ReuseKNN: Neighborhood Reuse for Differentially Private KNN-Based Recommendations

PETER MÜLLNER, Know-Center GmbH and Graz University of Technology, Austria
ELISABETH LEX, Graz University of Technology, Austria
MARKUS SCHEDL, Johannes Kepler University Linz and Linz Institute of Technology, Austria
DOMINIK KOWALD, Know-Center GmbH and Graz University of Technology, Austria

User-based *KNN* recommender systems (*UserKNN*) utilize the rating data of a target user's k nearest neighbors in the recommendation process. This, however, increases the privacy risk of the neighbors, since the recommendations could expose the neighbors' rating data to other users or malicious parties. To reduce this risk, existing work applies differential privacy by adding randomness to the neighbors' ratings, which unfortunately reduces the accuracy of *UserKNN*. In this work, we introduce *ReuseKNN*, a novel differentially private KNN-based recommender system. The main idea is to identify small but highly reusable neighborhoods so that (i) only a minimal set of users requires protection with differential privacy and (ii) most users do not need to be protected with differential privacy since they are only rarely exploited as neighbors. In our experiments on five diverse datasets, we make two key observations. Firstly, *ReuseKNN* requires significantly smaller neighborhoods and, thus, fewer neighbors need to be protected with differential privacy compared with traditional *UserKNN*. Secondly, despite the small neighborhoods, *ReuseKNN* outperforms *UserKNN* and a fully differentially private approach in terms of accuracy. Overall, *ReuseKNN* leads to significantly less privacy risk for users than in the case of *UserKNN*.

CCS Concepts: • **Information systems** → **Recommender systems**; **Collaborative filtering**; • **Security and privacy** → **Privacy-preserving protocols;**

Additional Key Words and Phrases: Neighborhood reuse, differential privacy, collaborative filtering, k nearest neighbors, recommender systems, privacy risk, popularity bias

ACM Reference format:
Peter Müllner, Elisabeth Lex, Markus Schedl, and Dominik Kowald. 2023. ReuseKNN: Neighborhood Reuse for Differentially Private KNN-Based Recommendations. *ACM Trans. Intell. Syst. Technol.* 14, 5, Article 80 (August 2023), 29 pages.
https://doi.org/10.1145/3608481

This research is funded by the "DDAI" COMET Module within the COMET — Competence Centers for Excellent Technologies Programme, funded by the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry for Digital and Economic Affairs (bmdw), the Austrian Research Promotion Agency (FFG), the province of Styria (SFG) and partners from industry and academia. The COMET Programme is managed by FFG. This research received support by the TU Graz Open Access Publishing Fund, the Austrian Science Fund (FWF): P33526 and DFH-23; and by the State of Upper Austria and the Federal Ministry of Education, Science, and Research, through grant no. LIT-2020-9-SEE-113.
Authors' addresses: P. Müllner and D. Kowald, Know-Center GmbH, 8010, Sandgasse 36/4, Graz, Austria and Graz University of Technology, 8010, Rechbauerstraße 12, Graz, Austria; emails: pmuellner@know-center.at, pmuellner@student.tugraz.at, dkowald@know-center.at, dominik.kowald@tugraz.at; E. Lex, Graz University of Technology, 8010, Rechbauerstraße 12, Graz, Austria; email: elisabeth.lex@tugraz.at; M. Schedl, Johannes Kepler University Linz and Linz Institute of Technology, 4040, Altenberger Straße 69, Linz, Austria; email: markus.schedl@jku.at.

2157-6904/2023/08-ART80 $15.00
https://doi.org/10.1145/3608481

1 INTRODUCTION

Recommender systems often rely on neighborhood-based collaborative filtering [30] to generate recommendations. These systems can intuitively justify their recommendations to the target user and also efficiently incorporate new rating data from users, which are two key issues of modern recommender systems [16]. For example, user-based *KNN*, i.e., *UserKNN*, is a variant of neighborhood-based collaborative filtering that utilizes the rating data of the k nearest neighbors of a target user to process a rating query. A rating query is a request to a recommender system to predict a rating for a target user to a target item. However, the way in which rating queries are processed by *UserKNN* can increase the privacy risk of users since the estimated rating scores, which determine whether an item will be recommended, are generated based on rating data of users that are used as neighbors. In this regard, existing research [9, 49, 64] finds that these neighbors are susceptible to multiple privacy risks, such as the inference of their private rating data (see Section 3). To mitigate that privacy risk, several works [10, 24, 65] use *differential privacy (DP)* [18, 20] to protect users' rating data by adding a degree of randomness to the data. However, the added randomness typically leads to severe drops in recommendation accuracy [7].

