed learning**. To solve the learning objective proposed in Section 3.4, we make a complement to common FedRecs. Similar to a general federated learning system, the central server is responsible for coordinating the training process. For example, sampling a set of users participating in a training round, distributing parameters, aggregating and updating model weights [32]. Compared to the FedRec used as the base model, the server takes on extra filters updating. The server needs to aggregate gradients from those non-private users for filters and distribute them to all the users for privacy-preserving learning.

For clients, local training updates three types of neural networks. Clients perform several iterations to update non-private attribute filters weights $\phi_t, \forall t \in \mathcal{T} \backslash \mathcal{T}_u$, and compute the multi-task loss to update representation weights θ and score function weights ψ in minibatch training. The filters eliminate specific user attribute information in representations. We iteratively update models' weights for local epoch E times. Finally, the client adds Laplace noise to model weights and uploads the perturbed weights to the server. Similar as other federated learning system, the gradients are protected by LDP [34, 46].

3.5.2 **Central protection**. As users' privacy preferences change over time, UC-FedRec needs to provide protection promptly when users find an originally non-private attribute sensitive. To reduce the communication consumption, the server directly uses the learned filters to eliminate the sensitive information without federated learning. Assume that user u requests additional attribute t preservation, similar as the Equation (9), we have the objective function:

$$\min_{e_u} KL(g_{\phi_t}(u), \hat{y}_{u,t}). \tag{12}$$

The propagation layer contains the collaborative signals, updating weights will inevitably influence others' recommendation accuracy, therefore, in the central protection, we choose to simply update user's embedding layer rather than the whole representation neural network to minimize the influence on the whole recommendation model. We iteratively update user embeddings until the difference reaches a threshold ($|e'_u - e_u| \leq T$).

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of the proposed UC-FedRec. We aim to answer the following questions: **Q1:** Whether UC-FedRec can protect personalized privacy while maintaining high recommendation utility compared to its base FedRec model? **Q2:** How does UC-FedRec perform on different privacy preferences such as utility-privacy tradeoff, various private attributes, etc? **Q3:** Whether UC-FedRec can provide prompt protection when users' privacy preferences change?

We first introduce experimental setup and metric in section 4.1, then we evaluate UC-FedRec in sections 4.3, 4.4, and 4.5 to answer three questions respectively.

4.1 Dataset and Experiment Setting

4.1.1 **Dataset**. We select two real-world recommendation datasets with users' side information: MovieLens [19] and Douban [31]. Both datasets are widely used in recommendation system evaluation. MovieLens contains 1,000,209 ratings by 6,040 users on 3,952 items. We treat users' gender, occupation and age as private attributes. Douban includes 647,263 interactions by 6,368 users with 22,347 items, location is treated as privacy. We transform the datasets into implicit data where each observed rating is treated as a positive instance and indicated by an interaction signal. Besides, we treat user attributes as sensitive information. As some users lack attribute information, we only retain users with all features so as for convenient privacy evaluation. For each user, we randomly select a set of features as the private attributes with probability α and the remaining as the non-private attributes. In such a way, we can simulate various compositional privacy preferences. In the evaluation, if there is no further statement, we set $\alpha = 0.3$.

4.1.2 **Hyperparameter setting**. In the experiment, if there is no further statement, we use the following implementation settings. We use FedGNN [46] and FedNCF [36] as our base FedRecs and use dot product as the scoring function. The dimension of user and item embeddings and their hidden representations learned by FedRecs are 128. We follow the technique proposed in [37] to achieve the 1-LDP guarantees to protect the gradient transmission. We use SGD as the optimization algorithm with 0.1 learning rate. We use BPR loss to train the recommendation model. For the privacy-preserving part, we use 2-layer perceptrons (MLP) as attribute information filters and information leakage evaluation using SGD optimization with 0.01 learning rate. We set the privacy-utility tradeoff $\beta = 0.5$.

