Comprehensive Privacy Analysis on Federated Recommender System against Attribute Inference Attacks

Shijie Zhang and Hongzhi Yin, *Senior Member, IEEE*

Abstract—In recent years, recommender systems are crucially important for the delivery of personalized services that satisfy users' preferences. With personalized recommendation services, users can enjoy a variety of recommendations such as movies, books, ads, restaurants, and more. Despite the great benefits, personalized recommendations typically require the collection of personal data for user modelling and analysis, which can make users susceptible to attribute inference attacks. Specifically, the vulnerability of existing centralized recommenders under attribute inference attacks leaves malicious attackers a backdoor to infer users' private attributes, as the systems remember information of their training data (i.e., interaction data and side information). An emerging practice is to implement recommender systems in the federated setting, which enables all user devices to collaboratively learn a shared global recommender while keeping all the training data on device. However, the privacy issues in federated recommender systems have been rarely explored. In this paper, we first design a novel attribute inference attacker to perform a comprehensive privacy analysis of the state-of-the-art federated recommender models. The experimental results show that the vulnerability of each model component against attribute inference attack is varied, highlighting the need for new defense approaches. Therefore, we propose a novel adaptive privacy-preserving approach to protect users' sensitive data in the presence of attribute inference attacks and meanwhile maximize the recommendation accuracy. Extensive experimental results on two real-world datasets validate the superior performance of our model on both recommendation effectiveness and resistance to inference attacks.

Index Terms—Recommender System, Federated Learning, Local Differential Privacy, Attribute Inference Attack

✦

1 INTRODUCTION

In online services, the demand for recommender systems has increased more than ever before, due to their success in alleviating the problem of information overload by filtering vital information out of a large volume of data to efficiently deliver personalized contents and services for users [1], [2]. It is no wonder, these recommenders, if set up and configured properly, can significantly contribute to revenues as well as user experience. In recent years, various recommendation algorithms have been proposed and achieved immense success in practical applications. Collaborative filtering (CF) based recommender systems, which make recommendations by utilizing users' historical interaction data, are widely deployed in the online platforms, for the fact that they are effective and efficient. More recently, deep learning-based recommender systems have demonstrated advantageous effectiveness by advancing the representation learning capability and producing high-quality recommendations [3], [4], [5].

Despite the promising effectiveness, tremendous privacy concerns on recommender systems are raised in recent years. On one hand, to boost the recommendation performance, especially for fresh (i.e., cold-start) customers, these systems hungrily collect various side information data (a.k.a. user attributes or contexts) to better infer users' preferences [6], [7]. When registering accounts, users are
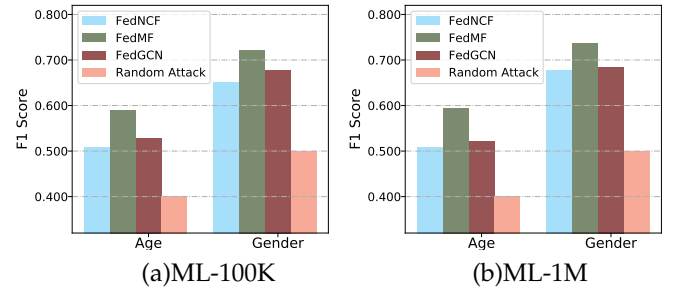

Fig. 1: Attribute inference attack results of popular federated recommenders on ML-100K and ML-1M.

required to complete questionnaires about their personal demographics (i.e., age and gender) to facilitate user profiling. Once intercepted by malicious third parties, the leakage of users' sensitive information can be catastrophic. On the other hand, recent research indicates that even users' unpublished private information can be inferred via their interaction history with high confidence [8], [9], [10]. Such personal information includes but not limited to age, gender, political orientation, health, financial status etc. Much worse, the attackers can utilize the inferred attributes to link user accounts across multiple platforms and break anonymity [11], [12]. Such attack is termed as attribute inference attack [13], where the malicious attackers can be data brokers, cyber criminals, advertisers, etc. An example is that [14] successfully deanonymizes Netflix users by utilization of the publicly accessible IMDb user profiles. Accordingly, the personalized recommendation results can

• *S. Zhang and H. Yin are with the school of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia, E-mail: shijie.zhang@uq.edu.au, h.yin1@uq.edu.au*

Hongzhi Yin is the corresponding author.

also cause privacy leakage since they imply users' interests and even their sensitive attributes [15], [16]. In previous work [15], a novel differentially private graph convolutional network named GERAI is proposed to address aforementioned privacy issues. Specifically, the graph convolutional networks (GCNs) [17], [18], [19] is adopted as the main building block, since it is capable of jointly exploiting the user-item interactions and the rich side information of users. Then, to provide a privacy guarantee, the authors design a novel dual-stage perturbation paradigm with differential privacy, which makes the recommendations less dependent on users' sensitive data, avoiding privacy leakage in recommendation results. Unfortunately, despite its success, this centralized recommendation paradigm still inevitably leads to increasing risks to user privacy, since user data stored on the central server might be accidentally leaked or misused.

In light of privacy issues in centralized recommender systems, there has been a recent surge in decentralized recommender systems, where federated learning [20], [21], [22] becomes one of the most representative frameworks in the development of privacy-preserving systems. Specifically, a federated recommender shown in Figure 2 allows users' data safely hosted on their personal devices, and the shared global recommender is collaboratively trained in a multi-round fashion by collecting a batch of locally updated models to the central server for parameters update. To avoid privacy concerns, the server is designed to have no access of each client's local data and training process. As a result, such federated recommenders seem to be regarded as 'safe' towards attribute inference attacks. Though federated learning framework can achieve comparable recommendation results without sharing users' sensitive data, recent works show that it is yet to provide a privacy guarantee of training data [23]. Specifically, model parameters uploaded during the training process provide a chance for inference attacks, since the well-trained parameters can remember local data information. [24], [25], [26] have studied that deep learning models in the federated setting are susceptible to membership inference attacks, where the attacker is able to infer the samples used to train the model.

Although membership inference attacks have been studied in the federated setting [24], [25], [26], the vulnerability of federated recommender against attribute inference attacks (i.e., users' attributes) remains unexplored. To validate whether federated recommender paradigm is susceptible to attribute inference attacks, we make the first attempt to infer users' private attributes through the uploaded parameters in the federated setting. Figure 1 demonstrates the F1 Score achieved by the attribute inference attacker (refer to details in Section 3.3) on three well-trained federated recommenders (MF [27], NCF [28] and GCN [19]). To show the significance of attribute inference from local model parameters, we build a random guessing classifier named Random Attack as a null model. From the results shown in Figure 1, we observe significant differences between the random attack and the inference attack based on local model parameters. It can be concluded that the shared model parameters in the federated learning process significantly reveal users' attribute information, demonstrating the demand for advanced defenses against attribute inference attacks to federated recommender systems. Recently, local

differential privacy (LDP) has become a gold standard for providing protection guarantee of local model parameters in the federated setting. [20], [29], [30], [31] successfully apply LDP mechanism in federated deep learning tasks to transform the local model parameters into a noisy version at each user device before being uploaded to the central server. Despite its success in many applications against membership attacks, it has been rarely studied to protect users' attribute information in the federated recommendation.

These aforementioned limitations motivate us to propose a novel privacy-aware federated recommender system that significantly improve both recommendation effectiveness and robustness against attribute inference attacks. However, how to apply LDP technique in the federated recommender systems faces tremendous challenges. Setting an appropriate value for the privacy budget is crucial for the utility of the attack-resistance recommender system in the federated setting. A low budget value (i.e., noise factor) can result in a high success rate of attribute inference attack since the noise added into model parameters are negligible and ineffective, while a high value will inevitably destroy model utility. For simplicity, existing works [20], [29] just fix a constant DP budget for all model parameters. However, designed for modelling nonlinear relations between users and items, federated deep recommender models have multiple components and layers, and their model parameters exhibit large variance, thus the vulnerability of each component or even each layer to the attribute inference attacks is different. Therefore, treating all components/layers under the same privacy protection mechanism results in unavoidable excessive utility loss.