To address this problem, we introduce *ReuseKNN*, a novel differentially private KNN-based recommender system that reduces the number of neighbors to which differential privacy needs to be applied. Intuitively, instead of utilizing new users as neighbors for processing new rating queries, *ReuseKNN* reuses useful neighbors from past rating queries. Hence, *ReuseKNN* constructs small but highly reusable neighborhoods for every target user by fostering the neighbors' reusability for many rating queries. With this, as illustrated in Figure 1, *ReuseKNN* minimizes the set of users that need to be protected with DP—we call them "vulnerable users". Plus, most users do not need to be protected with DP, as their rating data is only rarely used in the recommendation process—we call them "secure users". As shown, we also introduce a data usage threshold τ, i.e., a hyperparameter that allows adjusting the maximum data usage for a user to be treated as secure. In this way, we leave it to the recommender system provider to specify what degree of data usage is tolerated despite the resulting privacy risks and which users need to be protected.

We evaluate the proposed approach in a two-stage procedure: (i) neighborhood reuse only, i.e., *ReuseKNN*, and (ii) neighborhood reuse with DP, i.e., $ReuseKNN_{DP}$. In the first stage, *ReuseKNN* does not use DP at all. With this, we focus on how neighborhood reuse can increase the reusability of neighbors and preserve *UserKNN*'s recommendation accuracy. In the second stage, we combine *ReuseKNN* with DP, i.e., $ReuseKNN_{DP}$, to protect vulnerable users with DP. This allows the investigation of how $ReuseKNN_{DP}$ can mitigate all users' privacy risk while generating accurate recommendations. We evaluate *ReuseKNN* and $ReuseKNN_{DP}$ on five different datasets: *MovieLens 1M*, *Douban*, *LastFM*, *Ciao*, and *Goodreads*. Plus, we compare *ReuseKNN* and $ReuseKNN_{DP}$ with five KNN-based baselines that utilize DP (e.g., [65]) and the concept of neighborhood reuse in different ways with respect to recommendation accuracy and users' privacy risk. Additionally, the nature of neighborhood reuse may raise concerns that the generated recommendations are biased towards items consumed by many users, i.e., popular items. Thus, we investigate whether the proposed approach is more or less prone to item popularity bias than the baselines.

Our results indicate that *ReuseKNN* yields significantly smaller neighborhoods than traditional *UserKNN*. Despite the smaller neighborhoods, *ReuseKNN* and $ReuseKNN_{DP}$ outperform our baselines in terms of recommendation accuracy. Moreover, $ReuseKNN_{DP}$ leads to significantly less privacy risk for users than *UserKNN* with DP. Also, the proposed approach does not increase item popularity bias. Overall, the three main contributions of this article are as follows:

(1) We present a novel *ReuseKNN* recommender system and compare two neighborhood reuse strategies to substantially foster the reusability of a target user's neighborhood and effectively reduce the number of vulnerable users.

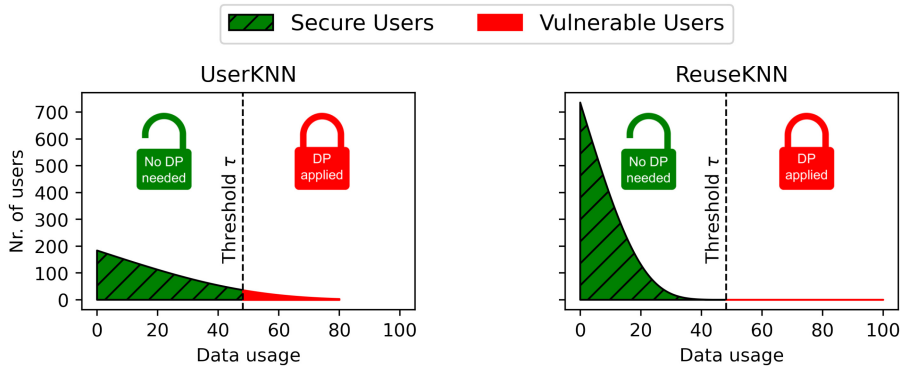

Fig. 1. Schematic illustration of the data usage (i.e., how often a user is used as a neighbor) distribution of traditional *UserKNN* and the proposed *ReuseKNN* recommender system. *ReuseKNN* increases the number of secure users (green, no differential privacy needed) and decreases the number of vulnerable users (red, differential privacy needs to be applied) compared with *UserKNN*. The dashed line illustrates the data usage threshold τ, a hyperparameter for adjusting the maximum data usage for a user to be treated as secure.