4.1.3 **Performance Evaluation**. To evaluate recommendation utility, we adopt the *leave-one-out* strategy which has been widely used in literatures [14]. We take out their latest interaction for the test set and use the remaining data for training. Since ranking all the items is time-consuming, we follow the common strategy [16, 21] that randomly samples 50 items that are not interacted with by the user and ranks the test item among the 50 items. The performance of a ranked list is adjusted by *Hit Ratio* (HR) [14] and *Normalized Discounted Cumulative Gain* (NDCG). We truncate the ranked list at 10 for both metrics. We calculate both metrics for each test user and report the average score. For privacy protection evaluation, we adopt attribute classifiers to evaluate the information leakage in the representations. We use neural networks to infer users' attributes. Binary attributes are evaluated using *Area under the ROC Curve* (AUC) and multi-class attributes are evaluated using *micro-F1*. Our goal is to prevent features from inference attacks, therefore lower scores indicate better privacy protection.

4.2 Problem Evaluation

We evaluate the attribute inference risks in several FedRecs and compare UC-FedRec's utility performance and private attribute protection ability to them. We select FedGNN [46], a graph-based FedRec on user-item graph expansion and private sharing, and FedNCF [36], a federated learning extension of neural collaborative filtering as our baselines. We implement these FedRecs basically following the original paper while setting the same embedding

Table 2: Performance on different methods in terms of recommendation and privacy.

Methods		MovieLens					Douban		
		Utility		Privacy			Utility		Privacy
Protection	Base Model	NDCG	Recall	Gender	Age	Occupation	NDCG	Recall	Location
-		**0.843**	**0.544**	0.817	0.489	0.171	**0.724**	**0.492**	0.265
Early stopping	FedGNN	0.794	0.509	0.787	0.364	0.158	0.692	0.483	0.249
UC-FedRec		0.783	0.509	**0.631**	**0.283**	**0.128**	0.688	0.482	**0.221**
-		**0.784**	**0.496**	0.755	0.358	0.152	0.696	0.489	0.263
Early stopping	FedNCF	0.742	0.471	0.653	0.284	0.131	0.686	0.476	0.245
UC-FedRec		0.734	0.471	**0.602**	**0.212**	**0.117**	0.683	0.477	**0.225**

(a) Evaluation on gender and HR@10   (b) Evaluation on age and HR@10   (c) Evaluation on occupation and HR@10

(d) Evaluation on gender and NDCG@10   (e) Evaluation on age and NDCG@10   (f) Evaluation on occupation and NDCG@10
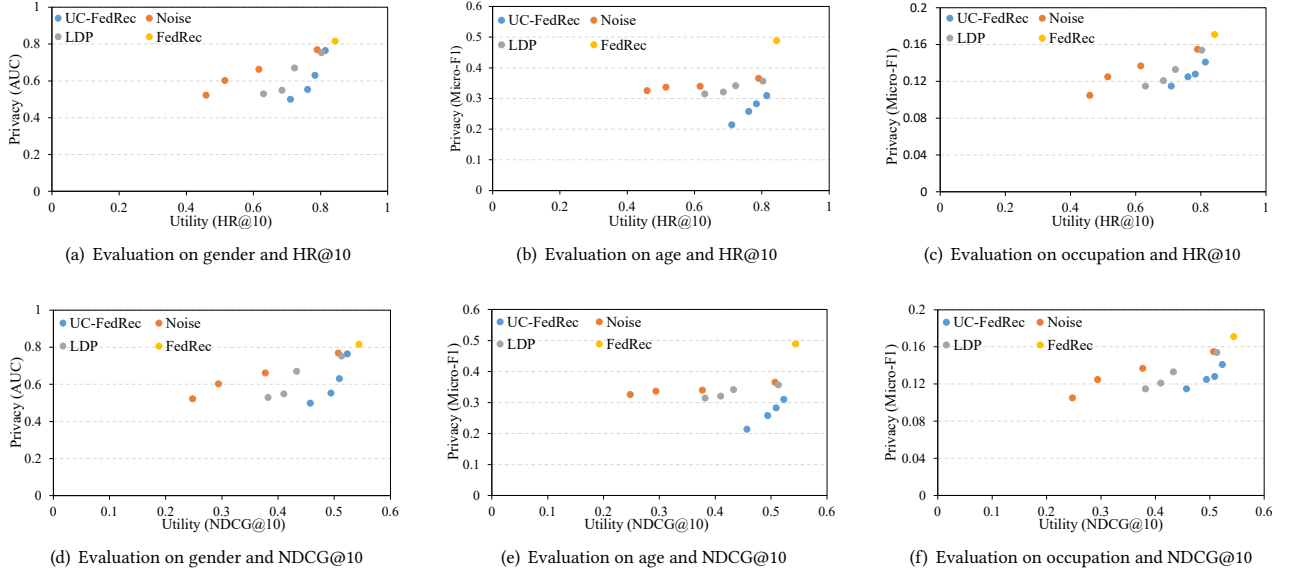
Figure 3: Utility-privacy tradeoff comparison of the framework with baselines on MovieLens.