In this paper, we perform a comprehensive privacy analysis of each component of the SOTA GCN-based federated recommender model, and the results show varied vulnerabilities of these components against attribute inference attacks. Specifically, we divide model parameters into three main parts based on functionality, namely User Component, Item Component and MLP. To achieve optimal privacy strength without sacrificing much recommendation accuracy, we design a novel adaptive LDP mechanism that can automatically adjust the utility loss of each component when defending attribute inference attacks. To conclude, we highlight our main contributions as follows:

- To the best of our knowledge, we are the first to present a comprehensive privacy analysis of federated recommender systems under attribute inference attacks. Our study reveals that well-trained recommenders are significantly susceptible to attribute inference attacks even in a decentralized environment.
- In order to address growing privacy concerns in the federated recommendation context, we design a novel adaptive privacy-aware mechanism to guard users' sensitive data against attribute inference attacks without sacrificing high-quality recommendation results. Our model innovatively takes advantage of the inherent informative bias to reduce overall utility loss in defending attribute inference attacks.
- Extensive experiments are conducted on two real-world datasets, and the results demonstrate the superior performance of our solution. Furthermore, compared with all baselines, the results show that our model provides
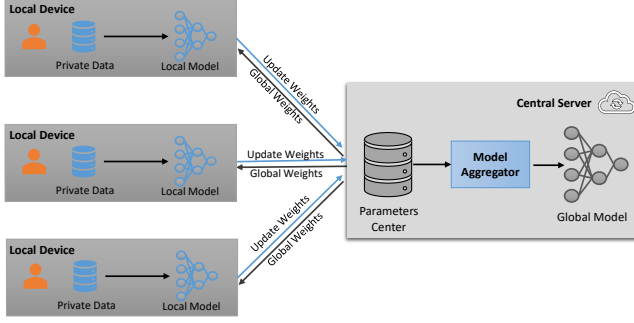
Fig. 2: Architecture of a typical Federated Recommender System.

a strong privacy guarantee with less compromise on the recommendation accuracy.

2 PRELIMINARIES

In this section, we first revisit the key definitions that are frequently used in this paper and then formulate our problem. Note that we use bold lowercase (e.g., $\mathbf{a}$) to denote vectors, and use bold uppercase (e.g., $\mathbf{A}$) to denote matrices. All sets are written in squiggle uppercase letters (e.g., $\mathcal{A}$).

Differential Privacy. Differential privacy (DP) can provide a rigorously mathematical guarantee in the machine learning tasks. The notion of differential privacy was first introduced by [32] and it can be utilized to defend against malicious attackers that infer useful information from a target model (e.g., outputs and structure of the model). Given a privacy parameter $\epsilon > 0$, the $\epsilon-$differential privacy ($\epsilon-$DP) is defined as follows:

Definition 2.1. ($\epsilon-$Differential Privacy) A randomized mechanism $f : \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$, and it satisfies $\epsilon-$DP if:

$$Pr[f(\mathcal{D}) \in O] \leq exp(\epsilon)Pr[f(\mathcal{D}') \in O], \quad (1)$$

where $Pr[\cdot]$ represents probability, $\mathcal{D}$ and $\mathcal{D}'$ are two adjacent datasets differing on only one data instance, and $O \subseteq \mathcal{R}$ denotes any subsets of possible output values. Eq.(1) implies that the probability of output distribution $f(\mathcal{D})$ is at most $exp(\epsilon)$ times smaller than that of $f(\mathcal{D}')$. On this basis, $f(\cdot)$ is not overly dependent on any individual data record, providing each instance roughly the same privacy. In the federated setting, the central server updates global model by aggregating the collected model parameters from a group of local devices. To defend against malicious attacks that infer user's private attributes via its device's updates, each local device should first perturb model parameters Θ by directly adding noise, and then the noised version of Θ^* is updated to the central server instead of original parameters. Hence, the model parameter generated by each local device is treated as a singleton dataset, and we require the random perturbation mechanism $f(\cdot)$ to perform by the local devices, not by the central server. Specifically, we introduce $\epsilon-$local differential privacy ($\epsilon-$LDP) which is a special case of differential privacy:

Definition 2.1. ($\epsilon-$Local Differential Privacy) A randomized mechanism $f(\cdot)$ satisfies $\epsilon-$LDP if and only if for any two input data Θ and Θ', we have:

$$Pr[f(\Theta) = \Theta^*] \leq exp(\epsilon) \cdot Pr[f(\Theta') = \Theta^*] \quad (2)$$

where Θ^* denotes the output of $f(\cdot)$. The lower ϵ provides stronger privacy but may result in severe performance drop of a federated model as each local model is heavily perturbed. Hence, ϵ determines privacy budget that controls the trade-off between privacy and model utility. With the privacy guarantee from LDP, an external attacker who collects Θ^* (e.g., perturbed model parameters) cannot accurately estimate the true data is Θ or Θ', and thus the sensitive information is obfuscated.

Federated Recommender Systems. Let $\mathcal{V}$ and $\mathcal{U}$ denote the sets of N items and M users, respectively. Each device used by an individual user $u \in \mathcal{U}$ has a local training dataset $\mathcal{D}_u$ that consists of implicit feedback tuples (u, v, r_{uv}), where $r_{uv} = 1$ if u has interacted with item $v \in \mathcal{V}$ (i.e., a positive instance), otherwise, we set r_{uv} to 0 (i.e., a negative instance). Due to the large number of unobserved interactions, we use a sample ratio of $1 : q$ to downsample the negative instances for each user u. We use $\mathcal{N}(u)$ to denote the set of items visited by u. Additionally, each user u privately preserves a dense input vector $\mathbf{x}_u \in \mathbb{R}^{d^1}$ in which each element represents either u's private attribute $s \in \mathcal{S}$ or a extracted statistical feature $s \in \mathcal{S}'$ based on $\mathcal{D}_u$. Not that each categorical feature (i.e., age and gender in our case) is represented by one-hot encoding in $\mathbf{x}_u$. The federated recommender system aims to train a global recommender across multiple decentralized user devices that hold local private data (i.e., $\mathbf{x}_u$ and $\mathcal{D}_u$), without direct access to them:

$$\underset{\Theta_u}{\operatorname{argmin}} \sum_{u \in \mathcal{U}} \mathcal{L}^{rec}(\mathcal{D}_u, \Theta_u) \quad (3)$$

mwhere $\mathcal{L}^{rec}(\cdot)$ is a loss function and Θ_u represents all the trainable parameters of u's local recommender. For notation convenience, we use Θ to represent the recommender system.

Task 1. For each user $u \in \mathcal{U}$, given its local dataset $\mathcal{D}_u$ and user feature vector $\mathbf{x}_u$, we aim to learn a privacy-preserving federated recommender system, in which malicious attackers are unable to infer user's private attributes (i.e., gender and age in our case) via u's uploaded model parameters with high confidence.

3 ATTRIBUTE INFERENCE ATTACKS

3.1 Base Recommender

Federated learning appears to be compatible with various latent factor models. The advantage of FL that users' data host on their local devices makes it an attractive alternative for developing privacy-preserving machine learning models. In this paper, we extend a GCN-based centralized model [15], [19] to a federated one, which is named FedRec. The local FedRec framework is shown in Figure 3, which consists of three key components to generate recommendations, namely user component, item component and MLP. The only difference between user component and item component is that the GCN layer is adopted as the key building block to learn user embeddings, because it
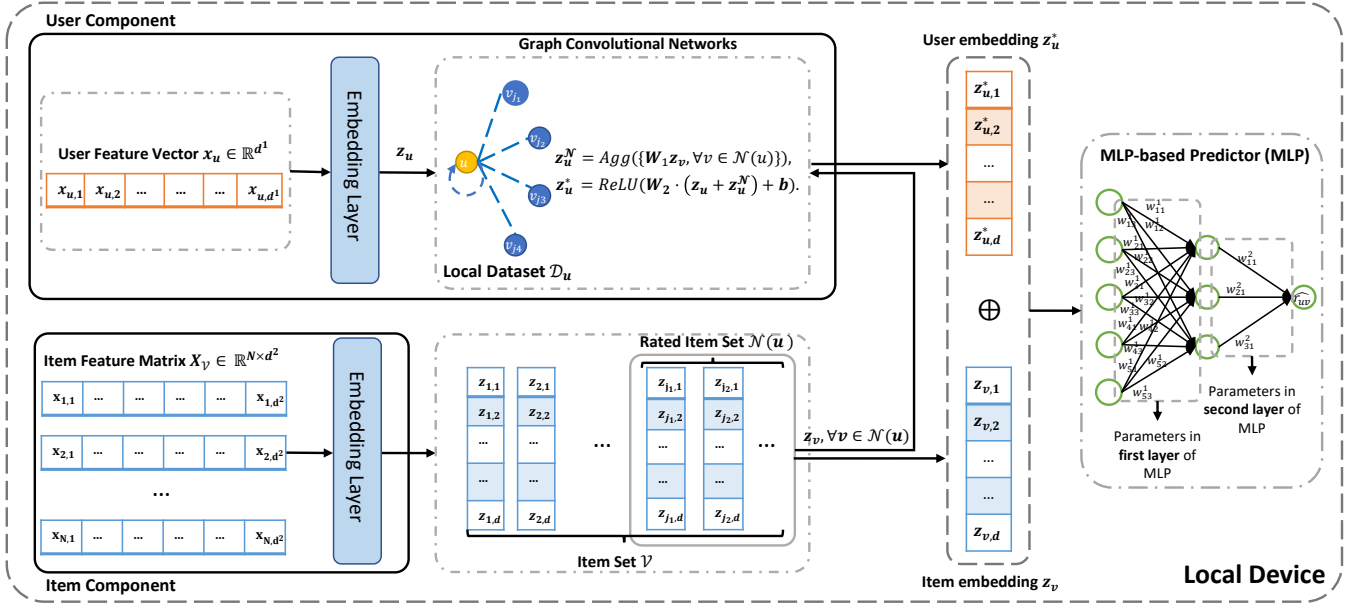
Fig. 3: Overview of Local model in each device.

