Robustness of privacy-preserving collaborative recommenders against popularity bias problem

Mert Gulsoy[1,2]**, Emre Yalcin**[3]**, and Alper Bilge**[2]

[1]**Distance Education Research Center, Alaaddin Keykubat University, 07400, Antalya, Turkey**
[2]**Computer Engineering Department, Akdeniz University, 07058, Antalya, Turkey**
[3]**Computer Engineering Department, Sivas Cumhuriyet University, 58140, Sivas, Turkey**

Corresponding author:
Emre Yalcin[3]

Email address: eyalcin@cumhuriyet.edu.tr

ABSTRACT

Recommender systems have become increasingly important in today's digital age, but they are not without their challenges. One of the most significant challenges is that users are not always willing to share their preferences due to privacy concerns, yet they still require decent recommendations. Privacy-preserving collaborative recommenders remedy such concerns by letting users perturb their sensitive preferences before submitting to the recommendation provider. Another recently discussed challenge is the problem of popularity bias, where the system tends to recommend popular items more often than less popular ones, limiting the diversity of recommendations and preventing users from discovering new and interesting items. In this paper, we comprehensively analyze the randomized perturbation-based data disguising procedure of privacy-preserving collaborative recommender algorithms against the popularity bias problem. For this purpose, we construct user personas of varying privacy protection levels and scrutinize the performance of ten recommendation algorithms on these user personas regarding the accuracy and beyond-accuracy perspectives. We also investigate how well-known popularity-debiasing strategies combat the issue in privacy-preserving environments. In experiments, we employ three well-known real-world datasets. The key findings of our analysis reveal that privacy-sensitive users receive unbiased and fairer recommendations that are qualified in diversity, novelty, and catalogue coverage perspectives in exchange for tolerable sacrifice from accuracy. Also, prominent popularity-debiasing strategies fall considerably short as provided privacy level improves.

Keywords: Recommender systems, popularity bias, privacy-preserving, collaborative filtering, unfairness

INTRODUCTION

Following the advent of online facilities through the Internet, many helpful digital platforms have been developed to support people in several aspects, such as daily activities (i.e., shopping, listening to music, watching movies) or interacting with people via social media. However, the amount of available information on these platforms has reached overwhelming levels in recent years, also known as the *information overload* problem, making it difficult for users to deal with (Jannach and Zanker, 2022). Recommender systems (RSs) are introduced as artificial intelligence-based solutions to overcome this problem by suggesting appropriate items/services to users' expectations/desires (Ko et al., 2022). Since they provide many benefits for two fundamental stakeholders of the platforms, i.e., *users* and *goods or service providers*, many prominent digital applications for different domains, such as Netflix[1], Twitter[2],

---

[1]https://netflix.com/
[2]https://twitter.com/

Spotify[3], YouTube[4], and Booking.com[5], have integrated such tools into their systems and continually attempted to improve them to provide better referrals to individuals.

The most effective approaches to providing individual recommendations are collaborative filtering (CF) techniques (Su and Khoshgoftaar, 2009), although some other scenario-specific practical approaches, like content- (Shambour et al., 2022) or knowledge-based filtering methods (Agarwal et al., 2022) exist. By operating on a user-item preference matrix, the collaborative recommenders usually consider similarities between users or items and perform recommendation tasks by (*i*) computing a prediction score for items that are not experienced by users or (*ii*) generating ranked lists (i.e., top-*N* lists) including recommended items for users (Bobadilla et al., 2013). Over the last two decades, many different algorithms following this approach have been proposed, and also, there exists a rich literature with their main limitations (Koren et al., 2022). Even if pioneering research has commonly aimed at handling their scalability, accuracy, or cold-start issues, their inclinations to feature a few popular items in their recommendations by disregarding many other ones, also referred to as the *popularity-bias* problem, have become one of the most critical concerns of RSs in recent years (Abdollahpouri et al., 2020; Elahi et al., 2021).

Such undesired bias propagation of the CF recommenders might be very problematic for users and service providers in real-world applications. On the one hand, users cannot receive qualified recommendations regarding beyond-accuracy aspects such as diversity, serendipity, and novelty, resulting in substantial decreases in their satisfaction with the platform (Yalcin and Bilge, 2022). On the other hand, items in the catalogue are not evenly represented in the recommendations, which leads to unfair competition in the market. Worse, if such weakness exists in the system, the platform might be subjected to manipulative attacks of malicious stakeholders aiming at featuring their products in recommendation lists or demoting rivals' items to obtain financial benefits (Si and Li, 2020). Also, for specific application domains, such as social media, such weakness could be viciously abused to manipulate users by dragging them into ideological echo chambers (Cinelli et al., 2021). Because of these reasons, several recent studies in RSs have evaluated the adverse effects of the popularity bias of the recommenders comprehensively and concerned to develop various treatment strategies, also referred to as popularity-debiasing methods, to provide more balanced recommendation lists in terms of item popularity (Abdollahpouri et al., 2019a; Yalcin and Bilge, 2021; Boratto et al., 2021). Even if several popularity-debiasing methods exist, *post-processing* ones attempting to re-ranking recommendation lists by penalizing popular items are the most utilized ones since they are independent of the internal mechanism of the considered recommendation approach and easily applied to the output of any recommenders.

The common agreement regarding why such a bias against popular items occurs in recommendations is the imbalances in the rating distribution observed in the original preference data on which the recommendation algorithm is trained (Abdollahpouri et al., 2018). More concretely, a few popular items typically receive most of the provided ratings in the dataset. In contrast, the remaining great majority of items are evaluated by a small number of users, which ends up with user preferences dispersed with a long-tail shape across items, as presented in Fig. 1 for three benchmark data collections crawled for different application domains.

Another primary limitation of CF recommenders is that it might be challenging to collect confidential preference data of users because of some general reasons, such as the lack of functionality of user interfaces and the laziness of individuals. This issue also might be more problematic, especially for specific application domains like social media, since users avoid sharing their opinions on sensitive content due to data leakage concerns (Lu et al., 2022). Being unable to obtain genuine profiles of users mostly ends up with unqualified recommendations, especially regarding predictive accuracy, as traditional CF recommenders are trained on data consisting of unreliable ratings (Yargic and Bilge, 2019). Therefore, there has been considerable interest in privacy-preserving collaborative filtering (PPCF) to provide accurate predictions at a satisfactory level without violating such privacy concerns of individuals (Ozturk and Polat, 2015; Pramod, 2022). Although several PPCF schemes exist, the most prominent approach in central server-based recommenders is to apply randomized perturbation and obfuscation techniques during preference elicitation stages to mask (*i*) the list of actually rated items and (*ii*) actual ratings given for these items (Polat and Du, 2003; Yargic and Bilge, 2019). Since the perturbation process is performed by adding random noise from a probability distribution with appropriate parameters, the

---

[3]https://spotify.com/
[4]https://youtube.com/
[5]https://booking.com/
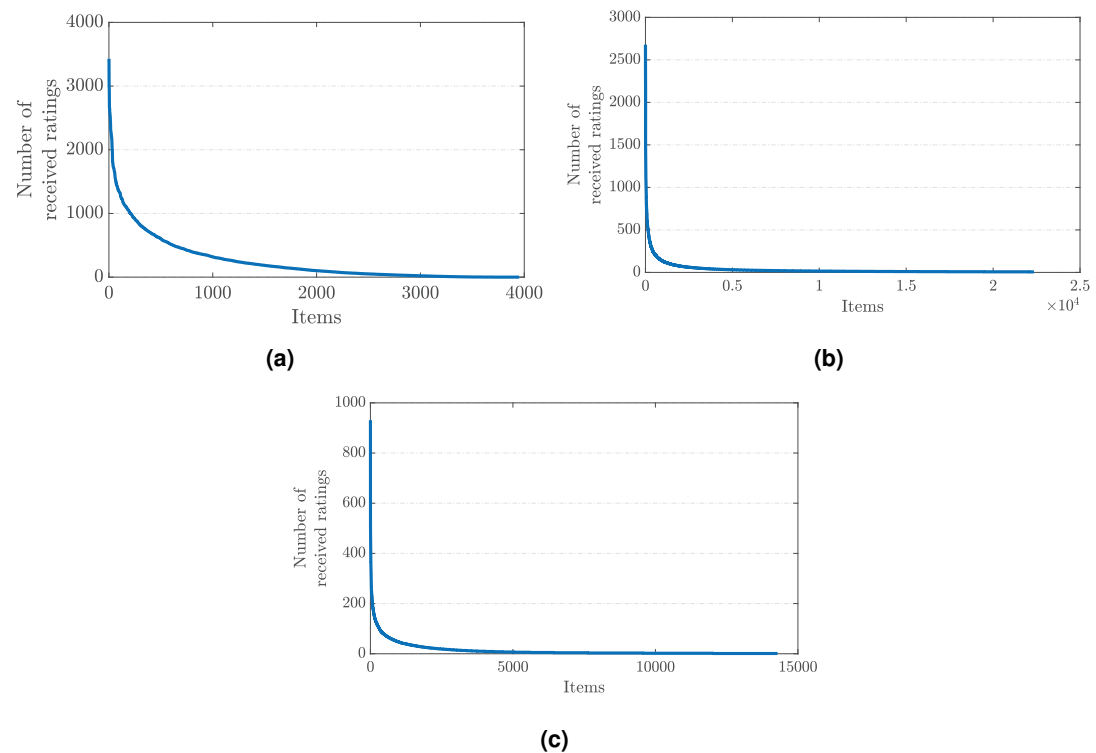
Figure 1. The long-tail distribution of the provided user preferences in the (a) MovieLens-1M (movie), (b) Douban Book (book), and (c) Yelp (local business) data collections.

aggregate still enables the central server to provide dependable yet private recommendations to users. Therefore, PPCF algorithms operate on a disguised version of the original preference collection where the perturbation level balances the confliction goals of accuracy and privacy.

The random noise inserted into a user profile conceals the list of actually rated items by injecting fake votes into some unrated cells, which might lead to significant changes in the distribution of ratings across items. Therefore, this process might cause the level of observed popularity bias in the recommendations to be significantly altered, as the main reason for this issue is the imbalances in the dispersion of user preferences. Moreover, injecting fake ratings into user profiles alters each item's total number of ratings received. Therefore, it would be expected that the performances of the popularity bias treatment methods, especially the *post-processing* ones, would significantly vary since they are mostly based on penalizing items based on their popularity rate (i.e., the total number of ratings received) while re-ranking recommendation lists (Abdollahpouri et al., 2019a; Yalcin and Bilge, 2021). Although privacy-preserving recommendation systems are of crucial importance with the increasing focus on data protection regulations today, their popularity bias propagation behaviour remains elusive, and how well-known popularity-debiasing methods would perform on these systems is not investigated.

In this study, we attempt to present a comprehensive analysis evaluating how randomized perturbation-based privacy-preserving strategies affect popularity bias propagation in recommendations. In addition, we consider three prominent popularity-debiasing methods and scrutinize their performance with varying levels of privacy provided for users. For this purpose, we specify four user personas according to their desired privacy-protection level. The first persona, *Utterly Unconcerned (UU)*, represents users who do not care about their privacy and require the highest possible level of accuracy in received recommendations. Note that *UU* also refers to users of traditional non-private CF recommenders. The second persona, i.e., *Barely Bothered (BB)* corresponds to users who are not so worried if their privacy is violated, and they pick low-level privacy parameters in a request of good accuracy. The third persona, *Pretty Pragmatists (PP)*, represents users who do not compromise their privacy but are still interested in receiving decent recommendations. Therefore, they pick privacy parameters to balance accuracy and confidentiality. Finally, the fourth persona, *Fiery Fundamentalists (FF)*, resembles highly concerned people with their

privacy, so they are only willing to share extremely perturbed preferences sacrificing accuracy completely. In a nutshell, the main contributions of our study are given in the following.

- We present a comprehensive experimental analysis of how much the popularity bias is propagated in recommendations for four scenarios, i.e., *UU*, *BB*, *PP*, and *FF*, on three benchmark datasets using ten state-of-the-art recommendation algorithms of different families.

- We analyze how the adverse effects of the popularity bias problem on the accuracy and beyond-accuracy recommendation quality differentiate for such privacy level-based user personas.

- We consider three well-known popularity debiasing methods and observe whether they still perform similarly when *BB*, *PP*, or *FF* scenarios are considered rather than no privacy protection (i.e., the case of *UU*) provided.

The remaining sections of the study are organized as follows, respectively: (*i*) the first presents a review of both privacy-preserving methods for CF algorithms and popularity bias-concerned studies, (*ii*) the second explains how we construct user personas according to varying privacy protection requirements, (*iii*) the third explains the experimental setups, including utilized datasets, recommendation algorithms, and evaluation protocols, followed by experimentation methodology and considered popularity-debiasing methods, (*iv*) the fourth analyzes how privacy-preserving strategies resist the popularity-bias problem, and the next section investigates how popularity-debiasing methods act for privacy-based user personas, and (*v*) the final concludes the study and introduces our future directions. Note that the source code accompanying this study is made publicly available[6] for reproducibility purposes.

RELATED WORK

Since our research is mainly based on two fundamental problems related to recommendation approaches, i.e., privacy protection and popularity bias issues, we organize this section into two sub-sections. The first provides a literature review on privacy-preserving strategies introduced for CF recommenders, while the second presents recent studies on the popularity bias problem and the treatment methods to cope with this issue.

Privacy-preserving approaches for CF recommenders

Recent years have seen a significant increase in research into the privacy concerns raised by conventional CF systems. Many studies have concentrated on developing PPCF approaches to address these issues (Wei et al., 2018). These approaches are commonly designed using *k*-anonymity, obfuscation, differential privacy, cryptography, and perturbation-based techniques.

For distributed recommendation systems, cryptographic techniques are frequently utilized to protect the privacy of individuals (Badsha et al., 2016; Li et al., 2017). For example, an unsynchronized secure multi-party computation protocol was introduced by Li et al. (2016). This protocol incrementally computes similarities between items without requiring individuals to be present when the calculation stages are performed. As another example, Shmueli and Tassa (2017) have offered a distributed recommendation strategy in which vendors communicate encrypted data with a mediator over secure protocols to calculate predictions or ranking items in generating recommendation lists. Also, some recent studies have used homomorphic encryption to encrypt the Quality of Service values to recommend personalized, high-quality web services based on user and location data without revealing confidential information (Badsha et al., 2018). Furthermore, Li et al. (2017) have proposed a successful approach for the group recommender systems that aim to safeguard users' privacy while making recommendations. The process of computing similarities between items is considered a probabilistic inference issue in Zou and Fekri (2015), and a semi-distributed belief propagation network-based approach is introduced for item-based PPCF systems. This way, disclosure of user preferences is prevented by including a subset of individuals in the network at a specific time.