dimension for fair comparison. We apply UC-FedRec framework to both base models to evaluate the effectiveness. Besides, we also report the performance of FedGNN and FedNCF with early stopping to maintain similar recommendation performance (recall) for convenient comparison. The results are summarized in Table 2. Compared to the base model, UC-FedRec can efficiently protect private information while sacrificing acceptable recommendation accuracy. With similar recommendation utility, models with UC-FedRec better protect private attributes than early-stopping. Besides, UC-FedRec provides privacy protection to both FedGNN and FedNCF, and its performance is related to the base model. UC-FedRec with FedGNN as its base model can perform better in recommendation utility while facing relatively severe privacy leakage than FedNCF.

4.3 Model Effectiveness

We also validate our method's effectiveness in privacy preservation. To the best of our knowledge, there are no techniques that can provide personalized and compositional privacy protection in federated learning, therefore, we only select 2 types of data privacy-preserving baselines [29] and compare our framework's

overall protection performance on all attributes with them for a fair comparison. A detailed introduction of these baselines is listed below:

Noise perturbation: We train the base FedRec and apply Gaussian noise $\mathcal{N}(0, \sigma^2)$ to trained private user embeddings in the server. The injected noise can provide strong privacy guarantees that avoid the model from privacy leakage. The tradeoff can be influenced by noise strength σ.

LDP: LDP is widely used to protect privacy, user injects Laplace noise to the gradients on the client side, and the privacy budget can be adjusted by the noise strength λ, therefore, we inject stronger noise to private user's gradients to provide better protection.

We compare the utility privacy tradeoff in our framework with the other two baselines. In our experiments, we provide three feature protection choices in MovieLens dataset, which are gender, age, and occupation. As the two baselines cannot provide personalized information preservation, we assume that there are 30% users who want to hide all their attributes and other users do not mind revealing some personal attribute information to have the best recommendation experience for a fair comparison. For our
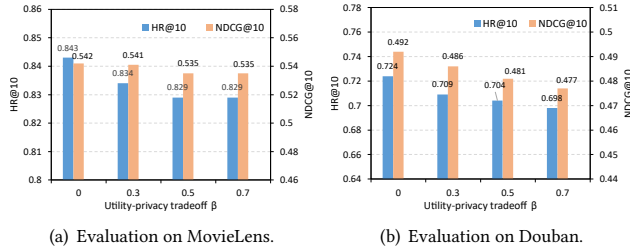
(a) Evaluation on MovieLens.

(b) Evaluation on Douban.

Figure 4: Utility evaluation on non-private users.

framework, we set different β for various utility-privacy balances. Here we set $\beta = \{0.3, 0.5, 0.7, 0.8\}$, respectively. For the noise perturbation baseline, we add different strength Gaussian noise to learned private user embeddings. Here we set $\sigma = \{1, 3, 5, 7\}$, respectively. In the LDP method, a smaller privacy budget provides stronger privacy guarantees for users. We differently set private users' smaller privacy budgets to protect their privacy. Here we set $\epsilon = \{0.1, 0.3, 0.5, 0.7\}$, respectively. We evaluate and compare private users' recommendation utility and privacy protection. As shown in Figure 3, it can be observed that privacy protection leads to a degradation in utility in all the techniques. However, our framework has a better utility-privacy tradeoff compared to the other two baselines. Utilizing the different user privacy preferences, the framework can eliminate the privacy information more accurately. Taking gender protection as an example, in Figure 3(a) and 3(d), the AUC of Gender inference attack on the base FedRec is 0.817 and decreases to 0.5 when the utility-privacy tradeoff β increases to 0.8. When our framework can eliminate all the private users' gender information (AUC = 0.5), it can still remain 0.75 HR and 0.457 NDCG which is 89.0% and 84.0% of the base model performance respectively. While LDP and noisy perturbation can only get about 72% and 50% original recommendation performance.

4.4 Privacy Influence

*4.4.1 **Influence on non-private users.*** Users have different privacy preferences and some users care more about their recommendation experience rather than personal privacy, therefore the recommendation utility for them cannot be largely influenced. We conduct experiments to evaluate our framework's influence on non-private users. For every user, we randomly sample private attributes with a probability of 30%, and evaluate the recommendation performance for those users with no private attributes.