is advantageous in capturing local structure information of the user-item interaction data and user's side information in a unified way. To guarantee user privacy, the neighbor set $\mathcal{N}(v)$ of each item v is highly sensitive and restricted, and thus GCN layer cannot be applied to item embeddings. In what follows, we will introduce the design details of each component in FedRec.

In user u's local device, given local dataset $\mathcal{D}_u$ and we assume that a feature vector $\mathbf{x}_u$ associated with u is available. Note that we use $\mathbf{z}_u$ and $\mathbf{z}_v$ to denote the latent representations of users and items in the same latent space, respectively. Specifically, $\mathbf{z}_u, \mathbf{z}_v$ can be initialized as follows:

$$\mathbf{z}_u = \mathbf{E}_{\mathcal{U}}\mathbf{x}_u, \ \mathbf{z}_v = \mathbf{E}_{\mathcal{V}}\mathbf{x}_v, \tag{4}$$

where $\mathbf{x}_u \in \mathbb{R}^{d^1}$ is user u's raw feature vector and $\mathbf{E}_{\mathcal{U}} \in \mathbb{R}^{d \times d^1}$ is the user feature transformation matrix. $\mathbf{x}_v \in \mathbb{R}^{d^2}$ and $\mathbf{E}_{\mathcal{V}} \in \mathbb{R}^{d \times d^2}$ represent item feature vector and item feature transformation matrix. To ensure our model's generalizability, each item feature vector $\mathbf{x}_v$ is initialized using randomized values as we do not assume the availability of item features.

In each forward iteration, to learn the embedding of user u, GCN firstly computes its embedding by iteratively aggregating information from its first-order neighbors, i.e., items interacted with u:

$$\mathbf{z}_u^{\mathcal{N}} = Agg(\{\mathbf{W}_1\mathbf{z}_v, \forall v \in \mathcal{N}(u)\}), \tag{5}$$

where $\mathbf{W}_1$ is trainable weight matrix and $Agg(\cdot)$ is the aggregation function which aggregates the neighborhood information $\mathbf{z}_v$ into a unified vector representation. In the experiment, we adopt the average aggregation function for simplicity.

Then, the user's current representation $\mathbf{z}_u$ is added to the aggregated neighborhood vector $\mathbf{z}_u^{\mathcal{N}}$, and then being fed through a fully connected layer to form an updated user embedding $\mathbf{z}_u^*$:

$$\mathbf{z}_u^* = ReLU(\mathbf{W}_2 \cdot (\mathbf{z}_u + \mathbf{z}_u^{\mathcal{N}}) + \mathbf{b}), \tag{6}$$

where $ReLU(\cdot)$ denotes the rectified linear unit for nonlinearity, and $\mathbf{W}_2$ and $\mathbf{b}$ are trainable weight matrix and bias vector.

On the item side, as the neighbor set of each item (i.e., the set of users who interact with the item) is unavailable in the federated setting, we do not perform convolution operation when learning item representations. Therefore, the parameters in user component and item component can be represented as $\Theta^{\mathcal{U}} = \{\mathbf{E}_{\mathcal{U}}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{b}\}$ and $\Theta^{\mathcal{V}} = \{\mathbf{E}_{\mathcal{V}}\}$, respectively.

Afterwards, in user u's local device, the ranking score $\hat{r}_{uv}$ for an arbitrary item $v \in \mathcal{N}(u)$ can be predicted by u's local recommender. To achieve this, local recommender first concatenate u's and target item v's current embeddings, and then adopt a L-layer perceptron network $MLP(\cdot)$ to model the user-level interactions and estimate $\hat{r}_{uv}$:

$$\hat{r}_{uv} = \sigma(MLP(\mathbf{z}_u^* \oplus \mathbf{z}_v)), \tag{7}$$

where $\mathbf{z}_u^*, \mathbf{z}_v \in \mathbb{R}^d$ are user and item embeddings respectively, $\oplus$ is the concatenation operation, and σ is the sigmoid function that rectifies the ranking score to the range $[0, 1]$.

3.2 Federated Learning Protocol.

As shown in Algorithm 1, FedRec aims to train a shared recommender with a central server by coordinating individual user devices to train local models based on private dataset $\mathcal{D}_u$ and user attribute information $\mathbf{x}_u$. To train FedRec, each user's local model is optimized locally with a distance-based loss function:

$$\mathcal{L}^{rec} = -\sum_{(u,v,r_{uv}) \in \mathcal{D}_u} r_{uv} \log \hat{r}_{uv} + (1 - r_{uv}) \log(1 - \hat{r}_{uv}), \tag{8}$$

where $\hat{r}_{uv} \in [0, 1]$ is obtained via Eq (7). Notably, we adopt cross-entropy loss to minimize the difference between $\hat{r}_{uv}$ and the ground truth r_{uv}. With the user-specific loss $\mathcal{L}^{rec}$ computed, we can obtain updated parameters Θ_u^t of u's local model. Specifically, at each epoch t, a subset

---

Algorithm 1: Procedures for Training FedRec

Input: The number of global rounds $\mathcal{T}_g$ and local
Epochs $\mathcal{T}_l$, sampled clients $\mathcal{U}^t$ at each time t,
initialized model parameters Θ and local
learning rates μ_1;

for $t \leq \mathcal{T}_g$ **do**
  Local Training Process for $u \in \mathcal{U}^t$:
  Initialization: $\Theta_u^t \leftarrow \Theta$;
  for $t_2 \leq \mathcal{T}_l$ **do**
    Draw a minibatch $\mathcal{B}$;
    for $\mathcal{B}_i \in \mathcal{B}$ **do**
      $\Theta_u^t \leftarrow \Theta_u^t - \mu_1 \frac{\partial \mathcal{L}^{rec}(\mathcal{B}_i, \Theta_u^t)}{\partial \Theta_u^t}$;
    end
    $t_2 \leftarrow t_2 + 1$;
  end
  Send Θ_u^t to Server;
  Model Aggregating process:
  Update the global parameters Θ as:
  $\Theta^{t+1} = \frac{1}{|\mathcal{U}^t|} \sum_{u \in \mathcal{U}^t} \Theta_u^t$;
  $\Theta \leftarrow \Theta^{t+1}$;
  $t \leftarrow t + 1$;
end

---

of users $\mathcal{U}^t$ are randomly drawn, and each selected local device should download the latest global recommender Θ_t and then update its local recommender on $\mathcal{D}_u$. Then, each device uploads its updated local model parameters to the central server. Once the central server receives all local model parameters submitted by $|\mathcal{U}_t|$ users, it aggregates the collected parameters to facilitate global recommender update. Specifically, FedRec follows the commonly used FedAvg protocol [33] to update global recommender Θ_{t+1}:

$$\Theta^{t+1} = \sum_{u \in \mathcal{U}_t} \Theta_u^t. \qquad (9)$$

The training continues iteratively until the global recommender achieves convergence criteria. Unlike centralized recommenders, FedRec collects only each user's local model parameters instead of personal data (i.e., $\mathcal{D}_u$ and demographics $\mathbf{x}_u$), and local model parameters are not directly shared across users.