Another crucial method for protecting user privacy is anonymization, which aims to break the link between particular users and their rating profiles. For example, a micro aggregation-based PPCF solution is suggested by Casino et al. (2015) to ensure user privacy protection by using *k*-anonymity masks. Chen

---

[6]https://doi.org/10.5281/zenodo.7789914

169 and Huang (2012) explore achieving *k*-anonymity for large-scale and sparse database kinds (e.g., RSs)
170 and introduce a *k*-anonymity heuristic approach based on clustering for protecting privacy in data sharing.
171 In addition, Wei et al. (2018) propose a (*p*, *l*, *a*)-diversification technique for improving the performance
172 of existing *k*-anonymity methods, where *p* denotes previous information on the rating profile of the users
173 and (*l*, *a*) denotes user diversity to increase the degree of provided privacy. As the last example of the
174 anonymization-wised method, Zhang et al. (2018) apply this strategy for a scenario of privacy-aware
175 smart cities and model privacy issues in cities as a PPCF issue.

176 Differential privacy is another prominent successful mechanism employed in CF-based approaches
177 to quantify the loss of privacy that may occur when a user's personal preferences are considered when
178 generating a suggestion. Dwork et al. (2006) is the pioneering study propounding the differential privacy
179 concept, and it is later used by McSherry and Mironov (2009) for the RSs domain by randomizing actual
180 user preferences using a private covariance matrix. A more recent study (Guerraoui et al., 2015) offers a
181 distance-based differential privacy solution to alter real profiles by exchanging components at specific
182 distances within the profile. Shen and Jin (2014) propose an alternative method that combines noise
183 calibration, ensures theoretical privacy and utility, and allows for adjustable instance-based noise injection
184 into the preference profiles of the users. Finally, Hou et al. (2018b) suggest a two-stage methodology for
185 neighbourhood-based differential privacy recommendations and private neighbour selection in the context
186 of medical recommendations.

187 In CF systems, data modification techniques like data obfuscation and randomized perturbation are
188 the most used to protect privacy in the literature. More specifically, the data obfuscation technique
189 tries to conceal private information while maintaining access to the information required for making
190 recommendations. For example, Parameswaran and Blough (2007) present a permutation-based data
191 obfuscation technique for central server-based CF applications. Also, a user-based PPCF recommender
192 utilizing semi-honest parties to execute additional calculations is proposed by Badsha et al. (2017). For
193 distributed recommendation contexts, Boutet et al. (2016) suggest employing coarse-grained obfuscated
194 versions of user profiles to reveal private information. Also, Elmisery and Botvich (2017) extend
195 traditional sole obfuscation level-based methods to identify more reliable target users by considering
196 arbitrary obfuscation degrees.

197 Randomized Perturbation Techniques (RPTs), an alternative to previous approaches, can safeguard
198 user privacy by falsifying actual user preferences before they are sent to the recommendation server (Polat
199 and Du, 2003, 2005a,b). For example, Bilge and Polat (2013) develop a scalable PPCF technique based
200 on bisecting *k*-means clustering and randomized perturbations. In another study (Gong, 2011), the privacy
201 of user-profiles spread across various repositories in recommendation processes is protected by combining
202 RPTs with secure multi-party computation techniques. By utilizing various levels and ranges of random
203 values, Polatidis et al. (2017) suggest enhancing the RPT-based PPCF process. In addition, Liu et al.
204 (2017) develop a hybrid approach to privacy preservation that combines RPTs with differential privacy to
205 offer more robust user preference protection than current RPT-based approaches. Finally, Yargic and Bilge
206 (2019) apply RPTs to protect individuals' privacy in the multi-criteria recommender systems domain.

207 Apart from the RSs domain, some prominent privacy-protection mechanisms also exist, especially
208 for machine learning, particularly federated machine learning (Yin et al., 2021). For example, Xu et al.
209 (2019) have introduced *HybridAlpha* to protect against privacy risks in federated learning using Secure
210 Multiparty Computation (SMC) protocol based on functional encryption. Another recent study offers a
211 hybrid technique combining SMC and differential privacy to cope with the trade-off between accuracy
212 and small data in parties, including small amounts of data (Truex et al., 2019). Also, Kerkouche et al.
213 (2021) have introduced a privacy protection mechanism for federated learning in the context of medical
214 research and patient care, relying on differential privacy to improve bandwidth efficiency by enabling
215 confidentiality. Finally, Lu et al. (2022) introduce a federated learning approach to detect malfunctions
216 that may occur in wind turbines in an organization in the energy sector. To ensure inter-institutional data
217 privacy, they have enriched their proposed approach with a biometric authentication technique to reduce
218 privacy-preserving risks.

219 As it is known, RSs typically need rating data of individuals to produce appropriate referrals to
220 them. Therefore, the size/dimension of data collections for prediction purposes directly affects the
221 recommendation quality; the need for big data also brings two different problems. One is the transfer,
222 analysis, and processing costs of such massive data. Second, it is not easy to ensure the confidentiality
223 of big data and protect user privacy. Some recent studies, therefore, have also been carried out on using

distributed environments to protect data privacy (Omar et al., 2019; Bosri et al., 2020; Lin et al., 2021). For example, Bosri et al. (2020) introduce a privacy-preserving recommendation system, *Private-Rec*, by integrating artificial intelligence strategies with blockchain technology. This system provides a secure environment for users by utilizing distributed features where users' data is employed with their permissions. Another recent study introduces a blockchain-based recommender system platform that prioritizes privacy preservation (Omar et al., 2019). The decentralized nature of blockchain allows clients to control their data and consent to its use in the analysis. This platform incentivizes clients to share their information by offering rewards such as points or discounts, which can be used for relevant recommendations provided by the online company. Finally, Lin et al. (2021) develop a blockchain-based recommendation system using local sensitive hashing and differential privacy techniques in the blockchain environment. Thus, the cost of prediction calculations decreased while the quality of the recommendation and the level of user privacy increased.

Popularity bias issue of CF recommenders

One of the most critical topics in RSs is how fairly the system acts for two essential stakeholders, i.e., users and item/service providers, without propagating any bias in provided referrals (Abdollahpouri and Burke, 2019; Yalcin, 2022a). However, recent studies explore that recommendation algorithms mostly fail to handle items in their produced ranked lists fairly due to several reasons, such as their internal mechanisms or characteristics of data where they are trained (Chen et al., 2020).

One well-known example of such an issue is the popularity bias problem, which leads to achieving recommendation lists dominated mostly by popular items, even if they would not be desired extremely (Abdollahpouri et al., 2020; Yalcin and Bilge, 2022). This problem, unfortunately, causes significant decreases in the beyond-accuracy quality of the recommendations, such as diversity, catalogue coverage, or novelty, and worse weakens the robustness of the system against some malicious attacks. Therefore, there has been a rising interest in scrutinizing its adverse effects on recommendations and developing practical popularity-debiasing methods to achieve more appropriate recommendations (Abdollahpouri et al., 2019a; Yalcin and Bilge, 2021; Boratto et al., 2021).

The initial research on this problem has mainly discovered its effects on recommendations for different application domains such as movies, music, travelling, and online courses. For example, Jannach et al. (2015) consider several CF recommenders with different categories, such as matrix factorization- and neighbourhood-based and observe that they intrinsically feature a small number of popular items repeatedly in their recommendations. They also analyze how finetuning parameters of such CF algorithms affect their bias toward popular items. Boratto et al. (2019) perform a similar analysis for massive open online courses by considering the same set of CF recommenders to analyze how they are biased towards the popularity of courses and their categories. Another pioneering study (Abdollahpouri et al., 2020) analyses how the effects of such an issue differ for users and item suppliers.

The literature also considers how the recommenders' popularity bias unfairly affects individuals with different interest levels in item popularity. For example, in movie recommendations, Abdollahpouri et al. (2019b) observe that this problem is more harmful to users interested in popular items in their original profiles when compared to selective ones who mostly rate niche ones. Such an analysis is later re-performed for music recommendations with a more extensive set of CF recommenders (Kowald et al., 2020). Also, Yalcin and Bilge (2022) classify individuals based on the essential characteristics of their rating behaviour and observe that highly-interacting, selective, and hard-to-predict users receive extremely unfair recommendations, especially in terms of their original propensities on item popularity. Finally, Yalcin and Bilge (2023) have analyzed the effects of popularity bias on users with different personality traits in the big-five-factor model and explored that users avoiding new experiences and individuals who are less extroverted are subjected to more unfair recommendations regarding item popularity.

Researchers have been exploring ways to mitigate the issue of popularity bias in recommender systems. For this purpose, various methods have been developed that can be broadly classified into three categories: pre-processing, in-processing, and post-processing, depending on how they reduce bias in recommendations (Boratto et al., 2021). Pre-processing methods typically aim to balance the distribution of user preferences by altering data used to train recommendation algorithms. One example is an approach that divides items into "head" and "tail" categories based on their popularity and uses different data sets for recommendations for each group (Park and Tuzhilin, 2008). Another pre-processing method introduced by Jannach et al. (2015) construct user-item pairs where popular items are filtered out to

achieve tuples where observed items are less-popular. After that, recommendation algorithms are trained on such constructed tuples. Additionally, a probability distribution function designed considering item popularity can give underappreciated items more prominence in recommended lists (Chen et al., 2018).

Several attempts have also been made to tackle the popularity bias problem in recommender systems, particularly through "in-processing" methods. These methods modify conventional recommendation algorithms to consider both relevance and popularity simultaneously; therefore, they are commonly considered algorithm-specific ungeneralizable solutions. Some prominent examples of these approaches include: (*i*) a method that estimates individual preferences for disfavored items and utilizes this information to penalize popular items during the recommendation process (Kamishima et al., 2014); (*ii*) an optimized variant of the RankALS algorithm that balances intra-list diversity and accuracy (Abdollahpouri et al., 2017); (*iii*) a personalized recommendation framework that estimates familiarity between items based on their popularity and prunes the most preferred ones in order to achieve a more balanced common-neighbor similarity index (Hou et al., 2018a); (*iv*) a debiasing approach that minimizes the correlation between user-item relevance and item popularity, which helps to treat items in the "long-tail" equally (Boratto et al., 2021); (*v*) and a recommendation method based on variational autoencoders that penalizes scores given to items based on their historical popularity to increase diversity in the recommendation results (Borges and Stefanidis, 2021).

On the other hand, post-processing methods for this issue can be applied to the output of any recommendation algorithm and involve re-ranking or creating a new list of recommendations that follows a specific constraint. One approach, proposed by Abdollahpouri et al. (2018), involves calculating final scores by weighting the predicted ratings inversely proportional to the popularity of items and then using these scores to re-rank the recommended list. Similarly, Yalcin and Bilge (2021) propose two methods for popularity-debiasing in recommendations, multiplicative and augmentative, specifically tailored for groups of users rather than individuals. Both are based on a re-ranking strategy of items but differ in how they penalize items based on their popularity. More specifically, the multiplicative approach attempts to reduce the average popularity of the recommended items by using the item weights as a factor when computing the ranking scores. On the other hand, the augmentative approach takes prediction scores as the primary influence and adds the item weights as a factor to the ranking scores to achieve a reasonable level of accuracy in the final recommendations. Another renowned approach, presented by Abdollahpouri et al. (2019a), is the xQuad re-ranking approach, which balances the trade-off between ranking accuracy and the coverage of long-tail items.

As can be followed by the presented two sections, there is a rising interest in the RSs literature to protect individuals' privacy and counteract the popularity bias of recommendation algorithms. However, to the best of our knowledge, there is no research analyzing how the bias propagation of CF recommenders differentiates when they are empowered with such privacy-protection mechanisms. In addition, it is required to perform more analyses of whether popularity-debiasing methods can still show similar performances when traditional CF recommenders are decorated with such privacy-preserving strategies.

CONSTRUCTING USER PERSONAS BASED ON VARYING PRIVACY PROTECTION REQUIREMENTS

Collecting accurate data from individuals is crucial for CF methods to provide qualified referrals. However, due to confidentiality issues, individuals usually avoid sharing their actual tastes for items/products, making it difficult to model them correctly, thus leading to poor referrals. Therefore, producing accurate recommendations by enabling users' privacy has become one of the most important topics in RSs in recent years (Ozturk and Polat, 2015; Pramod, 2022).

PPCF approaches introduced towards such aim mainly attempt to conceal (*i*) the list of items truly rated by users and (*ii*) the genuine values of the ratings provided for them. The first can guard users against malicious attacks, such as intending unsolicited marketing, while the second can protect them from potential price or profiling discrimination attempts. As one of the most common and effective approaches to preserving users' privacy, RPTs aim to randomly inject fake votes into some empty cells of the users' profiles and randomly perturb each genuine rating in their profiles Polat and Du (2003); Yargic and Bilge (2019). Thus, by obstructing the exposure of actual data of individuals without endangering the utility of preference data, these approaches are highly beneficial in providing privacy at a satisfactory level yet not sacrificing much from accuracy.

More concretely, these techniques suggest concealing a vote entry v by changing it with $v + r$, where r is a random value drawn from either a Gaussian or uniform distribution. The range of generated random values is adjusted by selected σ_u parameter, where $\sigma_u = [0, \sigma_{max}]$ and σ_{max} is determined by the central server to regulate the distortion level of actual user preferences. Suppose that α is a constant and equals $\sqrt{3}\sigma_u$; the random values are drawn from either $\mathscr{U}(-\alpha, +\alpha)$ or $\mathscr{N}(0, \sigma_u)$ distributions. Moreover, RPTs inject random values as fake votes to user profiles' arbitrarily selected empty cells. Here, the total number of empty cells to be injected is adjusted based on the selected β_u parameter, where $\beta_u = [0, \beta_{max})$ and β_{max} is determined by the central server relating to the density ratio of the entire user-item rating matrix.