(2) We combine *ReuseKNN* with DP to realize $ReuseKNN_{DP}$ and show that $ReuseKNN_{DP}$ improves recommendation accuracy over KNN- and DP-based baselines and, at the same time, does not increase item popularity bias.
(3) We show that $ReuseKNN_{DP}$ leads to significantly less privacy risk, since most users are rarely exploited in the recommendation process and only the remaining users, i.e., vulnerable users, are protected with DP.

Our work illustrates how to address privacy risks in KNN-based recommender systems through neighborhood reuse combined with DP. While the proposed approach focuses on traditional *KNN*, we additionally demonstrate the generalizability of the neighborhood reuse principle to user and item embeddings created by state-of-the-art neural collaborative filtering approaches [29].

2 RELATED WORK

We describe two research strands related to our work: (i) studies on the identification and quantification of users' privacy risks in recommender systems and (ii) privacy-aware recommender systems that mitigate users' privacy risks. Since *ReuseKNN* is a differentially private and KNN-based recommender system, we emphasize KNN-based methods when reviewing privacy risks in recommender systems as well as DP when reviewing privacy-preserving technologies for recommender systems. Also, we focus on the privacy risks that arise from the recommendations presented to potentially malicious target users. This can harm the neighbors used in the recommendation process.

2.1 Privacy Risks in Recommender Systems

Previous research [5, 23, 36, 49] describes many severe privacy risks for users of recommender systems. For example, according to Ramakrishnan et al. [49], the use of neighbors' rating data in the recommendation process can pose a privacy risk to the neighbors. Serendipitous recommendations could reveal unique connections between neighbors and items. In this way, the rating data of the neighbors can be uncovered or the neighbors' identities can be revealed within the recommendation database. Also, Zhang et al. [64] show that it could be possible to identify users whose data was used in the recommendation process. Their results suggest that the effectiveness of their attack depends on the number of generated recommendations. Moreover,

Calandrino et al. [9] propose to generate fake users, i.e., sybils, based on limited knowledge of a victim's data. These sybils can isolate the victim that is utilized as a neighbor and compromise its privacy.

To quantify users' privacy risks in computational systems such as recommender systems, several privacy risk metrics [13, 17, 42, 53, 56] have been proposed. These metrics often rely on the sensitivity of users' data, i.e., how strong this data puts users' privacy at risk. For example, Chen et al. [13] detect correlations within the dataset to measure whether a piece of data could reveal personal information about the users. Srivastava and Geethakumari [53] measure the relative sensitivity of a single piece of data compared with the remaining data of a user. Similarly, Domingo-Ferrer [17] relates the overall sensitivity of a user's data to the sensitivity of other users' data. Liu and Terzi's *privacy score* [42] weighs the sensitivity with the degree of visibility of a user's data (i.e., how often a user's data is utilized in the recommendation process).

Evaluating the privacy risk of users based on attacks only measures the privacy risk with respect to the specific attack scenario. Liu and Terzi's metric measures users' privacy risk independent of specific attack scenarios and, thus, allows investigating privacy risk in a recommender system at a more general level. Therefore, in our work, we utilize Liu and Terzi's metric to measure users' privacy risk in a general neighborhood-based recommendation scenario. Furthermore, we assume that all pieces of data are equally sensitive, since sensitivity typically depends on the application and the user's perception of privacy [38].

2.2 Privacy-Aware Recommender Systems

Several works [33, 55, 63] mitigate users' privacy risks by applying *homomorphic encryption* [25] to users' rating data. Here, recommendations are generated based on the encrypted rating data, and, thus, users' rating data remains protected in the recommendation process. Homomorphic encryption, however, has high computational complexity. Thus, Tang and Wang [55] apply homomorphic encryption on the rating data of a target users' friends only, i.e., a small subset of users, to improve computational efficiency. Besides homomorphic encryption, *federated learning* [44] is used to lower users' privacy risks [27, 41, 48, 60]. Specifically, instead of a user's rating data, the parameters of the user's local recommendation model are utilized in the recommendation process. For example, Perifanis and Efraimidis [48] combine federated learning with neural collaborative filtering [29] to improve privacy. However, since federated learning could still leak user data [47, 50], research proposes to learn a user's local model by utilizing only a subset of the rating data [4, 14, 46]. Moreover, *differential privacy (DP)* [18, 20] has been leveraged for collaborative filtering recommender systems [10–12, 24, 60, 65]. These techniques add randomness to users' data to hide the actual data. Therefore, they face a trade-off between accuracy and privacy (e.g., [7]). To address this trade-off, Xin and Jaakkola [61] assume a moderate number of public users who tolerate disclosing their rating data. With this unprotected rating data, recommendation accuracy can be preserved while the privacy requirements of the remaining users are respected.