3.3 Attribute Inference Attack

Generally, federated recommender system assumes that the central server is trusted and works under non-adversarial setting. In real-life scenarios, the server is honest-but-curious, which means server is curious in inferring the information of individuals via received updates but honest in processing the updates, leaving a backdoor to estimate users' private attributes. To perform attribute inference attacks on FedeRec, traditional attack approaches designed for centralized recommenders have limited success to attack federated models. The main reason is that the prior knowledge (e.g., all user-item interactions and recommendation results) that is requisite for the malicious attackers [15], [16], cannot be obtained in the federated setting as such information is kept privately at the user side. Furthermore, the federated setting substantially restricts the knowledge

acquisition of an attribute inference attack model, and we summarized the accessible prior knowledge as follows:

I. The adversary can access user u's local model parameters Θ_u^t at any iteration t.
II. The adversary knows sensitive attributes of a small group of users $\mathcal{U}_{adv}$ who may cooperate with the untrusted server due to financial incentives, or general registers who are willingness to share their information with the online platforms.

With the updated model weights Θ^t, we can obtain $\Delta \Theta = \frac{\Theta^{t-1} - \Theta^t}{\mu_1}$, which can be regarded as the gradient of model parameters over one epoch of SGD optimization method. $\Delta \Theta^t$ reflects how much each parameter has to change and contains sufficient information of the training dataset. As a result, the adversary has a dataset $\mathcal{D}_{adv} = \{(\Delta \Theta_u^t, y_u) | \forall u \in \mathcal{U}_{adv}\}$. To construct an attribute inference attack model, the adversary needs to find the meaningful mapping between the model's updated parameters and user attributes. Given the dataset $\mathcal{D}_{adv}$, the most straightforward way of learning such relationship is to train the attack model in a supervised way, and use it to attack the rest users whose attributes are unshared. Then, suppose there are C classes of target attribute, the attribute inference attacker $f_{adv}(\cdot)$ is a three-layer deep neural network that inputs a model update gradient $\Delta \Theta_u$, and outputs a C-dimensional vector $\hat{\mathbf{y}}$ in which each element $\hat{\mathbf{y}}[c] \in \hat{\mathbf{y}}$ ($c = 1, 2, ..., C$) denotes the probability that user u is classified to label c. We train $f_{adv}(\cdot)$ with cross-entropy loss on all training dataset $\mathcal{D}_{adv}$:

$$\mathcal{L}_{adv} = - \sum_{\forall u \in \mathcal{U}_{adv}} \sum_{c=1}^{C} \mathbf{y}[c] \log \hat{\mathbf{y}}[c], \qquad (10)$$

where $\mathbf{y} = \{0, 1\}^C$ is the one-hot label and the hidden dimension is set to 100 and 30. To evaluate FedRec's robustness against the proposed attack model, we randomly choose ζ of users in $\mathcal{U}^t$ as $\mathcal{U}_{adv}$ to train our attacker, and the remainder is utilized to evaluate the inference accuracy. To quantify target recommender's resistance ability, we leverage a widely-used classification metric *F1 Score* to measure the performance of the attacker and show results in Figure 4. Note that we set user/item dimension d to 64 and use a 2-layer MLP in Eq (7), and report inference accuracy when train-test split ratio ζ is 10% and 20% respectively. Correspondingly, lower F1 Score demonstrates higher resistance to this attribute inference attack.

Firstly, we take the whole $\Delta \Theta$ (i.e., Full Version) as the input of the proposed attacker model by simply flattening all parameters from different components of a well-trained FedRec. In Figure 4, we can see that the FedRec that is based on the state-of-the-art GCN-based recommender are significantly vulnerable to our proposed attribute inference attack. Specifically, the inference attack accuracy achieves 0.528 on age attribute and 0.677 on gender attribute on ML-100K, while 0.522 on age attribute and 0.684 on gender attribute on ML-1M, though a small group of users (i.e., $\zeta = 10\%$) are compromised. It confirms that even a fully trained deep recommender in the federated setting can leak significant amount of information about users' sensitive attributes.

Then, to understand and demonstrate the impact of model parameters derived from different components, we

(a) Above: Age and Gender on ML-100K and ML-1M Inference results w.r.t. ζ

(b) Above: Age and Gender on ML-100K and ML-1M Inference results w.r.t. ζ
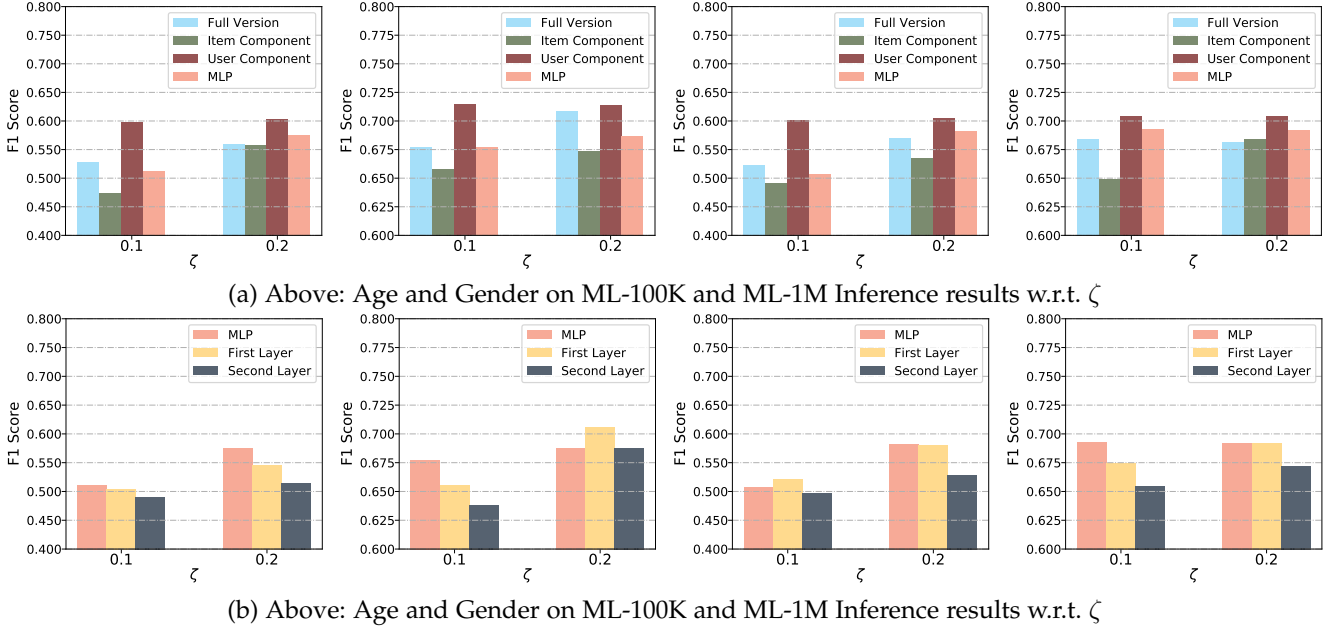
Fig. 4: Attribute inference attack results of FedRec on ML-100k and ML-1M.

compare the inference accuracy of the attacker on each component of the local RedRec separately. As introduced in Section 3.1, the local FedRec is mainly composed of three components, namely user component, item component and MLP. From Figure 4 (a), it is clear that these three components exhibit various degrees of information leakage. User component leaks more attribute information about each user on both two dataset, compared to the other two components. The reason behind this is twofold. By directly processing the user features, the parameters in the user embedding layer inevitably remember much more information of the original user feature, thus leaking more information. Additionally, the GCN layer aggregates the embeddings of the user's interacted items that contain other similar users' attribute information. Furthermore, though MLP does not directly contain user features, it still leaks much user information.

Due to that the recommender accuracy mainly depends on the generalization ability of the MLP, we further perform attacks with individual layers in MLP to study the vulnerability of each layer in the MLP. The results from Figure 4 (b) show that combining all parameters from multiple layers of MLP does not obtain significant accuracy gain. This is because of the information overlap among various layers and high dimension problem that is common in classification task. Notably, the first layer leaks more attribute information, compared to the last layer. One possible reason is that the first layer directly interacts with the concatenated embeddings of user and item. Based on the results, we propose a resistance function $l(\cdot)$ that maps each parameter to the following resistance degree against attribute inference attacks.

$$l(\theta_i) = \begin{cases} 0, & \theta_i \in \Theta^{\mathcal{U}} \\ 1, & \theta_i \in \Theta^{\mathcal{V}} \\ 2, & \theta_i \in \Theta^{MLP_1} \\ 3, & \theta_i \in \Theta^{MLP_2} \end{cases} \quad (11)$$

Therefore, the traditional assumption that the parameters from different components have the same attack vulnerability does not hold in the federated recommender, which motivates us to design an adaptive privacy-preserving federated recommender system to minimize utility loss.