Algorithm 1 Randomized perturbation procedure for data concealing

---

    Input: User rating vector $(u_{1 \times n})$, σ_{max}, and β_{max}
    Calculating z-scores $(\to Z)$:
  1: **for all items** in u $(j \leftarrow 1 \text{ to } n)$ **do**
  2:     $z_j = [u_j - \text{MEAN}(u)]/\text{STD}(u)$
  3: **end for**
    Adjusting privacy parameters:
  4: $\sigma_u \leftarrow \text{RAND}(0, \sigma_{max})$, $\beta_u \leftarrow \text{RAND}(0, \beta_{max})$, and $\alpha \leftarrow \sqrt{3}\sigma_u$
  5: $e \leftarrow \text{\# of empty cells}$, $g \leftarrow \text{\# of genuine ratings}$
  6: $F \leftarrow e \times \beta_u\%$
    Selecting dispersion:
  7: $dist \leftarrow \text{RAND}(\text{Gaussian, uniform})$
    Creating random values:
  8: **if** $dist$ is Gaussian:
  9:     $R \leftarrow \mathscr{N}(g + F; \mu = 0, \sigma_u)$
10: **else:**
11:     $R \leftarrow \mathscr{U}(g + F; \alpha)$
12: **end if**
    Disguising z-scores $(\to Z')$:
13: **for all items** in u $(j \leftarrow 1 \text{ to } n)$ **do**
14:     $z'_j = (z_j + R_j)$
15: **end for**
    return Z'

In these settings, each user tunes their own σ_u and β_u values based on desired privacy level and then masks their profiles, as described in Algorithm 1. Afterwards, they share their disguised profiles instead of the original ones with the service provider. Thus, a disguised preference matrix, i.e., $R'_{m \times n}$, where m and n are the total numbers of users and items, respectively, is achieved on which the recommendation generation process will be performed via any proper CF recommender.

To comprehensively analyze how users with varying degrees of privacy are differently affected by the popularity bias issue in recommendations, we define four types of users according to their desired confidentiality protection level by applying Algorithm 1. The first, *Utterly Unconcerned (UU)*, indicates individuals of the standard non-private CF approaches. These users are as completely reckless about their privacy and expect to receive recommendations with the highest possible accuracy. The second, *Barely Bothered (BB)*, refers to users who prefer low-level privacy protection and do not get anxious if their privacy is slightly violated. The third, *Pretty Pragmatists (PP)*, indicate individuals who pragmatically balance privacy and accuracy; therefore, they desire to receive decent recommendations as long as their privacy is not violated. Finally, the fourth, *Fiery Fundamentalists (FF)*, refers to users who are hypersensitive about their privacy and never act as troublemakers regarding accuracy performance.

When guaranteeing the privacy of users by applying the *RPTs*, the selected values of σ_{max} and β_{max} are crucial for them in adjusting their desired privacy level. Remember that higher (resp. lower) σ_{max} and β_{max} values refer to more (resp. less) advanced privacy protection. Therefore, we construct these four user personas by considering the parameter tunings given in Table 1 when adopting the *RPTs* and examine each separately to observe the adverse effects for each privacy level-based user personas more clearly.

Although provided privacy levels are intuitively observed in Table 1, we also present a methodology to quantify the concrete level of privacy obtained by β_{max} and σ_{max} parameters. For this purpose, we

User Persona		σ_{max}	β_{max}	Dispersion	Privacy-level
Utterly Unconcerned	*(UU)*	0	0	N/A	No privacy
Barely Bothered	*(BB)*	2	5	Uniform/Gaussian	Low-level
Pretty Pragmatists	*(PP)*	3	10	Uniform/Gaussian	Mid-level
Fiery Fundamentalists	*(FF)*	4	25	Uniform/Gaussian	High-level

Table 1. Parameters of the constructed privacy-based user personas

perform separate sets of analyses for privacy quantification provided by these two parameters.

According to Algorithm 1, each disguised user profile contains genuine and fake ratings, filling $\beta_{max}\%$ of the empty cells. Therefore, privacy leakage by the server can only be realized by first estimating the exact set of genuine items with some uncertainty. Intuitively, such uncertainty can be quantified by estimating the probability of any randomly selected rating being genuine. Let $\mathbf{a} = \{a_1, a_2, \cdots, p_a\}$ denotes the set of genuine ratings and $\mathbf{f} = \{f_1, f_2, \cdots, f_{e \times \beta_{max}}\}$ defines the set of inserted fake ratings. Then, the probability of an arbitrarily selected rating, r, to be genuine is $\Pr(r \in \mathbf{a})$ can be estimated as $\frac{|\mathbf{a}|}{|\mathbf{a}| + |\mathbf{f}|}$. Note that such probability of being lower means a better level of provided privacy protection, and vice versa.

We present histograms of estimated probabilities for three personas in three datasets (i.e., MovieLens-1M (ML), Douban Book (DB), and Yelp) in Fig. 2. Note that the details of these datasets are also given in the following section. As expected, *FF* users mainly locate in the lowest probability bin in histograms, while *PP* and *BB* users tend to distribute into higher probability bins, indicating that personas obtain a declining level of confidentiality from *FF* to *BB*. Note that as the ratio of the number of ratings per user to the number of items in the dataset decreases in datasets, the effect of β_{max} on privacy level scales up, as can be observed in larger datasets such as Yelp and DB.
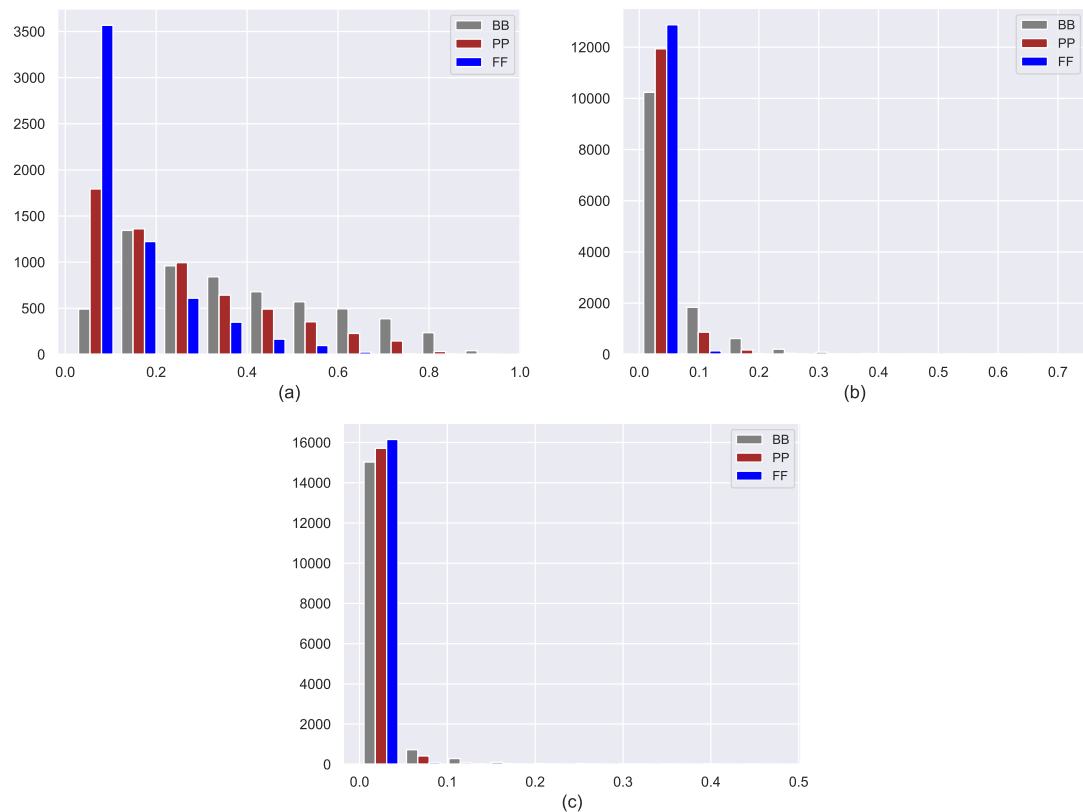
Figure 2. Histograms of estimated probabilities for three personas in the (a) ML, (b) DB, and (c) Yelp datasets.

In the context of rating distribution, it is crucial to measure the privacy obtained by adding random

noise to distinguish authentic votes from fake ones. Although central servers can differentiate between them, extracting genuine values from their perturbed z-score forms is necessary. To quantify the privacy of an additive noise-based perturbed variable, Agrawal and Aggarwal (2001) propose a metric based on differential entropy. This metric is applied in PPCF context by (Polat and Du, 2005b; Bilge and Polat, 2013). In this context, the original user vector is represented by P, and the perturbing random data is represented by R, yielding $U = P + R$. The *average conditional privacy* of P is defined as $\Pi(P|U) = 2^{H(P|U)}$, where $2^{H(P|U)}$ is the *conditional differential entropy* of P given U. Since P and R are independent random variables, the privacy level of P after disclosing U can be calculated as $\Pi(P|U) = \Pi(P) \times (1 - Pr(P|U))$, where $Pr(P|U) = 1 - 2^{H(U|P) - H(U)}$. Assuming a normal distribution for P, the privacy levels, $\Pi(P|U)$, for various perturbation levels are shown in Table 2. The output shows that the privacy levels increase with increasing perturbation of profiles, as expected.

Dataset	User Personas		
	BB	*PP*	*FF*
ML	2.644	2.883	3.021
DB	1.864	1.985	2.051
Yelp	1.464	1.540	1.579

Table 2. Privacy levels $(\Pi(P|U))$ for different personas

EXPERIMENTAL SETUP

In this section, we introduce the experimental settings, including utilized datasets, recommendation algorithms, and evaluation protocols, followed by experimentation methodology, and considered treatment methods for the popularity bias issue.

Datasets

In the experiments, we use three publicly available real-world benchmark datasets collected for different application domains, as in a very related recent study (Yalcin and Bilge, 2022); MovieLens-1M[7] (ML) for the movies (Harper and Konstan, 2015), Douban Book[8] (DB) for the books (Shi et al., 2018), and Yelp[9] for the local business reviews (Shi et al., 2018). User preferences in all utilized datasets are discrete and in five-star rating scale. Table 3 presents detailed information about ML, DB, and Yelp datasets. Also, they have varying dimensions and sparsity ratios, as can be followed in Table 3, enabling us to analyze how such two important aspects of the datasets are related to potential unfairness issues for user personas with different privacy levels.

Dataset	# of Users	# of Items	# of Ratings	Sparsity (%)	Rating Average	# of Ratings per Item	# of Ratings per User
ML	6,040	3,952	1,000,209	95.8	3.58	253.1	165.6
DB	13,024	22,347	792,062	99.7	4.05	35.4	60.8
Yelp	16,239	14,284	198,397	99.9	3.77	13.9	12.2

Table 3. Properties of MovieLens-1M (ML), Douban Book (DB), and Yelp datasets

Utilized recommendation algorithms

It is very important to enrich the experiments with different parameters to analyze better the potential unfairness issues for individuals preferring varying privacy levels. The most convenient way to achieve it is to employ different algorithms in the recommendation generation phase. Indeed, this process can be described as generating recommendation lists using different algorithms after performing data perturbation and obfuscation operations to see how the privacy protection measures affect each algorithm's bias towards popularity.

---

[7]https://grouplens.org/datasets/movielens/1m/
[8]https://www.douban.com/
[9]https://www.yelp.com/

408 Therefore, ten different algorithms, two non-personalized and eight state-of-the-art approaches of
409 four different CF families (i.e., matrix factorization-based, probabilistic, clustering-based, and neural
410 networks-based), have been determined for recommendation purposes, as detailed in Table 4. Two
411 non-personalized methods, i.e., *ItemAverage (IA)* and *MostPopular (MP)*, are selected due to their wide
412 usage in the literature to provide straightforward recommendations (Boratto et al., 2019; Yalcin and Bilge,
413 2022). More specifically, the *IA* recommends items with the highest average rating, while the *MP* sorts
414 items in descending order based on the number of their received ratings and then considers the most
415 popular ones as recommendable. Therefore, by its nature, *MP* produces the same recommendation lists for
416 each individual and denotes the maximum level of popularity bias that would be propagated. On the other
417 hand, the remaining eight CF algorithms are selected as the best-performing ones from different families
418 and applied via the Python-based library called Cornac[10] (Salah et al., 2020). Also, for reproducibility
419 purposes, we left hyperparameters of the algorithms set to their default values. Note that since the primary
420 motivation of the present study is to evaluate whether recommendation algorithms propagate disparities
421 among privacy level-based user personas, we do not provide detailed information about these algorithms'
422 internal mechanisms for clarity.

No	Family	Recommendation Algorithm	Abbreviation
1	Non-personalized	Most Popular	MP
2	Non-personalized	Item Average	IA
3	Matrix factorization-based	Maximum Margin Matrix Factorization (Weimer et al., 2008)	MMMF
4	Matrix factorization-based	Weighted Matrix Factorization (Hu et al., 2008)	WMF
5	Matrix factorization-based	Hierarchical Poisson Factorization (Gopalan et al., 2015)	HPF
6	Probabilistic	Weighted Bayesian Personalized Ranking (Gantner et al., 2012)	WBPR
7	Probabilistic	Indexable Bayesian Personalized Ranking (Le and Lauw, 2017)	IBPR
8	Clustering-based	Spherical *k*-means (Salah et al., 2016)	SKM
9	Neural networks-based	Neural Matrix Factorization (He et al., 2017)	NEUMF
10	Neural networks-based	Variational Autoencoder for CF (Liang et al., 2018)	VAECF

Table 4. Detailed information about utilized recommendation algorithms

Evaluation metrics

424 Since the study aims to measure the effect of perturbation and obfuscation processes on popularity
425 bias in different aspects, nine essential evaluation metrics are considered through the experiments. One
426 of them, i.e., *Average Percentage of Recommended Items (APRI)*, is introduced by Yalcin (2022b) to
427 measure the level of popularity of recommended items solely. The remaining metrics can be grouped
428 under two main categories, i.e., accuracy and beyond-accuracy. More specifically, we employ the well-
429 known *nDCG*, *Precision*, *Recall*, and *F1-score* metrics for measuring accuracy performances, while
430 *Average Percentage of Long-tail Items (APLT)*, *Novelty*, *Long-tail Coverage (LTC)*, and *Entropy* for
431 assessing beyond-accuracy recommendation quality. Here, although the selected beyond-accuracy metrics
432 focus on measuring different aspects of the provided recommendations, such as popularity, diversity,
433 coverage, novelty, and inequality, all of them indeed associate with the well-known fairness concept in
434 RSs literature from the item perspective (Abdollahpouri and Burke, 2019). We describe these metrics
435 below in detail, where $i_1, i_2, \cdots, i_N$ denotes the items in the top-N list for user u, i.e., N_u, demonstrated by
436 a recommendation algorithm.