It has been shown in several studies [1, 39, 43] that users often receive more recommendations for popular items, and correspondingly non-popular items receive less exposure. This behavior of recommender systems, which is referred to as *popularity bias*, leads to disparate, i.e., unfair, treatment of less popular items. Dwork et al. [19] and Zemel et al. [62] show that, formally, there is a close connection between fairness and DP. However, the sole application of DP is insufficient to ensure fairness due to correlations within the dataset [21]. Moreover, Ekstrand et al. [21] and Agarwal [3] highlight a trade-off between user privacy and fairness. Overall, related work suggests that DP can severely impact recommendations in different ways, for example, result in popularity

Table 1. Overview of the Notation Used in this Article

Symbol	Description
k	Number of neighbors to process a rating query for target user u and target item i.
Q_u	Rating queries for target user u, i.e., the items in u's test set R_u^{test}.
$\mathcal{R}^k$	User-based *KNN* recommender system utilizing k neighbors to predict ratings.
$\mathcal{R}^k(u,i)$	Estimated rating score for target user u and target item i by recommender system $\mathcal{R}^k$.
$\mathcal{R}^k_{top}(u)$	Items with the highest estimated rating score for target user u.
$r_{u,i}$	Rating score of user u to item i.
U	The set of users.
U_i	The set of users that rated item i.
I	The set of items.
I_u	The set of items rated by user u.
R	The set of ratings.
$N^k_{u,i}$	The k nearest neighbors for target user u and target item i.
$N_{u,i}$	Neighbors of target user u and rated item i.
N_u	The set of neighbors for target user u across all rating queries.
$N_u^{(q)}$	The set of neighbors for target user u across q rating queries.
$sim(u,n)$	Similarity score between target user u and neighbor n.
$reusability(c\|u)$	Reusability score of candidate neighbor c for target user u.
$ranking(\cdot)$	The ranking function that ranks candidate neighbors w.r.t. similarity and reusability.
τ	Data usage threshold, i.e., the maximal usage of a user's data that is tolerated.
m_{DP}	Differential privacy mechanism that utilizes plausible deniability.
ϵ	Privacy parameter.
S	Secure users that do not need to be protected with DP.
V	Vulnerable users that need to be protected with DP.
R_S	Rating data of secure users.
$\tilde{R}_V$	DP-protected rating data of vulnerable users.
α	Significance level used for the statistical tests.
σ_x	Sample standard deviation of variable x.
$\sigma_{x,y}$	Sample covariance of variables x and y.

bias. Therefore, we believe that it is important to evaluate the proposed approach, *ReuseKNN*, also in terms of item popularity bias.

Similar to our work, previous research by Zhu et al. [65] prevents the inference of neighbors' rating data by applying DP to the users' rating data in *UserKNN*. However, to preserve recommendation accuracy, Zhu et al. vary the degree of randomness that is added to all users' rating data based on the sensitivity of the data. In contrast, *ReuseKNN* preserves recommendation accuracy by adding randomness only where it is necessary, i.e., to vulnerable users with a high privacy risk. In the remainder of the article, we use a variant of the approach of Zhu et al. that is comparable to the proposed approach as baseline (i.e., $UserKNN_{DP}^{full}$) for our experiments.

3 PROBLEM DEFINITION

In the following, we discuss one key vulnerability of *UserKNN*, which poses privacy risks to the neighbors utilized in the recommendation process. Also, we precisely model the adversary's goal, i.e., the inference of the neighbors' rating data. A summary of the notation used in this article is given in Table 1.

3.1 Vulnerability Analysis of UserKNN

Typically, a user-based *KNN* recommender system $\mathcal{R}^k$, i.e., *UserKNN*, generates an estimated rating score for a rating query of a target user u and a target item i by utilizing the ratings of k other

users that have rated i, i.e., the k nearest neighbors $N^k_{u,i}$:

$$\mathcal{R}^k(u,i) = \frac{\sum_{n \in N^k_{u,i}} sim(u,n) \cdot r_{n,i}}{\sum_{n \in N^k_{u,i}} sim(u,n)}, \tag{1}$$