4 ADAPTIVE PRIVACY-PRESERVING MECHANISM

In this section, we present the design of our privacy-preserving mechanism that can defend against attribute inference attacks via an adaptive local differential privacy constraint where each component has a different privacy budget (i.e., noise factor λ_i). Algorithm 2 depicts the workflow of our adaptive privacy-preserving local training in FedRec. As the local model is trained based on a set of user-item interactions and user features, a traditional private training approach works by perturbing the model updates based on LDP techniques with fixed privacy budget for all model components before being submitted to the central server. However, assuming each component exhibits the same vulnerability against attribute inference attack would lead to suboptimal performance in both recommendation accuracy and privacy protection. To this end, we propose an adaptive privacy-preserving training mechanism. Specifically, we allocate a larger share of the privacy budget (larger noise factor) to model parameters with low attack resistances and a smaller share (smaller noise factor) to model parameters with high resistances. Thus, the privacy-preserving FedRec is able to defend against attribute inference attacks with little loss of recommendation accuracy. Despite the success of many DP- and LDP- based approaches [31], [34], [35], [36] on classification task in the federated setting, most of them are not applicable in federated recommender scenarios or result in significant performance drop. Specifically, designed for modelling nonlinear relations between users and items, federated recommenders are optimized towards completely different learning objectives with much more complicated

Algorithm 2: Adaptive Privacy Protection Local Training for FedRec

---

Input: The number of local Epochs $\mathcal{T}_l$, local learning rate μ_1, global model parameters Θ, clipping bond δ, privacy budget function $p(\theta)$;

Output: Perturbed Θ_u^t

Local Training Process:

Initialization: $\Theta_u^t \leftarrow \Theta$;

for $t_2 \leq \mathcal{T}_l$ **do**

    Draw a minibatch $\mathcal{B}$;

    for $\mathcal{B}_i \in \mathcal{B}$ **do**

        $\Theta_u^t \leftarrow \Theta_u^t - \mu_1 \frac{\partial \mathcal{L}^{rec}(\mathcal{B}_i, \Theta_u^t)}{\partial \Theta_u^t}$;

    end

    Add noise:

    $\Theta_u^t \leftarrow clip(\Theta_u^t, \delta)$;

    for $\theta_i^t \in \Theta_u^t$ **do**

        $\lambda_i \leftarrow \frac{2\delta}{p(\theta_i^t)}$;

        $\theta_i^t \leftarrow \theta_i^t + Lap(0, \lambda_i)$;

    end

    $t_2 \leftarrow t_2 + 1$;

end

---

model structures where model parameters exhibit large variance. Particularly, applying strict clipping function applied to high-dimensional model parameters would lead to a large variance of the resulted model parameters and lose information contained in the original parameters. Following [20], [29] that successfully applied Laplace Noise-based LDP approach to federated recommender system, we design the following adaptive privacy-preserving mechanism based on Laplace Noise to achieve optimal privacy protection:

$$\mathcal{M}(\Theta) = clip(\Theta, \delta) + n,$$
$$n \sim Lap(0, \lambda_i), \tag{12}$$

where n is Laplace Noise with 0 mean. The noise factor $\lambda_i = \frac{max_{\theta_i, \theta_i'} |\mathcal{M}(\theta_i) - \mathcal{M}(\theta_i')|}{p(\theta_i)}$ that controls the strength of Laplace Noise is determined by the adaptive privacy parameters ϵ_i, and a larger λ_i can bring better privacy protection. Specifically, given privacy parameter $[\epsilon_{min}, \epsilon_{max}]$ of each parameter and resistance function $l(\cdot)$ in Eq (11), $p(\theta_i) = \epsilon_{min} + b \cdot l(\theta_i)$, where $b = \frac{\epsilon_{max} - \epsilon_{min}}{3}$ controls the privacy budget scoop. The function $clip(\cdot)$ is used to limit the value of each parameter with the scale of δ , and thus noise factor λ_i is limited to $[\frac{2\delta}{\epsilon_{max}}, \frac{2\delta}{\epsilon_{min}}]$. After clip and randomization operation, it is more difficult to infer the raw user side information from the model parameters. Then each user device uploads its randomized local model parameters to the server without privacy leakage.

5 EXPERIMENTs

In this section, we conduct experiments to verify the effectiveness of our solution on two tasks, namely privacy protection strength and recommendation effectiveness. In particular, we aim to answer the following research questions (RQs):

TABLE 1: Features extracted from the dataset.

- **Number of interacted products**
- **Number and percentage of each rating level** (i.e., 1-5) given by a user
- **Ratio of positive and negative ratings**: The percentage of low ratings (1 and 2) and high ratings (4 and 5) of a user.
- **Entropy of ratings**: It is calculated as $-\sum_{\forall r} Per_r \log Per_r$, where Per_r is the percentage that a user gives the rating of r.
- **Median, min, max, and average of ratings**
- **Gender**: It is either male or female.
- **Occupation**: A total of 21 possible occupations are contained.
- **Age**: Age attribute is divided into 3 groups: under 45, over 35, and between 35 and 45.

- **RQ1:** Can our model effectively protect personal attributes in the presence of attribute inference attacks?
- **RQ2:** How does our method perform in the recommendation task?
- **RQ3:** What is the contribution of the novel adaptive privacy-protection mechanism?
- **RQ4:** Can our model resist attribute inference attacks that utilize different kinds of attack models?
- **RQ5:** what is the impact of key hyperparameters in privacy-preserving strength and recommendation effectiveness of our method?

5.1 Datasets

We use two publicly available datasets for evaluation, namely **ML-100K** and **ML-1M** [37]. ML-100k contains $10,000$ ratings from 943 users on $1,682$ movies collected from the MovieLens website, while ML-1M is a larger dataset that contains 1 million ratings involving 6,039 users and 3,705 movies. Additionally, in each dataset, all users are associated with three private attributes, i.e., gender, age and occupation. In previous work [15], the experimental results show that the occupation attribute cannot be correctly inferred via the attribute inference attacks, compared to gender and age attributes. One possible reason is that occupation attribute is divided into 21 classes, which is hard for simple classification models to achieve acceptable accuracy. Therefore, we mainly focus on Gender and Age attributes in this work, and transfer the age and gender attributes into a 3- and 2-dimensional one-hot encoding vectors, following [15]. Table 1 provides a summary of all the user features we have used.

5.2 Evaluation Metrics

Recommendation Effectiveness. To evaluate the recommendation accuracy, we use the leave-one-out approach [38] to split datasets for evaluation. Specifically, one item is preserved as ground truth for each user to construct a test set. We leverage hit ratio at rank K ($Hit@K$) to measure the ratio of the ground-truth items that appear in the top-K recommendation lists. Note that we use the entire negative item sets rather than the sampled subsets to compute Hit@K.

Attribute Inference Attack Resistance. To quantify a model's resistance ability against attribute inference attacks, we employ a widely used classification metric *F1 score* to evaluate the inference accuracy of the attacker.

TABLE 2: Performance of attribute-inference attack.

Attribute	Method	ML-100k		ML-1M	
		$\zeta = 0.1$	$\zeta = 0.2$	$\zeta = 0.1$	$\zeta = 0.2$
Age	Pure FedRec	0.528	0.560	0.522	0.570
	F-DPMF	0.584	0.599	0.579	0.590
	F-GERAI	0.503	0.507	0.501	0.531
	F-GERAI-NL	0.509	0.559	0.505	0.539
	FedRec-GN	0.455	0.496	0.449	0.461
	FedNews	0.446	0.480	0.439	0.479
	ours	**0.429**	**0.470**	**0.414**	**0.434**
Gender	Pure FedRec	0.677	0.708	0.684	0.682
	F-DPMF	0.703	0.694	0.709	0.717
	F-GERAI	0.639	0.648	0.627	0.640
	F-GERAI-NL	0.657	0.658	0.634	0.665
	FedRec-GN	0.575	0.642	0.627	0.634
	FedNews	0.568	0.592	0.597	0.619
	ours	**0.559**	**0.581**	**0.574**	**0.603**

Fig. 5: Recommendation effectiveness results.