437 ***Average Popularity of Recommended Items (APRI):*** This metric calculates the average popularity
438   of recommended items by the algorithms; therefore, higher *APRI* values for top-N lists indicate
439   that the recommendation algorithm propagates a high-level of undesirable popularity bias (Yalcin,
440   2022b). More concretely, this metric first calculates the popularity ratio of each recommended item
441   i, i.e., P_i, by considering the ratio of the total number of users who rate the corresponding item to
442   the number of all users in the system. Then, it computes the average of the P values of the items in
443   the Top-N list to achieve an $APRI_u$ score for the corresponding user u, as formulated in Eq. 1.

$$APRI_u = \frac{\sum_{i \in N_u} P_i}{N} \qquad (1)$$

[10]https://cornac.preferred.ai/

***Precision, Recall, and F1-score*:** To measure the accuracy quality of produced top-*N* recommendations, we employ well-known *Precision*, *Recall*, and *F1-score* metrics (Yalcin and Bilge, 2022). More specifically, the *Precision* score of a top-*N* list (i.e., $P@N_u$) for a user *u* is calculated as the percentage of the recommended *N* items suitable for the user. On the other hand, the *Recall* score of the corresponding recommendation list (i.e., $R@N_u$) is computed as the ratio of the number of suggested suitable items to the number of all rated items in *u*'s profile. Here, when determining whether a recommended item is suitable, we set the threshold value as 3.5, as the literature has reached a consensus that 4 and 5 are the affirmative ratings in a 5-star rating scale (Bobadilla et al., 2013). Finally, *F1-score* of the considered top-*N* list (i.e., $F1@N_u$) is calculated as the harmonic mean of its $P@N_u$ and $R@N_u$ scores, as formulated in Eq. 2; thus, it provides a balance between such two metrics measuring accuracy level in different aspects.

$$F1@N_u = 2 \times \frac{P@N_u \times R@N_u}{P@N_u + R@N_u} \tag{2}$$

***Normalized Discounted Cumulative Gain (nDCG)*:** Another important accuracy metric used in the experiments is the *nDCG*, which considers the actual ratings of the recommended items and their positions in the produced top-*N* list (Yalcin and Bilge, 2022). Supposing that $r_{u,i}$ is *u*'s actual rating provided for item *i*, the $DCG^u_{N_u}$ and $nDCG^u_{N_u}$ values of the top-*N* recommendation list of *u* is calculated using Eqs. 3 and 4, respectively.

$$\text{DCG}^u_{N_u} = r_{u,i_1} + \sum_{n=2}^{N_u} \frac{r_{u,i_n}}{log_2(n)} \tag{3}$$

$$n\text{DCG}^u_{N_u} = \frac{\text{DCG}^u_{N_u}}{\text{IDCG}^u_{N_u}} \tag{4}$$

where $IDCG^u_{N_u}$ value is the maximum possible gain for *u* and is obtained by ideally ordering *N* items.

***Average Percentage of Long Tail Items (APLT)*:** Another metric used in the experiments to evaluate the quality of ranking-based recommendations is the *APLT*, measuring what percentage of the recommended items are from the long-tail part of the popularity curve (Abdollahpouri, 2020). In other words, it demonstrates the ability of the recommendation algorithms to suggest niche items. When categorizing items as *tail* or *head*, the most commonly utilized approach is the Pareto Principle (Sanders, 1987), labelling items that receive 20% of the total ratings as *head* and the remaining ones in the catalogue as the *tail* items. Suppose that *T* is the set of assigned *tail* items via this principle, the *APLT* score of a top-*N* recommendation list produced for user *u* is calculated as given in Eq. 5.

$$APLT_u = \frac{|\{i, i \in (T \cap N_u)\}|}{N} \tag{5}$$

Note that although $APLT_u$ reflects the diversification level belonging to a particular user. However, the overall *APLT* score is an average of such diversification levels, and the higher *APLT* scores do not necessarily mean that the considered algorithm can provide high-quality referrals in terms of diversity (Abdollahpouri et al., 2021). Nevertheless, lower *APLT* scores indicate a general lack of diversity in recommendation lists. Therefore, we also utilize *Entropy* as an alternative metric to measure the diversification level of all recommendation lists, which is explained in detail in the following.

Entropy: This metric mainly measures how often the recommendation algorithms suggest each item in the catalogue (Elahi et al., 2021). In other words, it reflects the inequality level across the frequency distribution of the suggested items. Therefore, higher *Entropy* values for algorithms indicate more homogeneity in frequencies and diversity in recommendations, which is more desirable, especially for system administration, to enable fairer competition in the market. More concretely, the produced top-*N* list for each individual is merged by including the duplicated items. Suppose that such constructed union of each top-*N* list is represented with N_{all} and the set of all items in the catalogue $\{i_1, i_2, \ldots, i_K\}$ is denoted with *K*, then the *Entropy* value of a recommendation algorithm is computed using the formula given in Eq. 6.

$$Entropy = -\sum_{i \in K}(\Pr(i))log_2(\Pr(i)) \tag{6}$$

478   where $\Pr(i)$ denotes the relative frequency of item *i* appearing in N_{all} set.

479   ***Novelty***: As a prominent beyond-accuracy metric, we also use the *Novelty* in the experiments, which can
480   be considered as the ability of the recommendation algorithms to present items that users did not
481   taste before (Yalcin and Bilge, 2022). In other words, it measures what percentage of a top-*N* list
482   produced for a user *u* has not been previously rated; thus, the capability of the algorithms to provide
483   novel alternatives to users. Suppose that I_u is the set of items rated by user *u*, the *Novelty* score of
484   top-*N* recommendations provided for the corresponding user, i.e., $Novelty_u$, can be computed using
485   Eq. 7.

$$Novelty_u = \frac{|\{i, i \in N_u \text{ and } i \cap I_u = \emptyset\}|}{N} \tag{7}$$

486   ***Long-Tail Coverage (LTC)***: The last employed metric in the experiments is the *LTC* measuring how
487   much the provided recommendations cover the tail part of the item catalogue (Abdollahpouri, 2020).
488   We first construct a non-recurring aggregation of top-*N* lists produced for each user, which we refer
489   to as $\mathbb{N}$. Contrary to the *Entropy* calculation, the duplicated items in the individual top-*N* lists are
490   eliminated here. Then, we determine an intersection item set, i.e., $I_{\mathbb{N} \cap T}$, by considering items that
491   appear both in $\mathbb{N}$ and *T* simultaneously, where the latter is the tail item set determined via Pareto
492   Principle (Sanders, 1987). Finally, we calculate the ratio of the size of $I_{\mathbb{N} \cap T}$ to the size of *T*, as in
493   Eq. 8. Therefore, higher *LTC* values for algorithms indicate better coverage of the unpopular, i.e.,
494   *tail*, items in the recommendations.

$$LTC = \frac{|I_{\mathbb{N} \cap T}|}{|T|} \tag{8}$$

495   Note that higher values of described metrics except for *APRI* indicate less-biased and more qualified
496   recommendations in terms of accuracy and beyond-accuracy perspectives.

Experimentation methodology

498   In the experiments, we consider four scenarios described in Table 1, where users differ based on their
499   desired privacy-protection level. Note that all experiments detailed in the following are realized for such
500   four scenarios separately.
501   For each user profile scenario, we generate top-10 recommendations for users by using one of the
502   previously explained algorithms. In doing so, we apply the well-known leave-one-out experimentation
503   strategy (Kearns and Ron, 1997). Accordingly, we use a particular user as test data and the remaining
504   ones as train data and compute predictions for all items for the user by running the considered algorithm
505   on the train set. This process is performed repeatedly for each user in the dataset. Then, we select top-10
506   items for each user as the recommendation list based on their computed predictions in descending order.
507   To comprehensively evaluate how users concerned with varying privacy levels are differently affected
508   by the recommendations, we compute *Precision*, *Recall*, *F1-score*, *nDCG*, *APRI*, and *Novelty* value for
509   the top-10 recommendation of each user and then calculate their averages to achieve final results. On

510 the other hand, when computing the *LTC* and *Entropy* performance of the algorithms, we consider the
511 aggregation of all top-10 recommendations of users, as these metrics are not user-centric and operate on
512 the set of all recommendations as a whole. Note that when reporting the experimental findings in the
513 following section, we consider average values of 100 trials realized for each user-profile scenario to avoid
514 potential inconsistencies during the randomization process of the perturbation and obfuscation phases.

The benchmark treatment methods for popularity bias problem

516 One of the main motivations of this study is to analyze how traditional popularity-debiasing methods'
517 performance varies when applied to disguised user profiles. To this end, we consider three benchmark
518 post-processing treatment methods for the popularity bias issue concerning the recommendations' different
519 quality aspects. All methods aim to re-rank the recommendations by penalizing popular items and utilize
520 a regularization parameter (i.e., λ) that weights predictive accuracy and one important beyond-accuracy
521 perspective, such as diversity or coverage.

522   Two utilized popularity-debiasing methods are variants of the *Enhanced Re-ranking Procedure*
523 proposed by Yalcin and Bilge (2021): *Augmentative (Aug)* and *Multiplicative (Mul)*. Both approaches
524 re-sort the items in the recommendation lists based on the synthetic ranking scores, which are estimated
525 for items by inversely weighting their produced predictions and popularity ratios. However, they follow
526 different mechanisms in producing such scores, i.e., *Mul* harshly penalizes popular items by weighting
527 prediction scores inversely with the popularity ratio of items in a multiplicative way, and *Aug* uses
528 prediction scores as a driving force by utilizing item popularities as minor factors. Thus, due to its
529 design, the former can considerably improve beyond-accuracy quality, which might significantly decrease
530 accuracy performance. On the other hand, the latter gives more importance to predictive accuracy while
531 combating popularity bias issues as much as possible.

532   The third utilized algorithm is the famous *xQuad (xQd)* algorithm (Abdollahpouri et al., 2019a), which
533 focuses on achieving final recommendation lists for users by dynamically selecting items based on their
534 categories: *(i) head* refers to the popular items, and *(ii) tail* refers to the unpopular (i.e., niche) items.
535 The algorithm first selects a candidate item set by only considering the prediction scores produced by
536 a recommendation algorithm. Then, it dynamically constructs the final recommendation lists based on
537 the ranking scores of candidate items. Such scores are calculated at each iteration by weighting their
538 prediction scores. The weighting process considers both the original propensities of users on the category
539 of the corresponding item (i.e., *head* or *tail*) and the cover ratio of the current recommendation list for
540 these categories.

ANALYZING ROBUSTNESS OF PRIVACY-PRESERVING COLLABORATIVE RECOMMENDERS AGAINST POPULARITY BIAS ISSUE

543 The first set of conducted experiments aims to observe how much the popularity bias of the recommenders
544 affects recommendation quality for the base case, i.e., *UU*, and three scenarios of privacy protection for
545 users, i.e., *BB*, *PP*, and *FF*. In other words, we measure how users with varying privacy concerns are
546 differently exposed to the adverse effects of bias propagation of the algorithms. To this end, we perform
547 several experiments by considering ten different recommendation algorithms (i.e., *MP*, *IA*, *MMMF*, *WMF*,
548 *HPF*, *IBPR*, *WBPR*, *SKM*, *NEUMF*, and *VAECF*) and three datasets (i.e., ML, DB, and Yelp), which are
549 explained above, and compute the obtained results for nine considered metrics for four privacy level-based
550 scenarios. Accordingly, Figs. [3 - 9] present the obtained beyond-accuracy, i.e., *APRI*, *APLT*, *Entropy*,
551 *Novelty*, and accuracy, i.e., *Precision*, *Recall*, *F1-score* and *nDCG* results for three datasets, respectively.

552   As can be followed by the *y*-axis of Fig. 3, regardless of the utilized algorithm and privacy level-based
553 user personas, the highest *APRI* values are usually obtained for the ML, followed by DB and Yelp datasets,
554 respectively. The main reason for this observation is that the average number of ratings for each item is
555 higher in the ML compared to DB and Yelp datasets, as presented in Table 3. This fact inevitably leads to
556 a higher popularity ratio for items and, thus, higher *APRI* values for the ML dataset. The *MP* algorithm is
557 selected to demonstrate the maximum level of popularity bias propagation in recommendations. Therefore,
558 as expected, the highest popularity bias in the recommendations is obtained with the *MP* algorithm for all
559 settings. In addition, an unexpected finding is that the neural networks-inspired algorithms, i.e., *NEUMF*
560 and *VAECF*, obtain almost identical *APRI* results with the *MP* algorithm, especially in the DB and Yelp
561 datasets. Furthermore, the *SKM* can be considered one of the worst-performing algorithms in terms of
562 *APRI* results for the ML dataset. Therefore, we can conclude that even if they are enhanced personalized
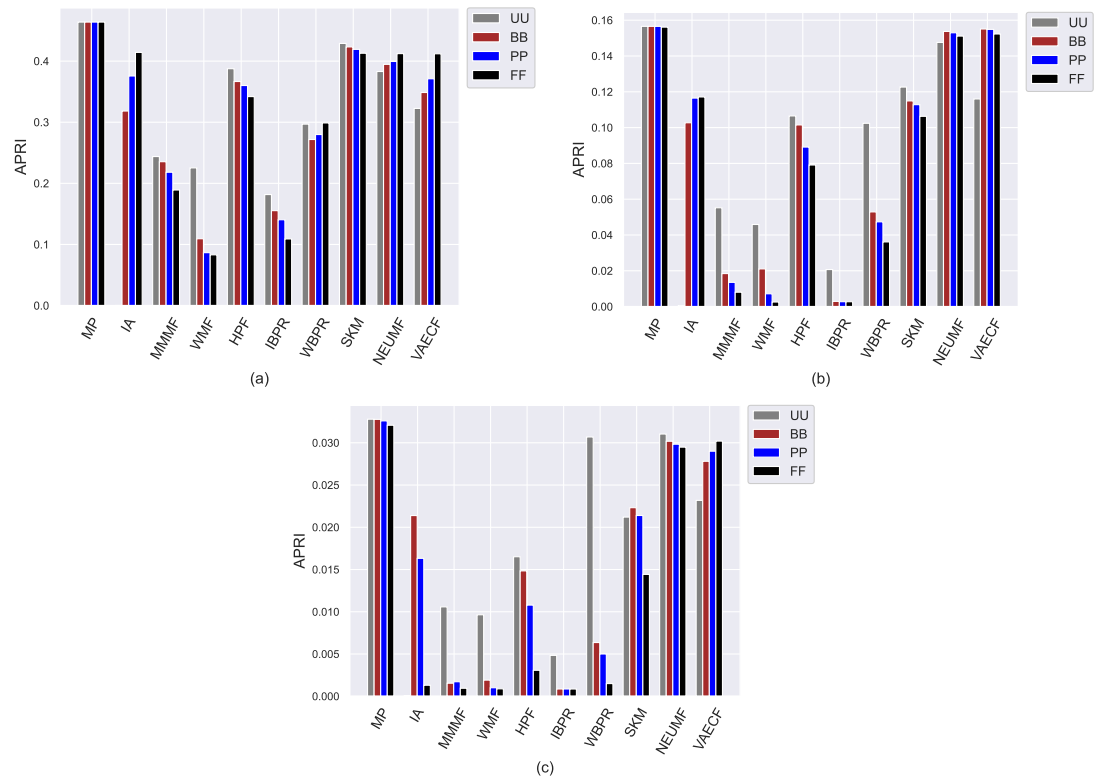
Figure 3. Obtained *APRI* results of the utilized algorithms for four different privacy level-based user profiles on (a) ML, (b) DB, and (c) Yelp datasets.