Figure 4 shows the recommendation performance for non-private users in MovieLens and Douban datasets. It indicates that our framework can well retain recommendation performance for non-private users. These users do not need to pay much utility cost for unnecessary privacy. For example, when $\beta = 0.7$, the HR and NDCG can still reach 0.698 and 0.477 which are 96.4% and 97.0% for non-private users compared to the base model in Douban dataset respectively. The recommendation performance still degrades due to the framework will eliminate the sensitive information in the whole system including collaborative signals.

*4.4.2 **Personalized privacy preference.*** Besides users' different privacy and utility preferences, they usually have various privacy
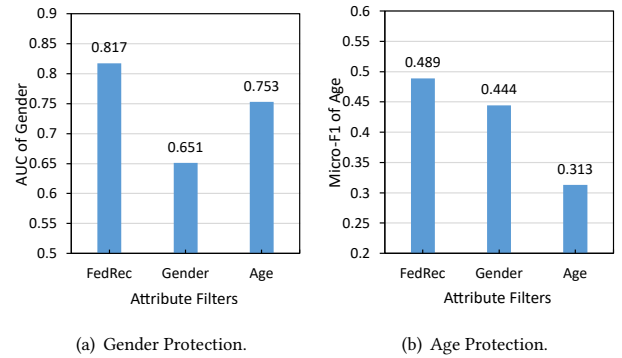

(a) Gender Protection.

(b) Age Protection.

Figure 5: The personalized protection of private attributes.

(a) HR@10 evaluation.
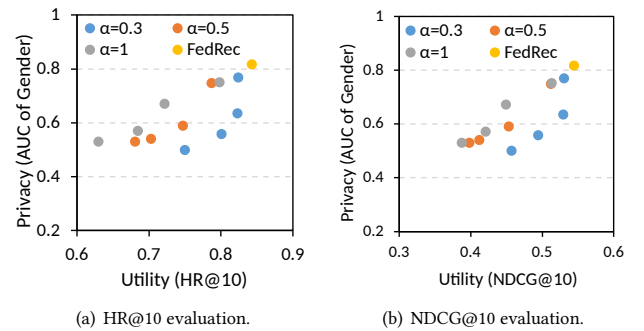
(b) NDCG@10 evaluation.

Figure 6: Comparison of different private user ratios.

preferences. For example, some users may find gender is more sensitive compared to age while others do not. We conduct experiments on single attribute protection to validate personalized privacy protection. In MovieLens, we randomly assign gender or age privacy protection for every user and infer their personal attributes. As shown in Figure 5, both gender and age feature filters can provide privacy protection for either gender and age compared to the base model. For example, the gender filter eliminates some age information so that the attacker's inference micro-F1 drops from 0.489 to 0.444. However, the age filter can protect users' age information more accurately where the attacker can only reach 0.313 mirco-F1, which means that our specialized filters can provide more precise protection to meet users' different personalized privacy preferences.

*4.4.3 **Private attribute amounts.*** In this part, we evaluate the impact of the number of private attributes. We divide users into different groups according to their private attribute amount, then evaluate the recommendation performance of each group. As shown in Figure 7, it can be observed that the recommendation utility degrades when the private attributes increase, which means that users pay more recommendation costs when they need to protect more attributes. However, it still remains high recommendation accuracy when all the attributes are protected. For example in MovieLens, users can have 0.773 HR and 0.479 NDCG which are 91.7% and 88.3% compared to non-private users respectively, it indicates that UC-FedRec can retain most of the collaborative filtering information and satisfy users' privacy needs with acceptable cost.
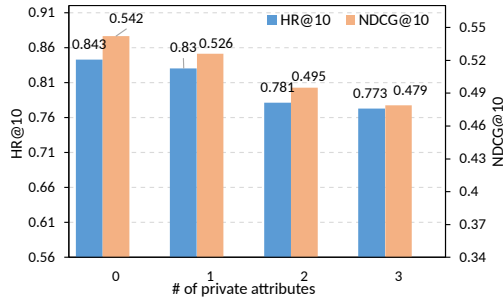
Figure 7: Recommendation utility evaluation on the number of private attributes.