5.3 Baseline Methods

- **Pure FedRec**: This is a pure GCN-based federated recommender system described in Section 3.1 without any privacy protection mechanism.
- **F-GERAI**: For the fair comparison, we extend the centralized recommender GERAI proposed in [15] to a federated version that uses the information perturbation mechanism in DP to protect user attribute information.
- **F-GERAI-NL**. It is a variant of F-GERAI, which only enforces ϵ-differential privacy by perturbing the objective function.
- **F-DPMF**: For the fair comparison, we extend DPMF (Differentially Private Matrix Factorization) proposed in [39] to a federated version F-DPMF. In its local recommender, objective perturbation is applied to make sure that the updated item embeddings satisfy differential privacy.
- **FedNews** [29]: Fednews is the first work that adopts Laplace Noise-based LDP method in federated news recommender system. In order to fit our recommendation setting where there is not textual content for items, the base recommender is replaced by our GCN-based recommender (refer to details in Section 3.1).
- **FedRec-GN**: Based on Pure FedRec, we add Gaussian Noise [32] into the uploaded parameters of each client, which can mask the original information.

5.4 Parameters Settings

In FedRec, we set the latent dimension d, local learning rate, local batch size, local epoch to 64, 0.001, 32 and 5, respectively. Each device is assumed to contain an individual user, and 50% and 10% of users are randomly selected on ML-100K and ML-1M at each round. Model parameters in FedRec are randomly initialized using Gaussian distribution, which has 0 mean and a standard deviation of 1. In our proposed adaptive privacy-preserving mechanism, we set $\delta = 0.5$, the noise factor $\lambda \in \{0.017, 0.020, 0.025, 0.033\}$ and privacy parameter $\epsilon \in \{30, 40, 50, 60\}$.

5.5 Privacy Protection Effectiveness (RQ1)

Table 2 demonstrates the F1 Scores achieved by the attribute inference attacker on all baselines. Note that lower F1 Scores show higher resistance to attribute inference attacks. Firstly, the attacker achieves higher inference accuracy on
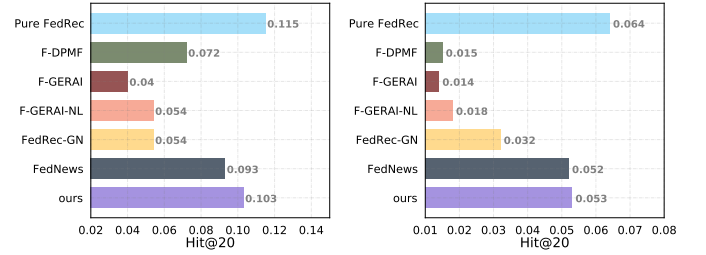
Pure FedRec than most of the recommender systems with differential privacy mechanisms, since it does not use any privacy-protection methods when uploading local model parameters to the central server. Notably, the attacker on F-DPMF achieves a better performance than FedRec, and a possible reason is that the local recommender of F-DPMF is a shallow model (i.e., Matrix Factorization) that simply represents users and items in a low dimensional latent space. Hence, massive original information can be preserved in the embeddings. Correspondingly, it is evidenced that the deep learning-based recommender can provide a stronger privacy guarantee due to the abstraction of multiple layers and complex nonlinear structure. Moreover, compared with recommenders that apply differential privacy on optimization process (i.e., F-GERAI, F-GERAI-NL and F-DPMF), the ones that make use of LDP methods (i.e., FedRec-GN, FedNews and ours) by directly adding noise into uploaded parameters show obvious superiority in defending against attribute inference attacks. The reason is that the privacy protection mechanisms utilized in those optimization perturbation methods cannot yield the same strength as the LDP-based methods in preventing the disclosure of sensitive information from model parameters. This also confirms that the adoption of DP-based approaches may preclude directly leaking private attributes, but these methods are unable to effectively perform higher privacy protection in the federated setting. Notably, compared with other LDP-based recommender systems, it is clear that our method yields the best performance in obscuring users' private attribute information. Finally, we can see that F-GERAI outperforms F-GERAI-NL in terms of F1 Score, due to the dual-stage perturbation setting where a relatively strong privacy protection method is applied for user feature perturbation. Meanwhile, our method can constantly achieve significantly better results without an extra privacy budget on original features, indicating our method endows the uploaded local model parameters a stronger privacy guarantee.

5.6 Recommendation Effectiveness (RQ2)

Recommendation accuracy is an important metric in the evaluation of privacy-preserving recommenders, since protecting user privacy is usually at the expense of their recommendation accuracy. Hence, a practical recommender should resist inference attacks without sacrificing high-quality recommendations. We report all methods' performance on personalized recommendation w.r.t. Hit@20 in Figure 5, and higher Hit@20 values imply higher recommendation quality. Clearly, recommendation methods that make use of privacy protection mechanisms have significant
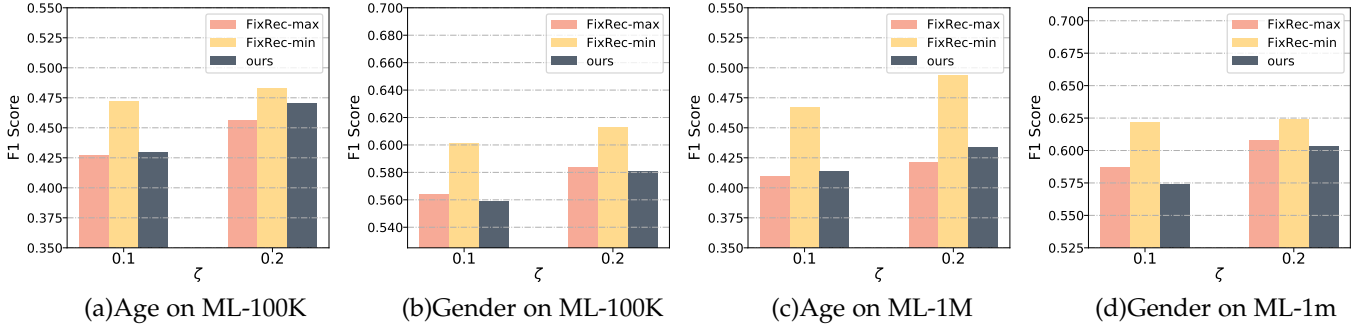
Fig. 6: Evaluation performance of adaptive privacy-preserving mechanism against attribute inference attacks with two privacy-preserving recommenders with fixed privacy budget.

TABLE 3: Recommendation results of our model with adaptive privacy budget and two variants with fixed privacy budget.

Method	ML-100K	ML-1M
FixRec-max	0.067	0.043
FixRec-min	0.105	0.054
ours	0.103	0.053

performance disparity in terms of the Hit@20. Our method outperforms all privacy-preserving baselines by a large margin in both datasets, thanks to our proposed adaptive perturbation mechanism in which the values of privacy budget are dynamically adjusted according to the vulnerability of each model component. Furthermore, LDP-based recommenders achieve significantly better results than DP-based ones while maintaining stronger privacy protection, which further confirms the effectiveness of LDP-based mechanisms in the federated scenarios. Although F-GERAI can achieve promising recommendation performance in the traditional centralized setting [15], it does not fit well the federated setting, as the way it adds noise to the side information is specialized for the centralized recommendation, and it is harmful to the federated recommendation. Compared with FedRec-GN, Fednews achieves higher recommendation accuracy, which implies that Laplace Noise can ensure recommendation effectiveness while avoiding breaching users' privacy. This is the reason that we adopt it as the basic privacy protection mechanism in our method.