CF algorithms, they, unfortunately, are biased towards popular items with extreme levels. On the other hand, the lowest *APRI* values are usually obtained with the *IBPR* or *WMF* among the utilized algorithms, especially in sparser and larger DB and Yelp datasets, which demonstrates that they are highly resistant to the popularity bias problem.

The most important insight gained from Fig. 3, on the other hand, is that since the lowest *APRI* results are usually obtained for the *FF* case, followed by *PP*, *BB*, and *UU*, as the privacy level of the users increases, the average popularity of their received recommendations decreases. This finding is also valid for almost all cases, with minor exceptions. Observed differences in the obtained *APRI* results among privacy level-based user personas can also reach significant levels for some settings. For example, for the Yelp dataset, the *UU* is almost five times more than *BB* for the *WBPR* algorithm, or the *PP* is almost four times more than *FF* for the *HPF* algorithm. Therefore, we can conclude that privacy-protection mechanisms for CF algorithms not only eliminate the concerns of the users regarding their privacy but also improve the recommendation quality by alleviating the bias of the algorithms against item popularity. Moreover, such a finding is more concrete for DB and Yelp than the ML dataset; this is an important finding as data collections of real-world applications are typically sparser and larger.

> **Observation 1**: The most common privacy-preserving approaches for recommenders can efficiently eliminate users' concerns regarding their confidentiality and provide notable decreases in the popularity ratio of recommended items simultaneously.

Contrary to the obtained *APRI* values, the achieved *APLT* results vary within the same interval for all utilized datasets, as can be followed by Fig. 4. The reason for this phenomenon is that it is a user-centric metric and calculates the average percentage of tail items in the produced top-10 recommendation lists. Since we have previously observed from achieved *APRI* results that the most vulnerable algorithms against popularity bias are *MP*, *SKM*, *NEUMF*, and *VAECF*, they are not talented in featuring tail (i.e., niche) items in their produced recommendation lists regardless of the considered privacy level-based
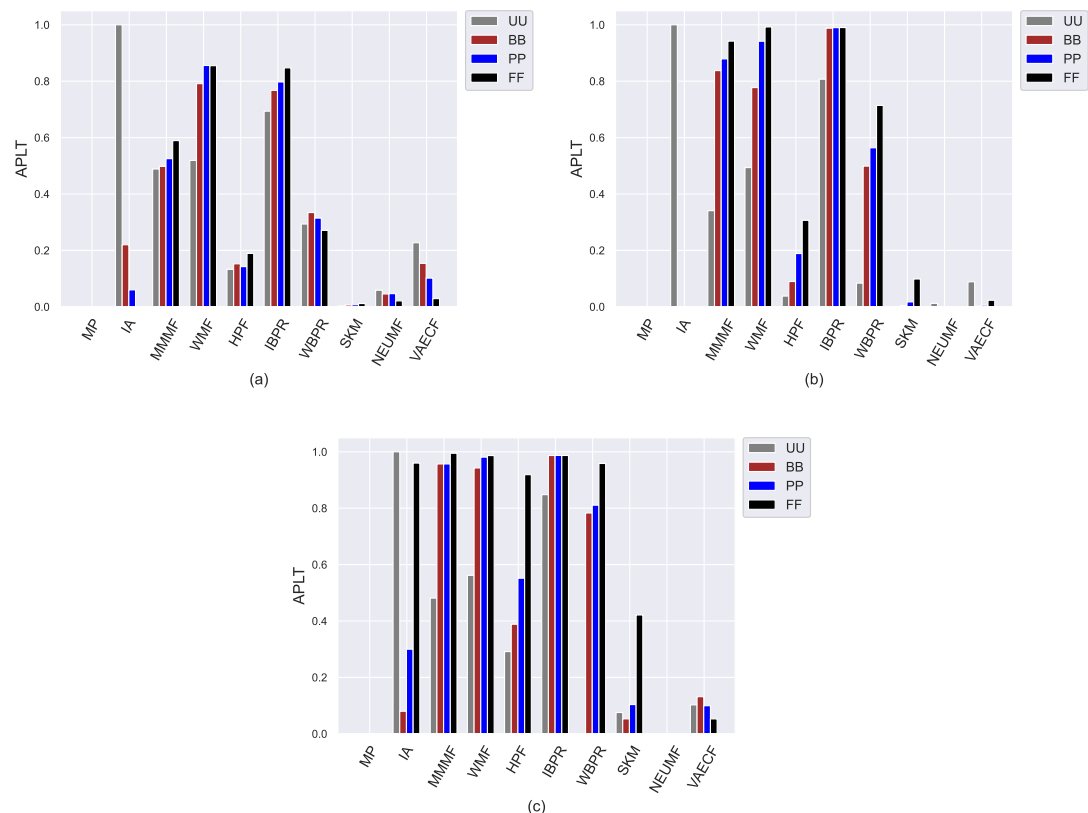
Figure 4. Obtained *APLT* results of the utilized algorithms for four different privacy level-based user profiles on (a) ML, (b) DB, and (c) Yelp datasets.

585 user personas, as expected. On the other hand, the *WMF* and *IBPR* are the most prominent algorithms in
586 providing diverse recommendations, as they achieve the highest *APLT* results. Also, the superiority of
587 these algorithms is more visible for sparser and larger DB and Yelp datasets. The main reason for this
588 observation could be that the size of constructed tail item set for the ML is notably smaller than those
589 formed for these datasets; thus, the chance of including the tail items in the users' recommendation lists
590 would be significantly increased in these two datasets.
591   Another crucial finding observed from Fig. 4 is that except for a few CF approaches (e.g., neural
592 networks-inspired ones), the highest *APLT* scores of the recommendations are usually achieved for *FF*
593 users, followed by *PP*, *BB*, and *UU*. This result reinforces that as the privacy level of users increases,
594 their received recommendations become more qualified in terms of diversity aspect. Additionally, such
595 differences in the obtained *APLT* scores are clearer for Yelp and DB datasets. The main reason for this
596 consequence is that increasing privacy protection of users by including fake ratings in their profiles leads
597 to significant changes in the distribution of the ratings, which makes algorithms more robust against
598 popularity bias, as previously discussed in the *APRI* results. This, in turn, intuitively increases the
599 probability of including tail items in the final recommendations.
600   Similar to the achieved *APLT* values, the obtained *Entropy*, *Novelty*, and *LTC* scores of the algorithms
601 spread out in the same range, i.e., [0, 1], for all datasets regardless of the privacy level-based user profiles
602 by their design, as can be followed by Figs. [5 - 7], respectively. We also observe from Fig. 5 that the
603 lowest information entropy results are usually obtained with *SKM*, *NEUMF*, and *VAECF* algorithms.
604 Therefore, we conclude that neural networks-inspired algorithms are highly unsuccessful in equally
605 representing candidate items in the catalogue in their recommendations.
606   Compared to other algorithms, such weaknesses of the *SKM*, *NEUMF*, and *VAECF* also appear in
607 providing recommendations that are qualified in terms of novelty and catalogue coverage, as seen in
608 Figs. 6 and 7, as they usually have the lowest *Novelty* and *LTC* scores for each considered privacy
609 level-based user personas. Also, since their *LTC* results are generally around zero, we can conclude that
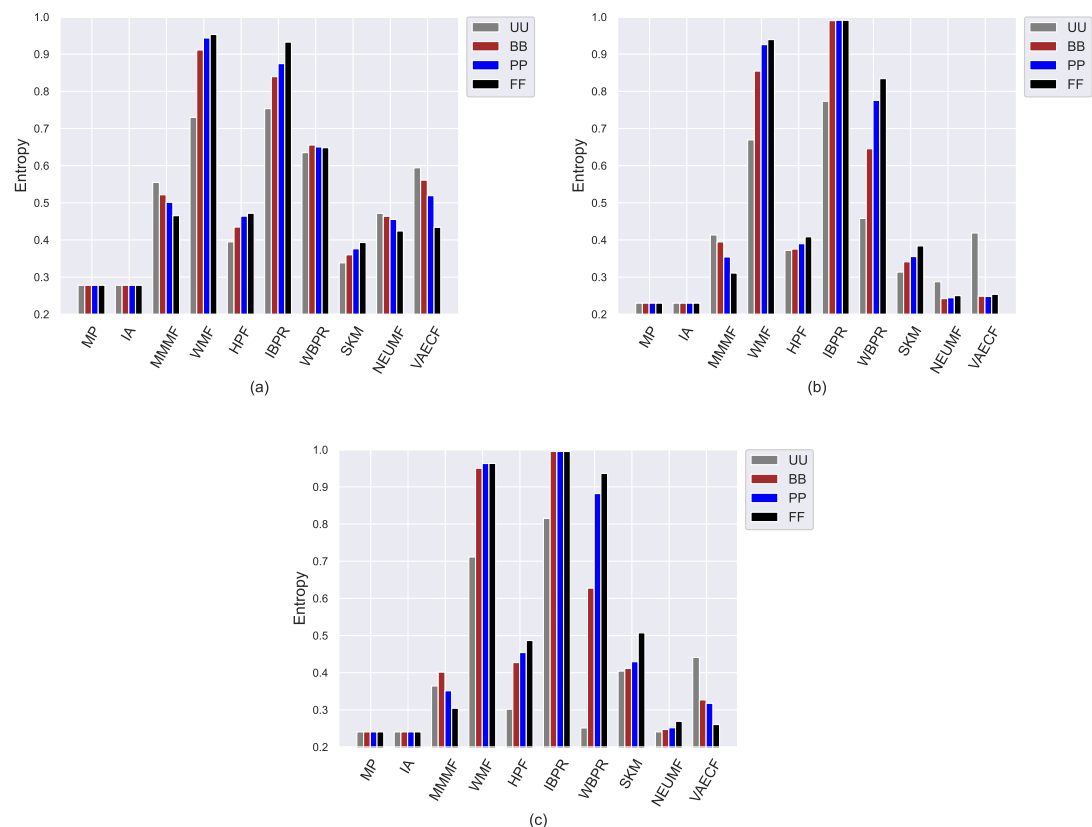
Figure 5. Obtained *Entropy* results of the utilized algorithms for four different privacy level-based user profiles on (a) ML, (b) DB, and (c) Yelp datasets.

these algorithms, including *HPF*, are highly inefficient for item/service providers attempting to sell each product at least once via suggestions covering the whole catalogue. Also, we observe that the *Novelty* scores of all algorithms, including neural networks-based ones, are lesser for ML than other DB and Yelp datasets. The main reason for this result is that the average number of ratings per user is around 166 for ML, while 61 and 12 for DB and Yelp datasets, respectively, as can be followed by Table 3. Such differences in data collections make the recommendation algorithms superior to the DB and Yelp datasets in locating items unknown/inexperienced by the individuals in the recommendations, resulting in higher *Novelty* scores for such sparser and larger datasets.

On the other hand, the *IBPR* and *WMF*, especially the former one, achieve the highest *Entropy*, *Novelty*, and *LTC* outcomes regardless of the considered datasets and privacy level-based user personas, as seen in Figs. [5 - 7]. In other words, the recommendations of these two algorithms are highly qualified in terms of covering and representing items in the whole catalogue fairly and featuring items that individuals have not experienced. Remember from the *APLT* analysis that *IBPR* and *WMF* are the most prominent algorithms in featuring items in the long-tail part of the catalogue in their recommendations, as well.

> **Observation 2**: In producing qualified recommendations in terms of beyond-accuracy aspects, i.e., diversity, novelty, equality, and catalogue coverage, the most successful algorithms are *IBPR* and *WMF*, while the least successful ones are mostly neural networks-inspired algorithms, i.e., *NEUMF* and *VAECF*.