where $sim(u,n)$ is the similarity between target user u and neighbor n, commonly determined via Pearson's correlation coefficient [6] or Cosine similarity between the users' rating vectors. For *UserKNN*, the neighborhood $N^k_{u,i}$ used for generating recommendations for target user u and item i, comprises the k most similar neighbors:

$$N^k_{u,i} = \arg\max_{c \in U_i}{}^{k}\, sim(u,c), \tag{2}$$

where U_i are all users that have rated i and sim is the similarity metric. *UserKNN* utilizes the rating data of the target user's k nearest neighbors to generate an estimated rating score (see Equation (1)). Therefore, the estimated rating score $\mathcal{R}^k(u,i)$ for target user u and item i is linked to the neighbors' rating data. Through learning the behavior of *UserKNN*, the estimated rating score could reveal the rating data of users that have been used as neighbors [9]. Therefore, the privacy threat for users can be traced back to them being utilized as neighbors in the recommendation process.

3.2 Attack Model

In this work, we assume that a user with malicious intent, i.e., the *adversary* a, exploits the vulnerability above via querying estimated rating scores from the recommender system, i.e., $\mathcal{R}^k(a) = \{\mathcal{R}^k(a,i_1), \mathcal{R}^k(a,i_2), \ldots, \mathcal{R}^k(a,i_l)\}$, where $\mathcal{R}^k(a,i_j)$ is the estimated rating score for item $i_j \in Q_a$ and Q_a is the set of a's queries. The adversary a can target a specific user n by increasing the likelihood of n being used as neighbor. To achieve this, a would modify its own user profile R_a such that it (partially) matches n's profile. Moreover, a can exploit publicly available data P, e.g., public rating data, product reviews, tweets, or lists of similar items, to better learn the behavior of *UserKNN* [9]. Given these assumptions, the adversary aims to infer the rating data of a neighbor n used to generate the estimated rating scores:

$$Pr[r_{n,i_1}, r_{n,i_2}, \ldots, r_{n,i_l} | \mathcal{R}^k(a,i_1), \mathcal{R}^k(a,i_2), \ldots, \mathcal{R}^k(a,i_l), P \cup R_a], \tag{3}$$

where r_{n,i_j} is the rating score of neighbor n for item i_j. Note that if a user is used as neighbor for many rating queries, many ratings could be targeted by an adversary. Thus, the degree to which a user's rating data is used in the recommendation process is an important indicator of this user's privacy risk (see the DataUsage@k metric in Section 5.2.3).

Given this attack model, the privacy threat lies on the rating level, i.e., the inference of neighbors' rating scores. Therefore, our approach aims at protecting the neighbors' rating scores. In the remainder of this work, we evaluate our approach in a rating-prediction task, since this fits well to our problem statement above (see Appendix B for results of a ranking-based experiment).

4 APPROACH

In the following, we first schematically illustrate *UserKNN*'s and *ReuseKNN*'s recommendation process based on an illustrative example. Then, we outline the two neighborhood reuse strategies of the *ReuseKNN* recommender system (Section 4.2). Finally, we present *ReuseKNN*$_{DP}$, i.e., neighborhood reuse with *differential privacy (DP)* (Section 4.3).

4.1 Example of the Recommendation Process in *UserKNN* and *ReuseKNN*

Figure 2 provides a schematic illustration of *UserKNN*'s and *ReuseKNN*'s recommendation process, showing the interplay between a user's data usage and the user's privacy risk. For simplicity, we