5.7 Importance of Adaptive Privacy Mechanism (RQ3)

To better understand the benefits brought by our proposed adaptive privacy mechanism, we compare with two variants of our method, FixRec-max and FixRec-min that hold fixed privacy budgets on ML-100K and ML-1M. Setting an appropriate value for the privacy budget λ is crucial, as it controls the trade-off between privacy protection level and recommendation accuracy. To explore the suitable range of λ in Eq (12), we first fix privacy parameter ϵ for all model components and then vary its value while keeping the other hyperparamerters unchanged. From the experimental results, we find that the value of ϵ should be limited to $E = [40, 60]$ and thus λ should be within the limits of $\mathcal{H} = [0.017, 0.033]$. A larger λ can cause a severe performance drop on recommendation accuracy, while the recommender with a too small value of λ fails to obscure users'

private attribute information. Hence, we set the privacy budget λ separately for each component according to their resistances obtained in Section 3.3, that is 0.017 ($\epsilon = 60$) for Item Component, 0.025 ($\epsilon = 40$) for first layer of MLP, 0.020 for second layer of MLP ($\epsilon = 50$) and 0.033 ($\epsilon = 30$) for User Component. Then we report the attribute inference results and recommendation results of our method and the two variants that respectively adopt the minimum fixed privacy budget (named as FixRec-min) and the maximum fixed privacy budget (named as FixRec-max) of $\mathcal{H}$.

From Figure 6, we can see that FixRec-max achieves the best performance in defending against attribute inference attacks, since a larger λ requires a larger amount of noise to be injected into the model parameters, leading to the larger information obfuscation. It is worth mentioning that our method not only outperforms FixFed-min but also achieves comparable results with Fixed-max, indicating our adaptive privacy mechanism can still effectively provide satisfactory privacy level with a lower privacy budget.

Table 3 shows that FixRec-min outperforms FixRec-max by a large margin, especially on ML-100K dataset. The reason is that a larger amount of noise is injected to the training process of FixRec-max, which negatively influences the recommendation accuracy. Furthermore, our model significantly outperforms FixRec-max and yields recommendation results that are close to the FixRec-min, indicating that an adaptive privacy budget can be beneficial to significantly reduce the utility loss that is inevitably caused by the LDP-based approaches. Notably, our method is able to achieve comparable recommendation results as FixRec-min and resistance ability as FixRec-max. It confirms the effectiveness of our proposed adaptive privacy mechanism, which helps our federated recommender system resist attribute inference attack and avoid unnecessary utility loss.

5.8 Robustness against Different Attribute Inference Attackers (RQ4)

In real-life scenarios, they are many available models that can be selected to perform attribute inference attacks for the adversary, so the attack models are usually unknown and unpredictable. To better understand the vulnerability of our method and other comparable methods, we design several different types of attack models, namely Decision Tree (DT), SVC and KNN, that are frequently adopted approaches in classification tasks. In this study, we use the full

TABLE 4: Performance of attribute-inference attack w.r.t. different types of attacker.

Attribute	Method	ML-100k			ML-1M		
		DT	SVC	KNN	DT	SVC	KNN
Age	Pure FedRec	0.427	0.465	0.491	0.460	0.487	0.520
	DPMF	0.417	0.401	0.512	0.443	0.390	0.449
	F-GERAI	0.406	0.422	0.483	0.415	0.408	0.425
	F-GERAI-NL	0.413	0.474	0.488	0.419	0.449	0.430
	FedRec-GN	0.399	0.380	0.446	0.397	0.381	0.445
	FedNews	0.401	0.406	0.417	0.388	0.390	**0.417**
	ours	**0.394**	**0.356**	**0.413**	**0.377**	**0.333**	0.428
Gender	Pure FedRec	0.601	0.566	0.677	0.671	0.640	0.680
	DPMF	0.580	0.559	0.656	0.572	0.520	0.647
	F-GERAI	0.571	0.566	0.613	0.612	0.559	0.599
	F-GERAI-NL	0.580	0.587	0.646	0.618	0.566	0.631
	FedRec-GN	0.564	0.481	0.625	0.616	0.520	0.601
	FedNews	0.561	0.474	**0.608**	0.614	0.513	0.596
	ours	**0.556**	**0.472**	0.613	**0.550**	**0.496**	**0.550**
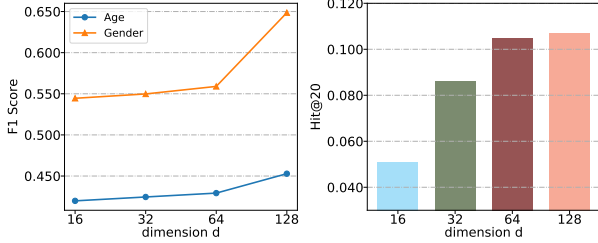
Fig. 7: Inference attack and recommendation results on ML-100K w.r.t. dimension d

TABLE 5: Inference attack results on ML-100K w.r.t. train-test split ratio ζ.

Attribute	ζ	Attack Models (F1 Score)			
		DNN	DT	SVC	KNN
Age	10%	0.429	0.394	0.356	0.413
	20%	0.470	0.403	0.361	0.430
	30%	**0.498**	**0.403**	**0.382**	**0.436**
Gender	10%	0.559	0.556	0.472	0.613
	20%	0.581	0.570	**0.483**	0.623
	30%	**0.585**	**0.572**	0.482	**0.627**

version of $\Delta\Theta$ that is derived from well-trained federated recommenders for all attackers and set $\zeta = 0.1$. Table 4 shows the attribute inference accuracy of each attacker. It is obvious that our proposed method outperforms all the comparison methods in most scenarios, which implies that our method can more effectively defend against attribute inference attacks and provide a stronger privacy guarantee when confronted with unknown attacker models. Though Fednews achieves slightly better results when attacker is a KNN-based model, it falls behind our model in all other cases and yields inferior performance in recommendation task. Furthermore, the FedRec-GN cannot perform as well as Fednews that uses Laplace Noise, which further verifies the advantages of utilizing Laplace Noise-based LDP method in defending against inference attacks.

5.9 Parameter Sensitivity (RQ5)

We answer RQ5 by investigating the performance fluctuations of our method with varied hyperparameters, particularly embedding dimension d and train-test split ratio ζ. Due to the space limitation, we only showcase the results on ML-100K dataset, and similar results are also achieved on ML-1M dataset. Specifically, we tune the value of d or ζ while keeping the other hyperparameters unchanged, and record the new recommendation accuracy and inference attack results achieved in Figure 7 and Table 5.

Dimension d. We examine the value of dimension d in $\{16, 32, 64, 128\}$. In general, the dimension d mainly controls the models' expressiveness. Obviously, the recommendation accuracy of our model benefits from a relatively larger dimension d, and then the performance gain appears

to become less significant when d reaches 64. In the attribute inference task, our method with a smaller d shows higher resistance to the inference attack. One possible reason is that the core of a classification model is the ability to assign a class to an object based on input features. A larger d can increase the model's expressiveness and thus the attacker can learn more information from the model parameters. Fortunately, our model is able to achieve competitive performance in both two tasks when $d = 64$.

Train-test split ratio ζ. Table 5 shows attack accuracy for different types of attackers with varied prior knowledge (i.e., the size of training data). As expected, increasing the size of the attackers' training dataset improves the accuracy of the attribute inference attacks. Another observation is that DNN-based attacker outperforms other attackers in most scenarios, since its superiority in learning non-linear correlation between input features and target labels. Though the attackers collect much more prior knowledge (i.e., $\zeta = 30\%$), our model still achieves acceptable performance in resistance to attribute inference attacks.

6 RELATED WORK

Attribute Inference Attacks and defenses. Attribute inference attacks aim to infer users' sensitive information by carefully designing an attacker model based on the collected information such as outputs and structure of the target model, and training dataset. [40], [41], [42] infer attributes information by incorporating available target users' friend information. Behavior-based approaches construct attack models based on users' behavioral data (e.g., movie-rating behavior [8] and Facebook likes [9]). [13], [43], [44] achieve adversarial purpose by leveraging both users' friend and behavior information. In the centralized recommendation

context, attribute inference attacks attempt to infer users' private information (e.g., demographic features) from publicly available information (i.e. recommendation results and user-item interaction data). To address such privacy issues, there have been emerging research efforts on developing privacy-preserving centralized recommender systems [15], [16]. RAP is proposed to enhance attack-resistance of conventional recommender systems by utilizing an adversarial learning paradigm where a recommender model and a predefined attack model are trained against each other. But the design of RAP makes it effectively defend against a specific attacker model. Another variant is encryption-based methods that use encryption techniques such as homomorphic encryption [26], [45]. However, in these approaches, an extra third-party crypto-service provider is required, so they are computation intensive. Recently, differential privacy becomes a well-established technique to address privacy issues, since it can provide a mathematically provable guarantee [46], [47], [48]. For example, GERAI [15] is proposed to perturb the user's side information and optimization process to prevent privacy leakage via recommendations. However, the centralized recommender systems that store and train users' personal data centrally are still suffering from enormous and unprecedented privacy issues.