Comparing the privacy level-based user personas, we observe from Fig.5 that the highest entropy values are usually achieved for users highly concerned with their privacy, i.e., *FF* users, followed by *PP* and *BB*, respectively. Therefore, as the privacy-protection level increases for individuals, the likelihood of receiving more qualified recommendations where items are equally represented improves with minor
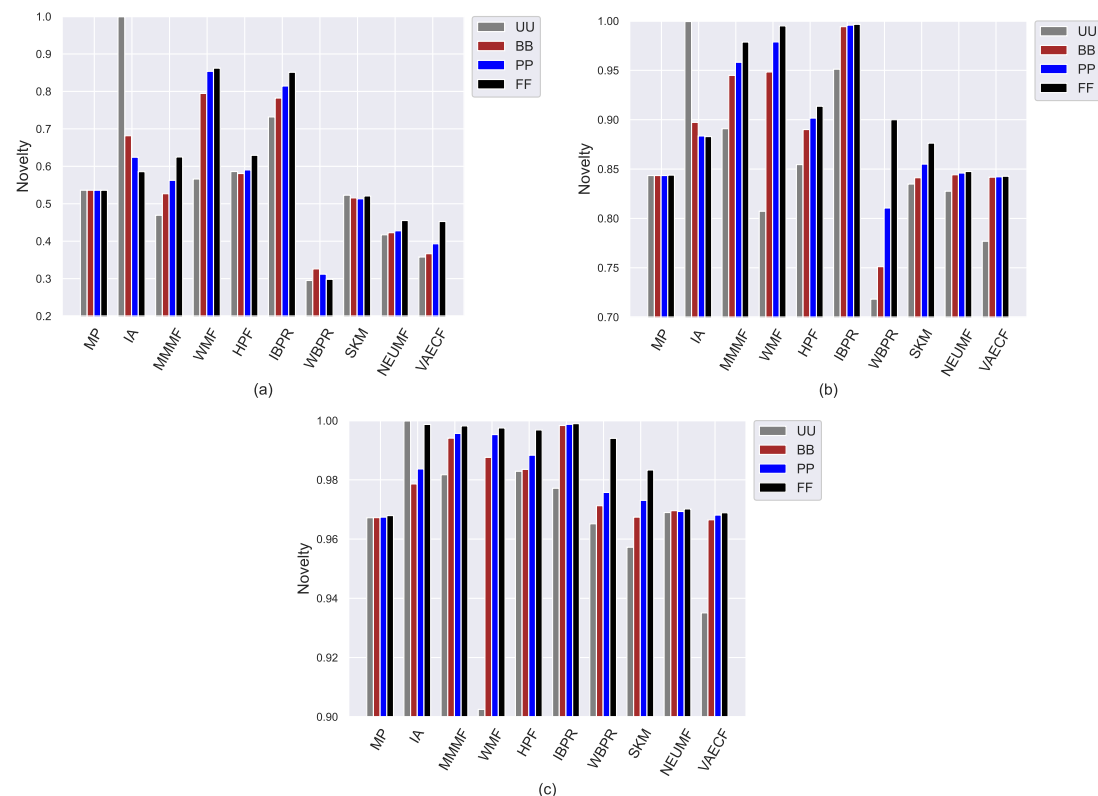
Figure 6. Obtained *Novelty* results of the utilized algorithms for four different privacy level-based user profiles on (a) ML, (b) DB, and (c) Yelp datasets.

exceptions. As can be followed by Fig.6, this advantage for privacy-sensitive users is also valid for obtaining novel recommendations, as the highest *Novelty* scores are again obtained for *FF*, followed by *PP* and *BB*, respectively.

However, the differences in the *Novelty* scores among the privacy level-based user personas seem to be more visible for the ML while limited for sparser and larger DB and Yelp datasets; this occurs due to the same reason related to dataset characteristics explained above. Additionally, even if such differences among privacy level-based user personas are not quite visible for the *LTC* results achieved for DB and Yelp datasets, it can be concluded that the most sensitive users in the ML against privacy protection can receive more qualified referrals in terms of catalogue coverage.

> **Observation 3**: As the level of privacy provided for users improves, their referrals also become more qualified in terms of diversity, novelty, catalogue coverage, and fairness (i.e., equality in representing items).

Ranking accuracy is one of the most important criteria for assessing the quality of provided recommendations. Therefore, we perform an additional set of experiments where we comprehensively analyze how recommendation algorithms perform differently for privacy level-based user personas in terms of accuracy by utilizing four important accuracy-oriented metrics (i.e., *Precision*, *Recall*, *F1-score*, and *nDCG*). To this end, we present the obtained *Precision* and *Recall* scores of ten algorithms for four user profiles in Fig. 8, and *F1-score* and *nDCG* results in Fig. 9.

Due to the well-known trade-off between accuracy and beyond-accuracy recommendation quality, we expect that the algorithms performing well in beyond-accuracy aspects would not provide satisfactory ranking accuracy or vice versa. Such a contradiction is also valid for privacy level-based user personas, i.e., users cannot simultaneously receive recommendations of good quality in both beyond-accuracy and accuracy aspects. The accuracy-focused experimental results validate this phenomenon. For instance, the
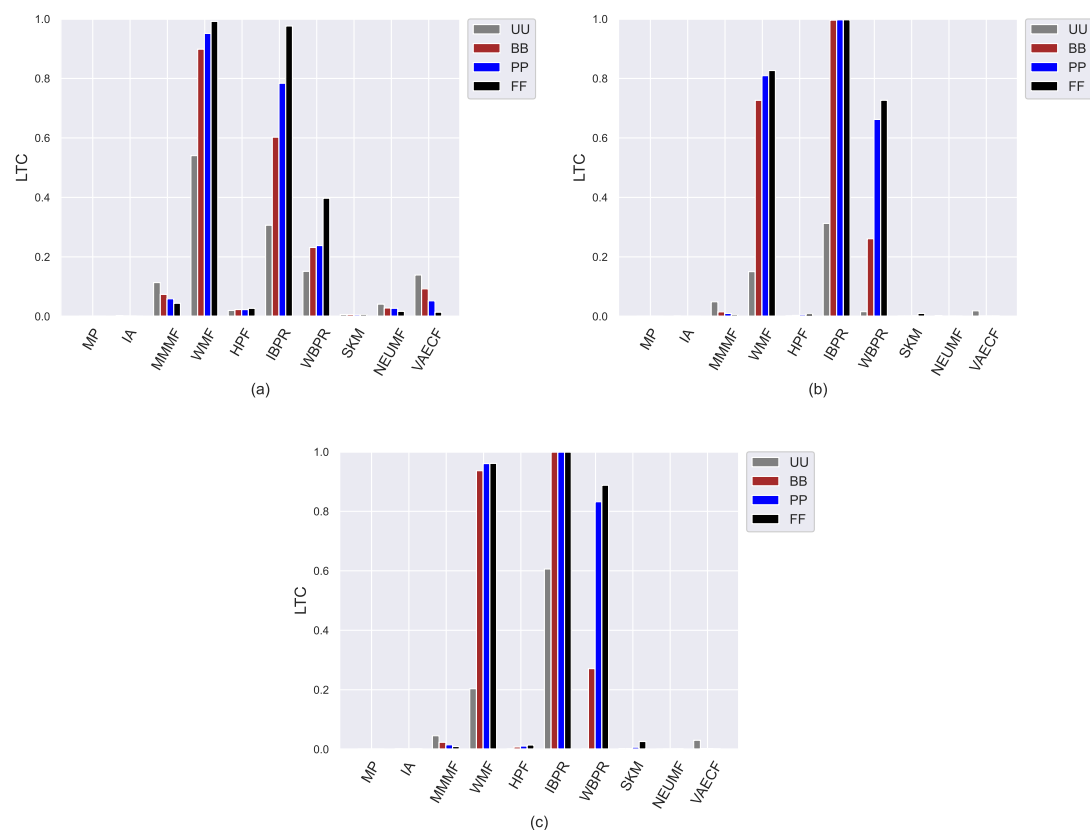
Figure 7. The obtained *LTC* results of the utilized algorithms for four different privacy level-based user personas on (a) ML, (b) DB, and (c) Yelp datasets.

algorithms succeed in beyond-accuracy quality, e.g., the *IBPR*, has usually lower *Precision*, *Recall*, *F1-score*, and *nDCG* results than others, as can be followed by Figs 8 and 9. On the other hand, the algorithms that are not succeeding in beyond-accuracy quality, e.g., neural networks-inspired ones (*NEUMF* and *VAECF*), show relatively better performance than others in providing accurate recommendations. Also, we can conclude that the *WBPR* is another successful algorithm to achieve accurate recommendations. However, the obtained results demonstrate that the *WMF* algorithm shines out among other algorithms in ranking accuracy when utilized without any privacy protection mechanism, i.e., the case of *UU*. This is an interesting finding since we have previously discussed that it is one of the best-performing algorithms regarding beyond-accuracy aspects. Therefore, we can conclude that the *WMF* can be a good option for a traditional recommendation scenario where no privacy protection is applied.

> **Observation 4**: For traditional recommendation scenarios where any level of privacy protection is provided, the *WMF* is the most prominent algorithm considering accuracy and beyond-accuracy aspects simultaneously. Also, the most accurate recommendations are usually obtained by the *VAECF* and *WBPR* algorithms.

As can be followed by Figs. 8 and 9, users receive more accurate recommendations in the *UU* case compared to privacy-preserving scenarios for almost all metrics; this is also valid for all settings except for non-personalized primitive *MP* and *IA* recommendation methods. When comparing privacy level-based user personas, the lower *Precision*, *Recall*, *F1-score* and *nDCG* results are obtained for *FF*, followed by *PP* and *BB*. This finding is valid for all personalized CF algorithms in the ML; however, the differences among such privacy level-based user personas seem to be limited for *NEUMF* and *VAECF* algorithms on DB and Yelp datasets. Therefore, we can conclude that as the privacy degree adjusted for users improves, the accuracy of their received recommendations becomes worse in general. The main reason for this
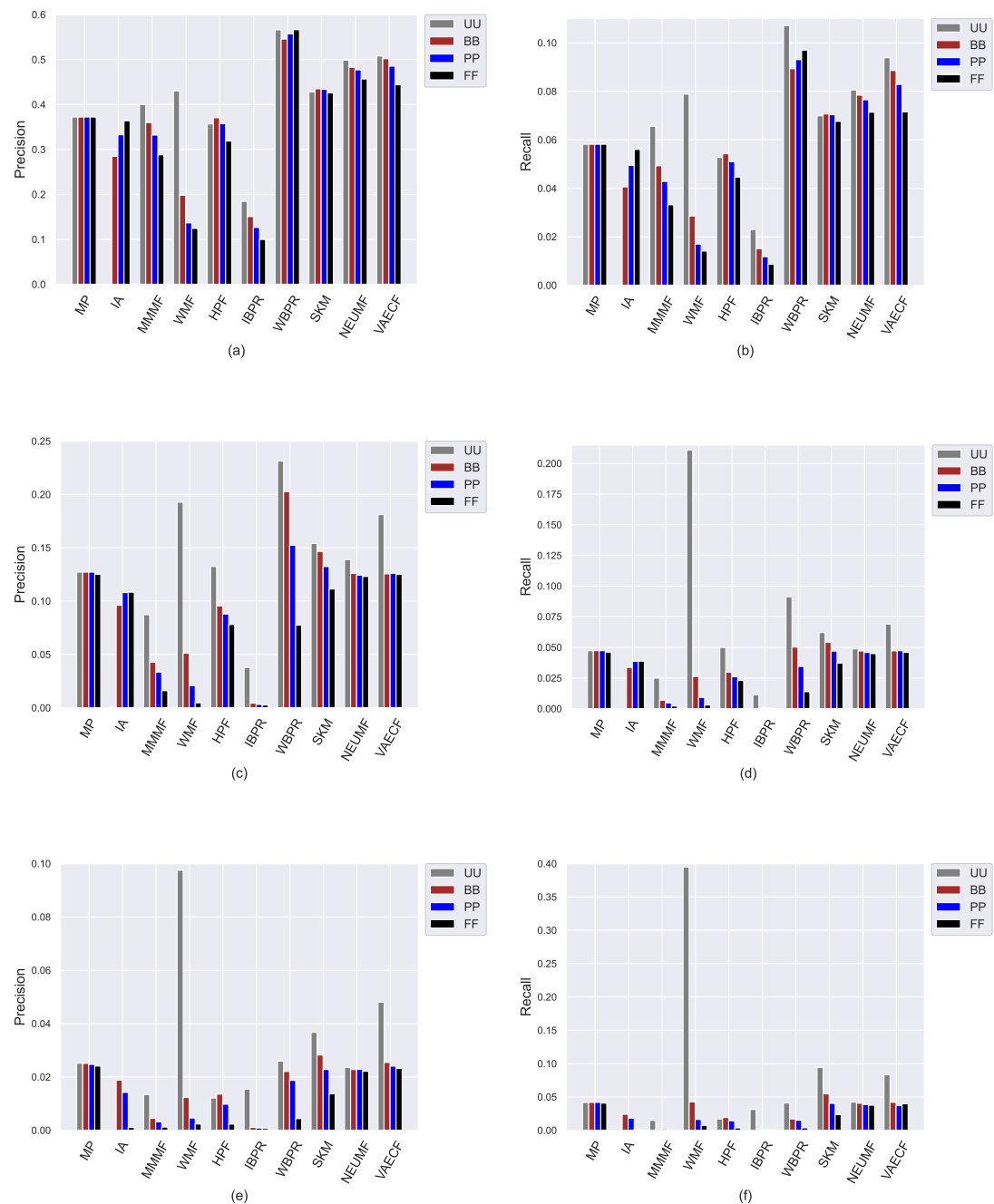
Figure 8. The obtained *Precision* and *Recall* outcomes of the utilized algorithms for four different privacy level-based user personas. *Precision* results are presented in (a), (c), and (e) for the ML, DB, and Yelp datasets, respectively, while *Recall* results are presented in (b), (d), and (f).

consequence is that the algorithms can model individuals better when their original profiles are handled and thus provide highly accurate recommendations; however, their performances significantly diminish when their disguised profiles are considered. Note that this finding parallels previous research on PPCF topics (Bilge and Polat, 2013).