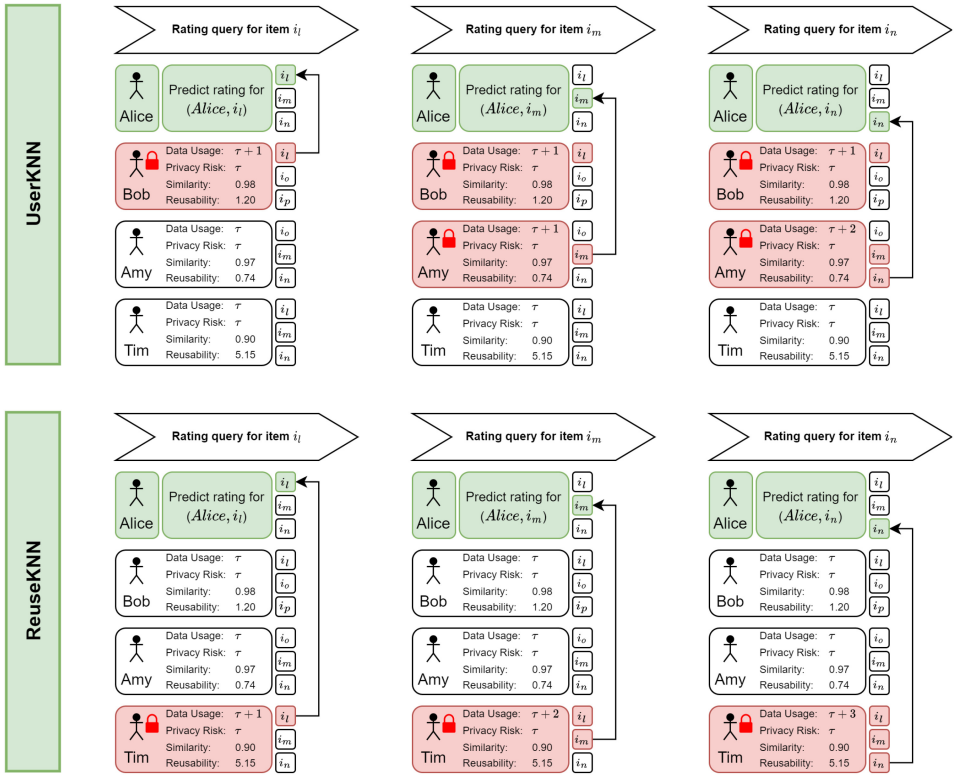

Fig. 2. Schematic illustration of the recommendation process for three rating queries in Alice's query set Q_{Alice} for *UserKNN* and *ReuseKNN*. A green shaded item indicates that the rating score for this item is estimated for the target user and a red shaded item indicates that the rating score of a neighbor has been utilized for the rating estimation. Traditional *UserKNN* selects those users as neighbors that rated the queried item and have the highest similarity value; in this toy example, those are Bob and Amy. Thus, Bob and Amy are vulnerable and need to be protected with DP. In contrast, *ReuseKNN* utilizes Tim as neighbor. As such, *ReuseKNN* substantially increases reusability (5.15 instead of 1.2 and 0.74) at the price of a slightly reduced similarity (0.90 instead of 0.98 and 0.97). This way, only Tim is vulnerable and is the only neighbor that needs to be protected with DP, as Bob and Amy remain unused.

assume that Bob, Amy, and Tim have been used as neighbors for τ rating queries, i.e., data usage and privacy risk is τ. To process Alice's rating queries for items i_l and i_m, *UserKNN* selects Bob and Amy as neighbors, as they have the highest similarity values across all users that rated the queried items. Due to the usage of Bob's and Amy's data, their data usage exceeds threshold τ and DP needs to be applied. For the rating query for item i_n, again, Amy is utilized in the recommendation process. Since she is already protected with DP, her privacy risk remains at τ. This is different from how *ReuseKNN* processes rating queries. For the rating queries for items i_l, i_m, and i_n, *ReuseKNN* selects Tim as neighbor, as Tim has a substantially higher reusability value and only marginally smaller similarity than Bob and Amy. Therefore, only Tim's data usage exceeds τ, and DP is needed to protect Tim.

In summary, in this illustrative example, *UserKNN* leads to two vulnerable users, Bob and Amy, that need to be protected with DP. In contrast, *ReuseKNN* leads to only one vulnerable user, Tim, to which DP has to be applied.

4.2 ReuseKNN

The key feature of *ReuseKNN* is to reuse neighbors from a target user u's previous rating queries to minimize the cardinality of the neighborhood $N_u = \bigcup_{i \in Q_u} N^k_{u,i}$ across all rating queries Q_u. As illustrated in Figure 1, this means that *ReuseKNN* decreases the data usage for most users, i.e., secure users, and in this way, also their privacy risk. Plus, *ReuseKNN* decreases the number of highly reused neighbors, i.e., vulnerable users with high data utilization and, thus, high privacy risk.

In addition to the similarity, *ReuseKNN* also considers the extent to which a target user u could reuse candidate neighbor c as a neighbor for many rating queries, i.e., $reusability(c|u)$. Since both similarity and reusability scores are differently distributed across their respective numeric ranges, we rank candidate neighbors according to their scores. Formally, for a user u, the rank $ranking(u) = |\{v \in U \setminus \{u\} : f(v) \leq f(u)\}|$, where U is the set of all users and f measures the $similarity$ or $reusability$ score. Note that $ranking(u) > ranking(v)$ if $f(u) > f(v)$ for users u and v, and that $ranking(u) = ranking(v)$ in case $f(u) = f(v)$. With this, the k neighbors $N^k_{u,i}$ are selected based on similarity and reusability. Formally:

$$N^k_{u,i} = \underset{c \in U_i}{\arg\overset{k}{\max}} \left[ranking(sim(u,c)) + ranking(reusability(c|u))\right], \tag{4}$$

where U_i are all users that rated item i, sim measures the similarity between two users, and $reusability$ depends on the given neighborhood reuse strategy of *ReuseKNN*. In the case in which multiple candidate neighbors have equal values for $ranking(sim(u,c)) + ranking(reusability(c|u))$, we choose these neighbors at random.