*4.4.4 **Private user ratio.*** As our framework utilizes users' different privacy preferences and needs users to share part of their non-private attributes to train the attribute distribution estimators. The ratio of private attributes for the whole recommendation system can influence the estimators' accuracy, thereby influencing the protection effectiveness. Therefore, we evaluate the framework on different private attribute ratios α. We choose privacy ratio $\alpha = \{0.3, 0.5, 1\}$ respectively and evaluate the privacy protection and recommendation utility under different tradeoffs.

Figure 6 shows that UC-FedRec can protect users' privacy under all the private users' ratios. However, if there are more users willing to share their own attributes, the framework can provide a better utility-privacy tradeoff balance for users, because the attribute distribution estimators are well-trained and the private information is eliminated precisely. When the private attribute ratio increases, the framework has to pay more recommendation utility to get the same protective effect. For example, though the attack AUC is similar (AUC = 0.56), the recommendation utility can retain HR = 0.75 when α = 0.3 while HR = 0.63 when α = 1. This is because of the performance drops in sensitive attribute estimators. Besides, even if there are no user-sharing attributes (α = 1), the framework can still provide protection because it adds random noise to the gradients and performs like an LDP protection under this circumstance.

4.5 Central Protection

In the framework, privacy protection can be done not only during the training stage but also for the trained model. As introduced in Section 3.5.2, when users' privacy preferences change, the framework can directly use trained filters to provide protection promptly. After the model is trained, we randomly choose 20% users who do not care about specified feature protection and apply central protection to protect the feature for these users. We set the threshold T = 100 and iteratively update the embeddings of these users until the difference reaches the threshold. Table 3 shows the central protection performance on the age attribute in MovieLens and Occupation attribute in Douban. The utility and privacy evaluation comparison indicates that the framework can remove sensitive information on a central server with acceptable recommendation utility costs.

Table 3: Results on central protection.

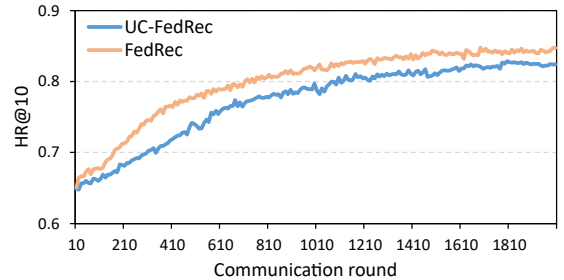|	Protection	Utility		Privacy
		HR	NDCG	Micro-F1
MovieLens	Non-private	0.829	0.535	0.387
(Age)	Private	0.768	0.502	0.291
Douban	Non-private	0.721	0.492	0.304
(Location)	Private	0.698	0.473	0.265

Figure 8: Convergence speed on MovieLens.

4.6 Convergence Rate

In this section, we evaluate whether the framework would influence the convergence rate. Figure 8 shows the framework's recommendation performance in different communication rounds compared to the base model in MovieLens. As shown in the figure, the framework slightly influences the convergence speed, including slower convergence speed and less stable training. However, the difference mainly happens at the beginning of training. It is mainly because the attribute distribution estimators need several rounds to converge. The total communication rounds for the FedRec and our UC-FedRec are similar.

5 CONCLUSION

In this paper, we present a personalized privacy protection framework for FedRec which handles the problem that traditional FedRec suffers inference attacks. Our framework can provide different user-level attribute preservation according to users' various privacy preferences. Experiments on two real-world datasets demonstrate that our framework outperforms the baselines in recommendation utility and privacy protection tradeoff. Our framework can provide flexible and effective privacy protection which does not pay much recommendation accuracy cost. In addition, the framework's convergence speed is similar to the base FedRec. In the future, we plan to extend our model to non-adversarial techniques and unsupervised learning scenarios to protect unseen attributes.

REFERENCES

[1] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888* (2019).

[2] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10, 3 (2015), 137–150.

[3] Raghavendran Balu and Teddy Furon. 2016. Differentially private matrix factorization using sketching techniques. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security.* 57–62.

[4] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-aware recommendation with private-attribute protection using adversarial learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining.* 34–42.

[5] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[6] Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning.* PMLR, 715–724.

[7] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. 2011. " You might also like:" Privacy risks of collaborative filtering. In *2011 IEEE Symposium on Security and Privacy.* IEEE, 231–246.