> **Observation 5**: As the level of privacy provided for users improves, the accuracy of their received recommendations diminishes.
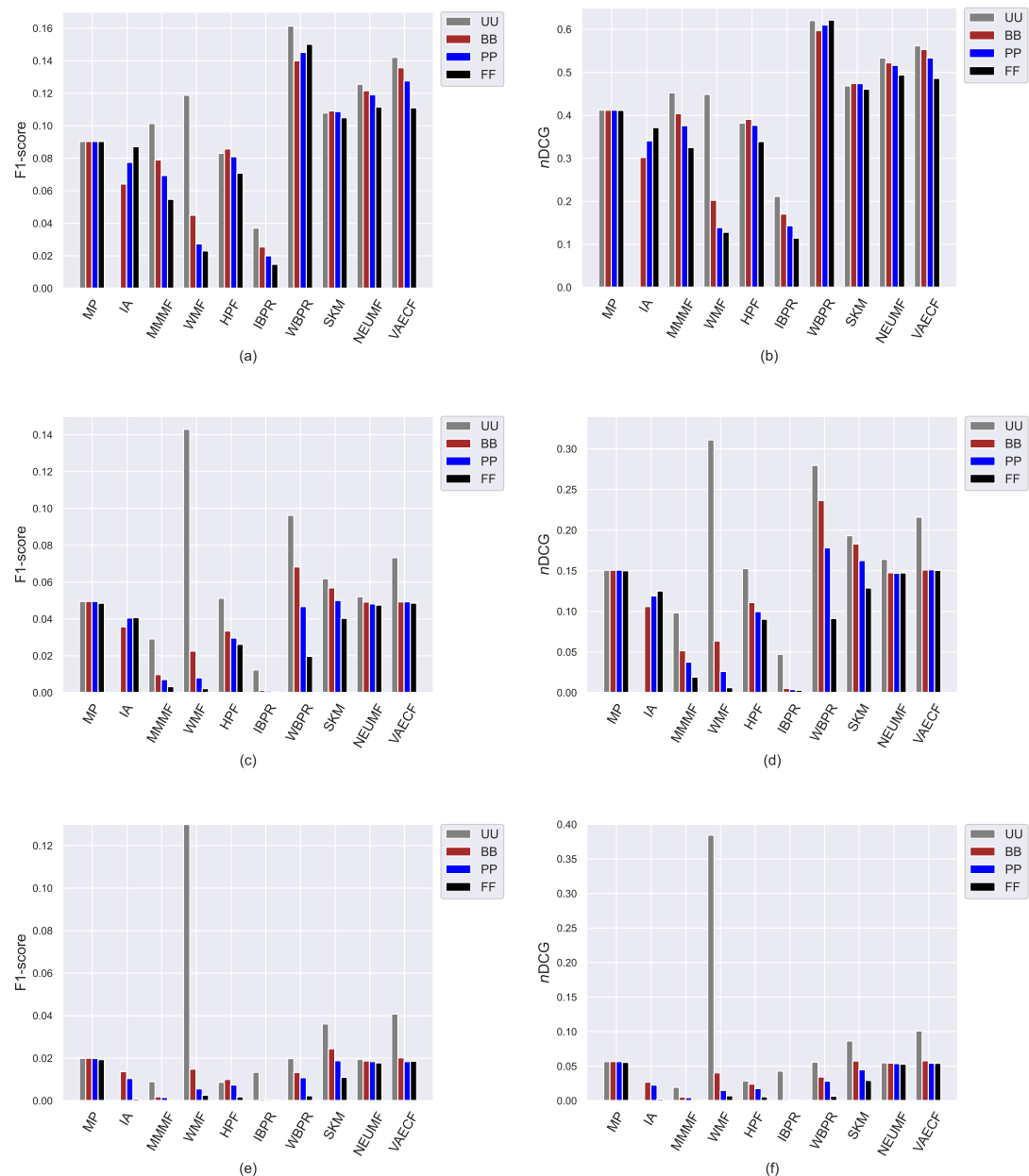
Figure 9. The obtained *F1-score* and *nDCG* outcomes of the utilized algorithms for four different privacy level-based user personas. *F1-score* results are presented in (a), (c), and (e) for the ML, DB, and Yelp datasets, respectively, while *nDCG* results are presented in (b), (d), and (f).

ANALYZING HOW POPULARITY-DEBIASING APPROACHES DIFFERENTLY PERFORM FOR PRIVACY LEVEL-BASED USER PERSONAS

This section presents the experimental results to investigate how the conventional popularity-debiasing methods perform for varying privacy level-based user personas. To this end, we consider three benchmark treatment methods previously explained, i.e., *Aug*, *Mul*, and *xQd*, and measure their effects on user profiles with varying privacy levels (i.e., *BB*, *PP*, and *FF*) and original profiles (i.e., *UU*). We realize this set of experiments on the Yelp dataset, as it is sparser and larger than others, like in most real-world applications. In addition, since the differences between privacy level-based user personas are more obvious for the *HPF* algorithm, we only consider this algorithm for the recommendation phase to see the debiasing methods' effects better and for the sake of clarity. In these settings, we present the obtained

684 improvements/deteriorations in beyond-accuracy (i.e., *APRI*, *APLT*, *Entropy*, *Novelty*, and *LTC*) and
685 accuracy (i.e., *Precision*, *Recall*, *F1-score*, and *nDCG*) results with percentages when the *Aug*, *Mul*, or
686 *xQd* applied in Figs 10 and 11, respectively.
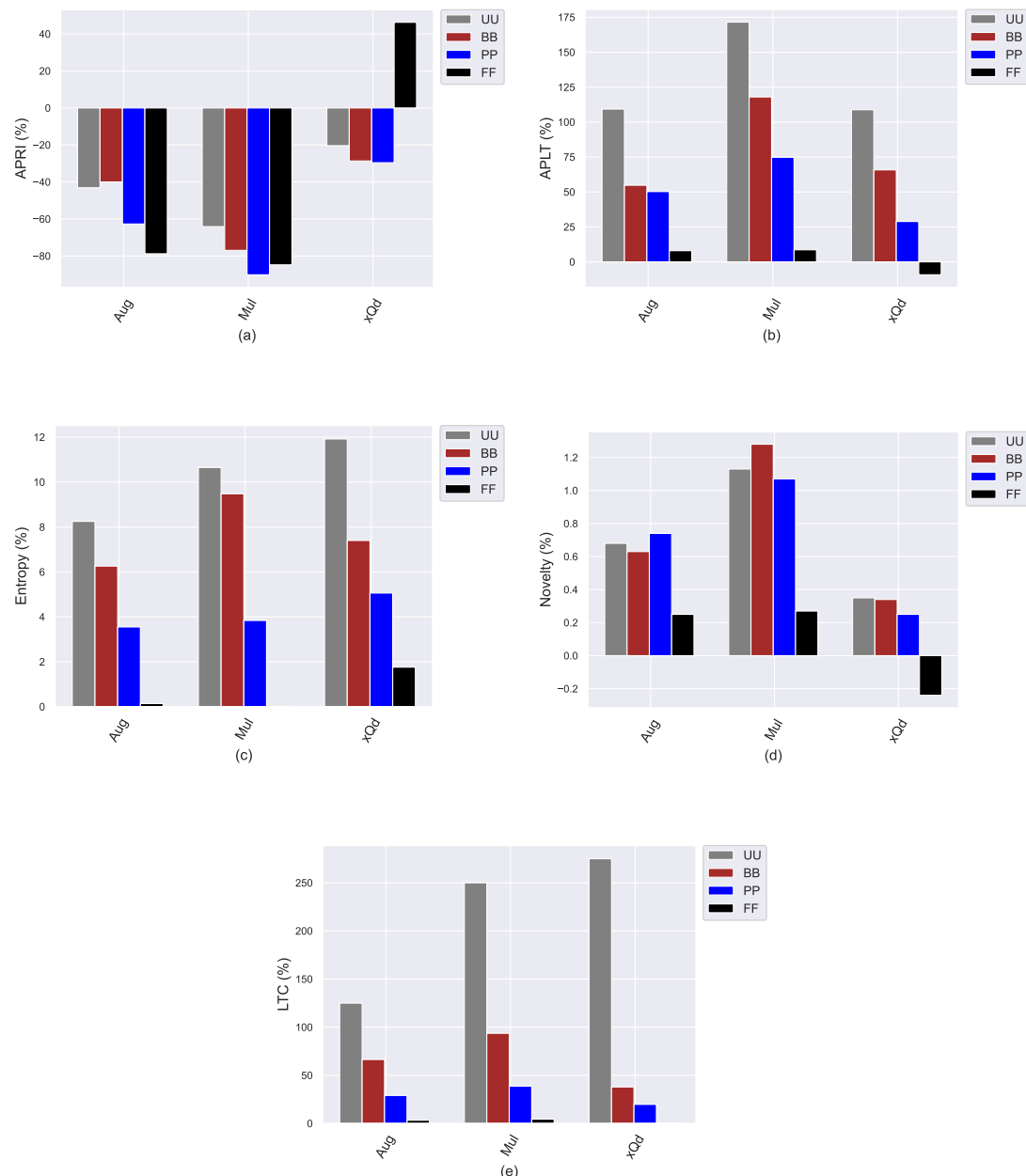
Figure 10. The obtained (a) *APRI*, (b) *APLT*, (c) *Entropy*, (d) *Novelty*, and (e) *LTC* improvement or deterioration level when the *Aug*, *Mul*, and *xQd* popularity-debiasing methods are applied for different privacy level-based user personas in the Yelp dataset.

687 As can be followed by Fig. 10, we observe that the most significant reductions for the *APRI* are
688 obtained for *FF*, followed by *PP*, *BB*, and *UU*. Considering that lower values for this metric indicate
689 less-biased recommendations against item popularity, we can conclude that more sensitive individuals
690 to their confidentiality benefit more from such treatment methods against popularity bias. On the other
691 hand, we have interestingly observed that the debiasing methods show worse performances for such
692 privacy-sensitive users in terms of accuracy. Note that the most severe deteriorations in ranking accuracy
693 are observed for *FF*, followed by *PP*, *BB*, and *UU*, as can be followed by all *Precision*, *Recall*, *F1-score*,

₆₉₄ and *nDCG* results given in Fig. 11. This observation is interesting since the accuracy-concerned metrics
₆₉₅ usually check how much the recommended items hit with the items rated in users' original profiles.
₆₉₆ Therefore, we expect that the overall accuracy performance improves with recommending popular items.
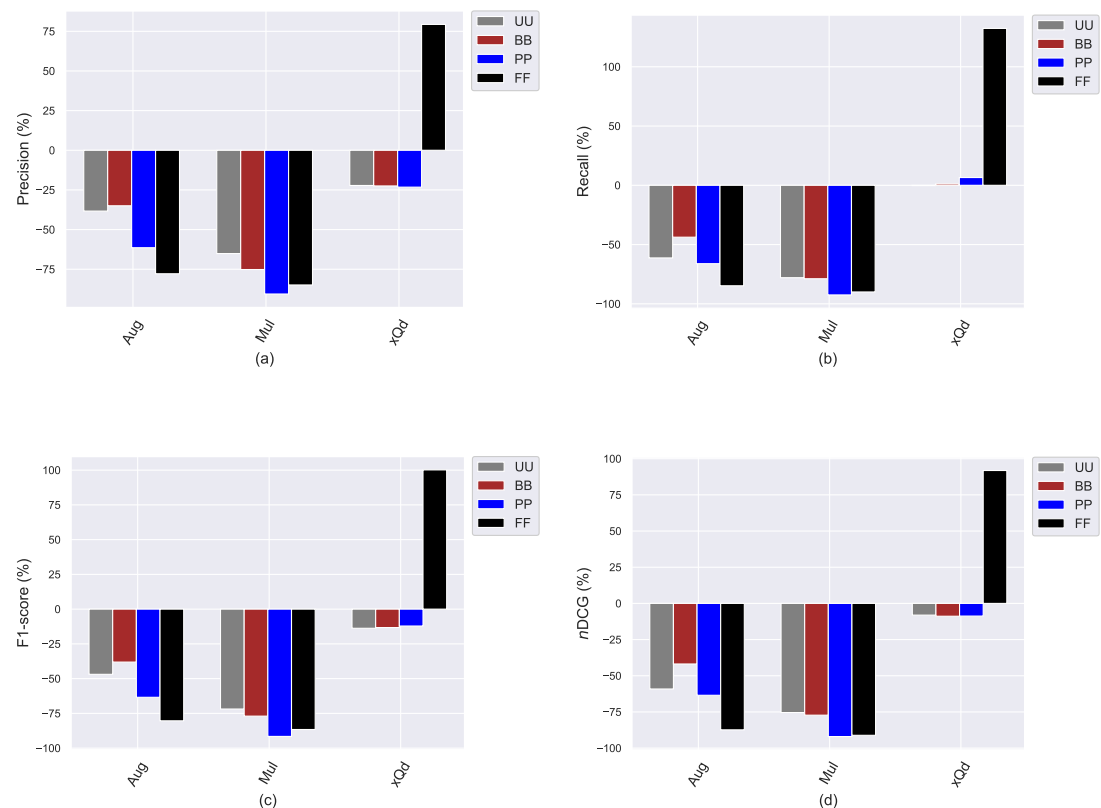
Figure 11. The obtained (a) *Precision*, (b) *Recall*, (c) *F1-score*, and (d) *nDCG* improvement or
deterioration level when the *Aug*, *Mul*, and *xQd* popularity-debiasing methods are applied for different
privacy level-based user personas in the Yelp dataset.

₆₉₇   Contrary to this pattern, on the other hand, the greatest improvements in *APLT*, *Entropy*, *Novelty*,
₆₉₈ and *LTC* results of the recommendations are obtained for *UU*, followed by *BB*, *PP*, and *FF*. This trend
₆₉₉ is also valid for all utilized popularity-debiasing methods, i.e., *Aug*, *Mul*, and *xQd*. In other words, the
₇₀₀ considered popularity-debiasing methods are more effective when the original profiles of the users are
₇₀₁ considered, and their ability to enhance beyond-accuracy recommendation quality decreases as the users'
₇₀₂ privacy level increases. This is an unsurprising finding since we have previously observed that as the level
₇₀₃ of privacy provided for users increases, the beyond-accuracy qualities of their received recommendations
₇₀₄ are notably improved. Thus, the effects of the utilized debiasing methods become limited for individuals
₇₀₅ who are more concerned about their privacy. In addition, such privacy-sensitive individuals also seem
₇₀₆ to be disadvantageous in terms of accuracy, as the highest decrements in ranking accuracy are usually
₇₀₇ obtained for *FF*, followed by *BB*, *PP*, and *UU*, as discussed above.

₇₀₈   One of the most important findings is that applying one of the highly prominent popularity-debiasing
₇₀₉ methods (i.e., the *xQd*) for *FF* users even significantly deteriorates recommendation quality (note
₇₁₀ that its improvements are negative for *APLT* and *Novelty* while positive for *APRI*). This important
₇₁₁ finding demonstrates that applying privacy-protection procedures to user profiles might render some
₇₁₂ well-performing popularity-debiasing methods useless. The empirical outcomes also suggest that the
₇₁₃ most successful popularity-debiasing method differs for privacy level-based user personas depending on
₇₁₄ the considered quality aspect.

> **Observation 6**: As users' privacy protection improves, the popularity-debiasing methods can perform better in achieving less-biased recommendations against item popularity. However, their effects on both accuracy and beyond-accuracy recommendation quality considerably diminish for users who are more sensitive to their confidentiality. Some of such treatment methods even become useless for *fiery fundamentalists*, i.e., users hypersensitive to their privacy.

CONCLUSION AND FUTURE WORK

One main task of recommender systems is to produce appropriate referrals to users by protecting their privacy. The most common approach for this aim is masking the list of rated items and actual ratings provided for these items. Then, any collaborative recommender is trained on disguised preference data rather than the original user-item matrix. Since the main reason for the well-known popularity bias issue of the recommenders is the observed imbalances in the rating distribution, such privacy-protection procedure might lead to significant changes in the degree of observed bias in the recommendations and, worse, make treatment methods developed for this problem less useful.