To estimate a candidate neighbor's $reusability$ score, *ReuseKNN* utilizes two neighborhood reuse strategies: *Expect* and *Gain*. The unpersonalized *Expect* strategy measures a candidate neighbor's reusability for an average target user, whereas the personalized *Gain* strategy measures the reusability for a specific target user. Next, we discuss two strategies to increase the reusability of a target user's neighbors: unpersonalized and personalized neighborhood reuse.

Unpersonalized Neighborhood Reuse: Expect. The more users rated an item, the more likely it is that a random target user will query a rating estimation for this item. Following this intuition, *Expect* promotes candidate neighbors that rated many popular items and penalizes candidate neighbors that either rated only a few items or many unpopular items. For *Expect*, the reusability score of candidate neighbor c is defined by

$$reusability(c|u) = reusability(c) = \sum_{i \in I_c} \frac{|U_i|}{|U|}, \tag{5}$$

where u is the target user, I_c are the items c rated, U_i are the users that rated an item i, and U is the set of all users. In this case, $reusability(c)$ is the summed-up popularity of c's rated items and measures the *expected* number of a random user's rating queries for which c could be used as a neighbor. This means that the reusability of a candidate neighbor is estimated for an average user and not for a specific target user (i.e., unpersonalized).

Personalized Neighborhood Reuse: Gain. In contrast to unpersonalized neighborhood reuse, *Gain* measures a candidate neighbor's reusability for a specific target user. Specifically, *Gain* quantifies how many of a target user's ratings a candidate neighbor could have covered in the past, i.e., how many ratings the target user could have *gained* from the candidate neighbor. Thus, *Gain* gives the fraction of a target user u's rated items for which a candidate neighbor c could have served as a neighbor:

$$reusability(c|u) = \frac{|I_u \cap I_c|}{|I_u|}, \tag{6}$$

where I_u are the items rated by u and I_c are the items rated by c. In contrast to the unpersonalized *Expect* strategy, *Gain*'s reusability score depends on a specific target user (i.e., personalized).

4.3 ReuseKNN$_{DP}$: Neighborhood Reuse and Differential Privacy

ReuseKNN leads to a minimal number of highly reused neighbors, i.e., vulnerable users, who are utilized more often as neighbors than the data usage threshold τ would allow. *ReuseKNN*$_{DP}$ addresses this high privacy risk resulting from the frequent usage of vulnerable users (see Section 3) by adding DP to our neighborhood reuse strategies. Specifically, for a rating query for user u and item i, a privacy mechanism m_{DP} is applied to the ratings for i of vulnerable users V that are used as neighbors, i.e., $\tilde{R}_V = \{m_{DP}(r_{n,i}) : n \in N^k_{u,i} \cap V\}$. In this way, *ReuseKNN*$_{DP}$ utilizes real ratings of secure users S, i.e., $R_S = \{r_{n,i} : n \in N^k_{u,i} \cap S\}$, plus the modified ratings $\tilde{R}_V$ of vulnerable users, to generate the estimated rating score $\mathcal{R}^k(u,i)$:

$$\mathcal{R}^k(u,i) = \frac{\sum_{n \in N^k_{u,i} \cap S} sim(u,n) \cdot r_{n,i} + \sum_{n \in N^k_{u,i} \cap V} sim(u,n) \cdot m_{DP}(r_{n,i})}{\sum_{n \in N^k_{u,i}} sim(u,n)}. \tag{7}$$