Federated Learning. To tackle privacy issues existing in centralized scenarios, a common practice is to deploy the online system in a federated setting, which enables users to collaboratively learn a global model while keeping all the sensitive data on local devices. Federated learning starts training by initializing a shared global model, then a subset of existing clients is selected to train their local models based on the private dataset and submit the updated model parameters. With these updates, the server operates aggregation of the received updates to replace the parameters of the global model. There are some works that attempt to develop federated recommender systems. [27] aims to bind Matrix Factorization approach into the federated setting. In FedFast [21], a novel sampling method that selects participating clients in each training iteration and an activate aggregation rule that combines locally trained models are devised to reduce the communication cost and speed up the convergence rate of current federate recommender systems. [49] applies meta learning in the federated model with a shared meta learner, which differs from the conventional FL setting which shares a global model. Though federated recommender can achieve satisfactory recommendation accuracy without accessing users' data, recent works show that it is yet to provide a privacy guarantee of users' privacy. In [22], the central server adopts a DP-based mechanism to perturb global recommender, which can defend against attacks from malicious participants who can infer private information via the shared global model. However, it is only effective when the central server is in a sterile environment. In real-life scenarios, the server is curious about inferring the users' private attributes via received updates, leaving a backdoor for attribute inference attacks. Hence, privacy protection methods should be applied to each individual's local model parameters before sending to the server. To enhance privacy protection, [20], [29] applied LDP-based approaches on the local model parameters. However, the existing privacy-preserving works in federated learning as-

sume all components exhibit the same resistance degree, which is violated and causes a severe recommendation performance drop. These limitations motivated us to propose an adaptive privacy-preserving recommender system that is able to counter attribute inference attacks in the federated setting, while maintaining high recommendation accuracy.

7 CONCLUSION AND FUTURE WORK

In this paper, we focused on the privacy issues of federated recommenders confronted with attribute inference attacks. To provide a comprehensive analysis of current federated recommender systems, we design a novel attribute inference attacker to show the vulnerability of each internal component of the recommender. In accordance with the experimental results, we proposed an adaptive privacy-preserving federated recommender system to protect users' sensitive data in defending against inference attacks while maintaining high-quality recommendation results. Specifically, to minimize utility loss caused by LDP-based approaches, we improve the naive LDP mechanisms with an adaptive privacy budget based on the resistance degree. The experimental results validate the superiority of our solution by comparing with the baseline approaches. In the future work of privacy-preserving federated recommenders, it will be appealing to further investigate privacy protection against active attackers that can participate in the training of federated recommender systems and craft adversarial parameter updates for follow-up attribute inference attacks.

REFERENCES

[1] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, and X. Li, "Air: Attentional intention-aware recommender systems," in *ICDE*, 2019, pp. 304–315.

[2] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *ICDE*, 2019, pp. 566–577.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *SIGKDD*, 2018, pp. 974–983.

[4] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *SIGIR*, 2019, pp. 165–174.

[5] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhance social recommendation with adversarial graph convolutional networks," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[6] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*, 2011, pp. 217–253.

[7] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui, "Gcn-based user representation learning for unifying robust recommendation and fraudster detection," in *SIGIR*, 2020, pp. 689–698.

[8] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *RECSYS*, 2012, pp. 195–202.

[9] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *PNAS*, pp. 5802–5805, 2013.

[10] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, "" you might also like:" privacy risks of collaborative filtering," in *IEEE symposium on security and privacy*, 2011, pp. 231–246.

[11] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, "User identity linkage across online social networks: A review," *Acm SIGKDD Explorations Newsletter*, pp. 5–17, 2017.

[12] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *WWW*, 2013, pp. 447–458.

[13] N. Z. Gong and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors," in *USENIX*, 2016, pp. 979–995.

[14] A. Naranyanan and V. Shmatikov, "Robust de-anonymization of large datasets," in *IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[15] S. Zhang, H. Yin, T. Chen, Z. Huang, L. Cui, and X. Zhang, "Graph embedding for recommendation against attribute inference attacks," in *WWW*, 2021, p. 3002–3014.

[16] G. Beigi, A. Mosallanezhad, R. Guo, H. Alvari, A. Nou, and H. Liu, "Privacy-aware recommendation with private-attribute protection using adversarial learning," in *WSDM*, 2020, pp. 34–42.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3844–3852.

[18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[20] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," *arXiv preprint arXiv:2102.04925*, 2021.

[21] K. Muhammad, Q. Wang, D. O'Reilly-Morgan, E. Tragos, B. Smyth, N. Hurley, J. Geraci, and A. Lawlor, "Fedfast: Going beyond average for faster training of federated recommender systems," in *SIGKDD*, 2020, pp. 1234–1242.

[22] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, "Fast-adapting and privacy-preserving federated recommender system," *VLDBJ*, 2021.

[23] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[24] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[25] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, 2019.

[26] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, 2020.

[27] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.

[28] S. Zhang, H. Yin, T. Chen, Z. Huang, Q. V. H. Nguyen, and L. Cui, "Pipattack: Poisoning federated recommender systems for manipulating item promotion," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1415–1423.

[29] T. Qi, F. Wu, C. Wu, Y. Huang, and X. Xie, "Privacy-preserving news recommendation model learning," in *EMNLP*, 2020, pp. 1423–1432.

[30] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *arXiv preprint arXiv:1606.05053*, 2016.

[31] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *ICDE*, 2019, pp. 638–649.

[32] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, pp. 211–407, 2014.

[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017, pp. 1273–1282.

[34] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[35] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *SIGSAC*, 2016, pp. 308–318.

[36] Z. Bu, J. Dong, Q. Long, and W. J. Su, "Deep learning with gaussian differential privacy," *Harvard data science review*, vol. 2020, no. 23, 2020.

[37] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems*, pp. 1–19, 2015.

[38] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*, 2016, pp. 549–558.

[39] H. Jingyu, X. Chang, and Z. Sheng, "Differentially private matrix factorization," in *IJCAI*, 2015, pp. 57–62.

[40] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Inferring private information using social network data," in *WWW*, 2009, pp. 1145–1146.

[41] J. He, W. W. Chu, and Z. V. Liu, "Inferring privacy information from social networks," in *ISI*, 2006, pp. 154–165.

[42] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *ACM Transactions on Intelligent Systems and Technology*, pp. 1–20, 2014.

[43] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "Attriinfer: Inferring user attributes in online social networks using markov random fields," in *WWW*, 2017, pp. 1561–1569.

[44] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *ACM Transactions on Privacy and Security*, pp. 1–30, 2018.

[45] J. Kim, D. Koo, Y. Kim, H. Yoon, J. Shin, and S. Kim, "Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption," *ACM Transactions on Privacy and Security*, pp. 1–30, 2018.

[46] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *SIGKDD*, 2009, pp. 627–636.

[47] A. Berlioz, A. Friedman, M. A. Kaafar, R. Boreli, and S. Berkovsky, "Applying differential privacy to matrix factorization," in *RECSYS*, 2015, pp. 107–114.

[48] Z. Liu, Y.-X. Wang, and A. Smola, "Fast differentially private matrix factorization," in *RECSYS*, 2015, pp. 171–178.

[49] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

Shijie Zhang is currently a computer science Ph.D. student at the School of Information Technology and Electrical Engineering, The University of Queensland. She obtained Master Degree from The University of Queensland in 2018 and Bachelor Degree from Shandong University. Her research interests include data mining, recommender system, deep learning, and federated learning.

Hongzhi Yin received the Ph.D. degree in computer science from Peking University in 2014. He is an Associate Professor with the University of Queensland. He received the Australia Research Council Future Fellowship and Discovery Early-Career Researcher Award in 2021 and 2016. He was recognized as Field Leader of Data Mining & Analysis in The Australian's Research 2020. His research interests include recommender system, graph embedding and mining, chatbots, social media analytics and mining, edge machine learning, trustworthy machine learning, decentralized and federated learning, and smart healthcare. He has published 200+ research papers in the top venues, including 15 publications in Top 1% (CNCI), 110+ CCF A and 70+ CCF B as well as 6 Best Paper Awards such as ICDE'19 Best Paper Award, DASFAA'20 Best Student Paper Award. He is currently serving as Associate Editor/Guest Editor/Editorial Board for ACM Transactions on Information Systems (TOIS), ACM Transactions on Intelligent Systems and Technology (TIST), etc.