[8] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2020. Secure federated matrix factorization. *IEEE Intelligent Systems* 36, 5 (2020), 11–20.

[9] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[10] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 27–34.

[11] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning.* PMLR, 1779–1788.

[12] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[13] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. 2018. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data.* 1655–1658.

[14] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.

[15] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 515–524.

[16] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web.* 278–288.

[17] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference.* 417–426.

[18] Neil Zhenqiang Gong and Bin Liu. 2016. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *25th USENIX Security Symposium (USENIX Security 16).* 979–995.

[19] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems (TIIS)* 5, 4 (2015), 1–19.

[20] Gaole He, Junyi Li, Wayne Xin Zhao, Peiju Liu, and Ji-Rong Wen. 2020. Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning. In *Proceedings of The Web Conference 2020.* 740–751.

[21] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web.* 173–182.

[22] Hui Hu, Lu Cheng, Jayden Parker Vap, and Mike Borowczak. 2022. Learning privacy-preserving graph convolutional network with partially observed sensitive attributes. In *Proceedings of the ACM Web Conference 2022.* 3552–3561.

[23] Qi Hu and Yangqiu Song. 2023. Independent Distribution Regularization for Private Graph Embedding. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management.* 823–832.

[24] Jinyuan Jia and Neil Zhenqiang Gong. 2018. {AttriGuard}: A practical defense against attribute inference attacks via adversarial machine learning. In *27th USENIX Security Symposium (USENIX Security 18).* 513–529.

[25] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 659–667.

[26] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[28] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[29] Sicong Liu, Junzhao Du, Anshumali Shrivastava, and Lin Zhong. 2019. Privacy adversarial network: representation learning for mobile data privacy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–18.

[30] Zhifeng Luo and Zhanli Chen. 2014. A privacy preserving group recommender based on cooperative perturbation. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.* IEEE, 106–111.

[31] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining.* 287–296.

[32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics.* PMLR, 1273–1282.

[33] Xuying Meng, Suhang Wang, Kai Shu, Jundong Li, Bo Chen, Huan Liu, and Yujun Zhang. 2018. Personalized privacy-preserving social recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[34] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. 2021. Stronger Privacy for Federated Collaborative Filtering With Implicit Feedback. In *Fifteenth ACM Conference on Recommender Systems.* 342–350.

[35] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. Fedfast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1234–1242.

[36] Vasileios Perifanis and Pavlos S Efraimidis. 2022. Federated Neural Collaborative Filtering. *Knowledge-Based Systems* 242 (2022), 108441.

[37] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-Preserving News Recommendation Model Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020.* 1423–1432.

[38] Charles D Raab. 1998. The distribution of privacy risks: Who needs protection? *The information society* 14, 4 (1998), 263–274.

[39] Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A McCann, and S Yu Philip. 2018. LoPub: high-dimensional crowdsourced data publication with local differential privacy. *IEEE Transactions on Information Forensics and Security* 13, 9 (2018), 2151–2166.

[40] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[41] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1770–1782.

[42] Binghui Wang, Jiayi Guo, Ang Li, Yiran Chen, and Hai Li. 2021. Privacy-Preserving Representation Learning on Graphs: A Mutual Information Perspective. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 1667–1676.

[43] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 950–958.

[44] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval.* 165–174.

[45] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications.* IEEE, 2512–2520.

[46] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. 2021. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925* (2021).

[47] Shiwen Wu, Fei Sun, Wentao Zhang, and Bin Cui. 2020. Graph neural networks in recommender systems: a survey. *arXiv preprint arXiv:2011.02260* (2020).

[48] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval.* 125–134.

[49] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17.* 3203–3209.

[50] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[51] Bo Zhang, Na Wang, and Hongxia Jin. 2014. Privacy concerns in online recommender systems: influences of control and user data input. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*. 159–173.

[52] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership Inference Attacks Against Recommender Systems. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 864–879.

[53] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[54] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Lizhen Cui, and Xiangliang Zhang. 2021. Graph embedding for recommendation against attribute inference attacks. In *Proceedings of the Web Conference 2021*. 3002–3014.

ETHICAL CONSIDERATIONS

UC-FedRec frameworks can effectively protect users' privacy from attribute inference attacks if the technology is being used as intended. However, we do not consider the situation that the FedRec server is malicious where some shared information may face greater risks. The regulations of service providers are needed and technical solutions remain future works.