Therefore, in this study, we mainly analyze how popularity bias inclinations of the recommenders vary for four user personas defined according to varying privacy protection levels. We also investigate how the performances of three benchmark popularity-debiasing methods change when such privacy level-based user personas are considered. One of the most important findings derived from the conducted comprehensive experimental studies is that disguising the actual preference data of individuals not only eliminates users' concerns regarding their confidentiality but also makes the recommendations less biased towards item popularity at the same time. In other words, increasing the privacy level for users makes the collaborative recommenders more robust against the popularity bias problem. Thus, more sensitive users about their privacy can receive more qualified recommendations regarding beyond-accuracy aspects like diversity, novelty, catalogue coverage and fairness (i.e., equality in representing items), even though they bearably sacrifice from accuracy. Another important finding observed from our analysis is that as users' privacy protection level improves, even if the performance of the considered popularity-debiasing methods enhances in featuring less-popular items in their recommendations, their effects on both beyond-accuracy and accuracy recommendation quality considerably diminish. Some of them even render impractical for individuals hypersensitive to their privacy.

When constructing privacy level-based user personas, we consider the randomized perturbation techniques as one of the most common approaches to provide user confidentiality. However, our analysis can be extended by considering alternative privacy-preserving methods, such as differential privacy and cryptographic techniques. Also, there is a need for novel popularity-debiasing methods that will not be affected by the data disguising procedure and thus show comparable performances for the scenario of users' privacy is preserved.

REFERENCES

Abdollahpouri, H. (2020). *Popularity bias in recommendation: A multi-stakeholder perspective*. PhD thesis, University of Colorado at Boulder.

Abdollahpouri, H. and Burke, R. (2019). Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158*.

Abdollahpouri, H., Burke, R., and Mobasher, B. (2017). Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 42–46.

Abdollahpouri, H., Burke, R., and Mobasher, B. (2018). Popularity-aware item weighting for long-tail recommendation. *arXiv preprint arXiv:1802.05382*.

Abdollahpouri, H., Burke, R., and Mobasher, B. (2019a). Managing popularity bias in recommender systems with personalized re-ranking. In *The thirty-second international flairs conference*.

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. (2019b). The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286*.

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. (2020). The connection between popularity bias, calibration, and fairness in recommendation. In *Fourteenth ACM conference on recommender systems*, pages 726–731.

Abdollahpouri, H., Mansoury, M., Burke, R., Mobasher, B., and Malthouse, E. (2021). User-centered evaluation of popularity bias in recommender systems. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 119–129.

Agarwal, A., Mishra, D. S., and Kolekar, S. V. (2022). Knowledge-based recommendation system using semantic web rules based on learning styles for moocs. *Cogent Engineering*, 9(1):2022568.

Agrawal, D. and Aggarwal, C. C. (2001). On the design and quantification of privacy preserving data mining algorithms. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*.

Badsha, S., Yi, X., and Khalil, I. (2016). A practical privacy-preserving recommender system. *Data Science and Engineering*, 1(3):161–177.

Badsha, S., Yi, X., Khalil, I., and Bertino, E. (2017). Privacy preserving user-based recommender system. In *2017 IEEE 37th international conference on Distributed Computing Systems (ICDCS)*, pages 1074–1083. IEEE.

Badsha, S., Yi, X., Khalil, I., Liu, D., Nepal, S., and Lam, K.-Y. (2018). Privacy preserving user based web service recommendations. *IEEE Access*, 6:56647–56657.

Bilge, A. and Polat, H. (2013). A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. *Information Processing & Management*, 49(4):912–927.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Boratto, L., Fenu, G., and Marras, M. (2019). The effect of algorithmic bias on recommender systems for massive open online courses. In *European Conference on Information Retrieval*, pages 457–472. Springer.

Boratto, L., Fenu, G., and Marras, M. (2021). Connecting user and item perspectives in popularity debiasing for collaborative recommendation. *Information Processing & Management*, 58(1):102387.

Borges, R. and Stefanidis, K. (2021). On mitigating popularity bias in recommendations via variational autoencoders. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1383–1389.

Bosri, R., Rahman, M. S., Bhuiyan, M. Z. A., and Al Omar, A. (2020). Integrating blockchain with artificial intelligence for privacy-preserving recommender systems. *IEEE Transactions on Network Science and Engineering*, 8(2):1009–1018.

Boutet, A., Frey, D., Guerraoui, R., Jégou, A., and Kermarrec, A.-M. (2016). Privacy-preserving distributed collaborative filtering. *Computing*, 98(8):827–846.

Casino, F., Domingo-Ferrer, J., Patsakis, C., Puig, D., and Solanas, A. (2015). A k-anonymous approach to privacy preserving collaborative filtering. *Journal of Computer and System Sciences*, 81(6):1000–1011.

Chen, C., Zhang, M., Liu, Y., and Ma, S. (2018). Missing data modeling with user activity and item popularity in recommendation. In *Asia Information Retrieval Symposium*, pages 113–125. Springer.

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., and He, X. (2020). Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*.

Chen, X. and Huang, V. (2012). Privacy preserving data publishing for recommender system. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, pages 128–133. IEEE.

Cinelli, M., De Francisci Morales, G., Galeazzi, A., Quattrociocchi, W., and Starnini, M. (2021). The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9):e2023301118.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Elahi, M., Kholgh, D. K., Kiarostami, M. S., Saghari, S., Rad, S. P., and Tkalčič, M. (2021). Investigating the impact of recommender systems on user-based and item-based popularity bias. *Information Processing & Management*, 58(5):102655.

Elmisery, A. M. and Botvich, D. (2017). An enhanced middleware for collaborative privacy in iptv recommender services. *arXiv preprint arXiv:1711.07593*.

Gantner, Z., Drumond, L., Freudenthaler, C., and Schmidt-Thieme, L. (2012). Personalized ranking for non-uniformly sampled items. In *Proceedings of KDD Cup 2011*, pages 231–247. PMLR.

Gong, S. (2011). Privacy-preserving collaborative filtering based on randomized perturbation techniques and secure multiparty computation. *International Journal of Advancements in Computing Technology*, 3(4):89–99.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.

Guerraoui, R., Kermarrec, A.-M., Patra, R., and Taziki, M. (2015). D 2 p: distance-based differential privacy in recommenders. *Proceedings of the VLDB Endowment*, 8(8):862–873.

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Hou, L., Pan, X., and Liu, K. (2018a). Balancing the popularity bias of object similarities for personalised recommendation. *The European Physical Journal B*, 91(3):1–7.

Hou, M., Wei, R., Wang, T., Cheng, Y., and Qian, B. (2018b). Reliable medical recommendation based on privacy-preserving collaborative filtering. *Computers, Materials & Continua*, 56(1):137–149.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee.

Jannach, D., Lerche, L., Kamehkhosh, I., and Jugovac, M. (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491.

Jannach, D. and Zanker, M. (2022). Value and impact of recommender systems. In *Recommender Systems Handbook*, pages 519–546. Springer.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2014). Correcting popularity bias by enhancing recommendation neutrality. In *RecSys Posters*.

Kearns, M. and Ron, D. (1997). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 152–162.

Kerkouche, R., Acs, G., Castelluccia, C., and Genevès, P. (2021). Privacy-preserving and bandwidth-efficient federated learning: An application to in-hospital mortality prediction. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 25–35.

Ko, H., Lee, S., Park, Y., and Choi, A. (2022). A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1):141.

Koren, Y., Rendle, S., and Bell, R. (2022). Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142.

Kowald, D., Schedl, M., and Lex, E. (2020). The unfairness of popularity bias in music recommendation: A reproducibility study. In *European conference on information retrieval*, pages 35–42. Springer.

Le, D. D. and Lauw, H. W. (2017). Indexable bayesian personalized ranking for efficient top-k recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1389–1398.

Li, D., Chen, C., Lv, Q., Shang, L., Zhao, Y., Lu, T., and Gu, N. (2016). An algorithm for efficient privacy-preserving item-based collaborative filtering. *Future Generation Computer Systems*, 55:311–320.

Li, D., Lv, Q., Shang, L., and Gu, N. (2017). Efficient privacy-preserving content recommendation for online social communities. *Neurocomputing*, 219:440–454.

Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. (2018). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698.

Lin, L., Tian, Y., and Liu, Y. (2021). A blockchain-based privacy-preserving recommendation mechanism. In *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*, pages 74–78. IEEE.

Liu, X., Liu, A., Zhang, X., Li, Z., Liu, G., Zhao, L., and Zhou, X. (2017). When differential privacy meets randomized perturbation: a hybrid approach for privacy-preserving recommender system. In *International Conference on database systems for advanced applications*, pages 576–591. Springer.

Lu, S., Gao, Z., Xu, Q., Jiang, C., Zhang, A., and Wang, X. (2022). Class-imbalance privacy-preserving federated learning for decentralized fault diagnosis with biometric authentication. *IEEE Transactions on Industrial Informatics*, 18(12):9101–9111.

McSherry, F. and Mironov, I. (2009). Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636.

Omar, A. A., Bosri, R., Rahman, M. S., Begum, N., and Bhuiyan, M. Z. A. (2019). Towards privacy-preserving recommender system with blockchains. In *Dependability in Sensor, Cloud, and Big Data Systems and Applications: 5th International Conference, DependSys 2019, Guangzhou, China,*

872   *November 12–15, 2019, Proceedings 5*, pages 106–118. Springer.

873   Ozturk, A. and Polat, H. (2015). From existing trends to future trends in privacy-preserving collaborative
874   filtering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):276–291.

875   Parameswaran, R. and Blough, D. M. (2007). Privacy preserving collaborative filtering using data
876   obfuscation. In *2007 IEEE International Conference on Granular Computing (GRC 2007)*, pages
877   380–380. IEEE.

878   Park, Y.-J. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In
879   *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18.

880   Polat, H. and Du, W. (2003). Privacy-preserving collaborative filtering using randomized perturbation
881   techniques. In *Third IEEE International Conference on Data Mining*, pages 625–628. IEEE.

882   Polat, H. and Du, W. (2005a). Privacy-preserving collaborative filtering. *International journal of electronic
883   commerce*, 9(4):9–35.

884   Polat, H. and Du, W. (2005b). Privacy-preserving collaborative filtering on vertically partitioned data.
885   In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 651–658.
886   Springer.

887   Polatidis, N., Georgiadis, C. K., Pimenidis, E., and Mouratidis, H. (2017). Privacy-preserving collaborative
888   recommendations based on random perturbations. *Expert Systems with Applications*, 71:18–25.

889   Pramod, D. (2022). Privacy-preserving techniques in recommender systems: state-of-the-art review and
890   future research agenda. *Data Technologies and Applications*, (ahead-of-print).

891   Salah, A., Rogovschi, N., and Nadif, M. (2016). A dynamic collaborative filtering system via a weighted
892   clustering approach. *Neurocomputing*, 175:206–215.

893   Salah, A., Truong, Q.-T., and Lauw, H. W. (2020). Cornac: A comparative framework for multimodal
894   recommender systems. *J. Mach. Learn. Res.*, 21(95):1–5.

895   Sanders, R. (1987). The pareto principle: its use and abuse. *Journal of Services Marketing*.

896   Shambour, Q., Hussein, A. H., Kharma, Q., and Abu-Alhaj, M. M. (2022). Effective hybrid content-based
897   collaborative filtering approach for requirements engineering. *Comput. Syst. Sci. Eng.*, 40(1):113–125.

898   Shen, Y. and Jin, H. (2014). Privacy-preserving personalized recommendation: An instance-based
899   approach via differential privacy. In *2014 IEEE International Conference on Data Mining*, pages
900   540.-549. IEEE.

901   Shi, C., Hu, B., Zhao, W. X., and Philip, S. Y. (2018). Heterogeneous information network embedding for
902   recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370.

903   Shmueli, E. and Tassa, T. (2017). Secure multi-party protocols for item-based collaborative filtering. In
904   *Proceedings of the eleventh ACM conference on recommender systems*, pages 89–97.

905   Si, M. and Li, Q. (2020). Shilling attacks against collaborative recommender systems: a review. *Artificial
906   Intelligence Review*, 53(1):291–319.

907   Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in
908   artificial intelligence*, 2009.

909   Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y. (2019). A hybrid
910   approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on
911   artificial intelligence and security*, pages 1–11.

912   Wei, R., Tian, H., and Shen, H. (2018). Improving k-anonymity based privacy preservation for collabora-
913   tive filtering. *Computers & Electrical Engineering*, 67:509–519.

914   Weimer, M., Karatzoglou, A., and Smola, A. (2008). Improving maximum margin matrix factorization.
915   *Machine Learning*, 72(3):263–276.

916   Xu, R., Baracaldo, N., Zhou, Y., Anwar, A., and Ludwig, H. (2019). Hybridalpha: An efficient approach
917   for privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial
918   intelligence and security*, pages 13–23.

919   Yalcin, E. (2022a). Exploring potential biases towards blockbuster items in ranking-based recommenda-
920   tions. *Data Mining and Knowledge Discovery*, pages 1–41.

921   Yalcin, E. (2022b). Pophybrid: a novel item popularity-aware hybrid approach for long-tail recommen-
922   dation. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic
923   Applications (HORA)*, pages 1–6. IEEE.

924   Yalcin, E. and Bilge, A. (2021). Investigating and counteracting popularity bias in group recommendations.
925   *Information Processing & Management*, 58(5):102608.

926   Yalcin, E. and Bilge, A. (2022). Evaluating unfairness of popularity bias in recommender systems: A

comprehensive user-centric analysis. *Information Processing & Management*, 59(6):103100.

Yalcin, E. and Bilge, A. (2023). Popularity bias in personality perspective: An analysis of how personality traits expose individuals to the unfair recommendation. *Concurrency and Computation: Practice and Experience*, 35(9):e7647.

Yargic, A. and Bilge, A. (2019). Privacy-preserving multi-criteria collaborative filtering. *Information Processing & Management*, 56(3):994–1009.

Yin, X., Zhu, Y., and Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, 54(6):1–36.

Zhang, F., Lee, V. E., and Choo, K.-K. R. (2018). Jo-dpmf: Differentially private matrix factorization learning through joint optimization. *Information Sciences*, 467:271–281.

Zou, J. and Fekri, F. (2015). A belief propagation approach to privacy-preserving item-based collaborative filtering. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1306–1318.