Specifically, the privacy mechanism m_{DP} utilizes *randomized responses* [59] to achieve DP [20]. With this, intuitively, neighbors can plausibly deny that their real rating was used in the recommendation process. The privacy mechanism m_{DP} flips a fair coin and if the coin is heads, the vulnerable neighbor's real rating is utilized in the recommendation process. If the coin is tails, m_{DP} flips a second fair coin to decide whether to utilize the vulnerable neighbor's real rating or a random rating drawn from a uniform distribution over the range of ratings. With this, the adversary is unaware whether the utilized rating is real, or random, which leads to the privacy guarantees within the DP framework [20]:

$$\frac{Pr[\text{Adversary's assumption: Real rating} \mid \text{Truth: Real rating}]}{Pr[\text{Adversary's assumption: Real rating} \mid \text{Truth: Random rating}]} = \frac{0.75}{0.25} = 3 \leq e^{\epsilon}, \tag{8}$$

which results in a privacy parameter of $\epsilon = \ln 3$. Reconsidering user-based *KNN*'s vulnerability (see Equation (1)), this means that if a neighbor n is considered as vulnerable, the DP-protected rating is used in the recommendation process instead of the real rating for item i (see Equation (7)). This impacts the adversary a's objective (see Equation (3)) of inferring n's rating data given the estimated rating scores for which n was used as neighbor and its own rating data R_a in combination with public knowledge P (see Section 3). Since a maximum of τ (i.e., the data usage threshold) real ratings of n are used by the recommender system, the remaining ratings are DP-protected. Thus, the adversary is not aware of whether the inferred rating data is the original rating data or random rating data as generated by the m_{DP} mechanism:

$$Pr[r_{n,i_1}, \ldots, r_{n,i_\tau}, m_{DP}(r_{n,i_{\tau+1}}), \ldots, m_{DP}(r_{n,i_l}) | \mathcal{R}^k(a,i_1), \mathcal{R}^k(a,i_2), \ldots, \mathcal{R}^k(a,i_l), P \cup R_a], \tag{9}$$

where r_{n,i_j} is n's rating for item i_j and $\mathcal{R}^k(a,i_j)$ is the estimated rating score of i_j for adversary a. Through combining non-DP and DP ratings, *ReuseKNN*$_{DP}$ yields the following privacy parameter ϵ for each of a vulnerable user's, in this case n, utilized ratings (for details, see Appendix A):

$$\epsilon = \ln\left(3 + 4 \cdot \frac{Pr[\text{Non-DP rating}]}{Pr[\text{DP rating}]}\right). \tag{10}$$

In this way, *ReuseKNN*$_{DP}$ combines neighborhood reuse with DP to reduce the number of neighbors to which DP needs to be applied and to ensure privacy. Overall, *ReuseKNN*$_{DP}$ can use two neighborhood reuse strategies with DP (for details, see Section 4.2):

(1) *Expect*$_{DP}$: Unpersonalized neighborhood reuse combined with DP
(2) *Gain*$_{DP}$: Personalized neighborhood reuse combined with DP

5 EXPERIMENTAL SETUP

We utilize a two-stage evaluation procedure to compare and evaluate the two neighborhood reuse strategies of (i) *ReuseKNN* and (ii) $ReuseKNN_{DP}$:

Neighborhood Reuse without DP: ReuseKNN. In the first stage, we evaluate *ReuseKNN* without protecting vulnerable neighbors with DP in order to better understand the advantages and disadvantages of the proposed neighborhood reuse strategies. Hence, we compare *Expect* and *Gain* to distill the impact of neighborhood reuse for recommendations.

Neighborhood Reuse with DP: $ReuseKNN_{DP}$. In the second stage, we combine *ReuseKNN* with DP to protect vulnerable users, i.e., $ReuseKNN_{DP}$. We compare our neighborhood reuse strategies $Expect_{DP}$ and $Gain_{DP}$ to investigate how $ReuseKNN_{DP}$ can address the accuracy–privacy trade-off.

5.1 Baselines

We compare *ReuseKNN* and $ReuseKNN_{DP}$ with five different KNN-based baselines. Concretely, for *ReuseKNN*, i.e., neighborhood reuse without DP, we use two non-DP baselines:

(1) *UserKNN* [30]: Traditional *UserKNN* without neighborhood reuse. No users are protected with DP (Vulnerable users $V = \emptyset$).
(2) *UserKNN+Reuse*: A variant of *UserKNN* with neighborhood reuse. Initially, for the first rating query, e.g., for item j, the k most similar users that rated j are selected as neighbors, as in case of *UserKNN*. However, for the following rating queries, e.g., for item i and user u, $k^{prev} = \min\{k, |N_{u,i}|\}$ neighbors from all previous rating queries that rated i (i.e., $N_{u,i}$) are reused. If too few previous neighbors rated i, i.e., $k^{prev} < k$, a minimal set of $k^{new} = k - k^{prev}$ new neighbors is additionally used, as given by:

$$N^k_{u,i} = \underset{n \in N_{u,i}}{\arg\max^{k^{prev}}} sim(u,c) \cup \underset{c \in U_i \setminus N_{u,i}}{\arg\max^{k^{new}}} sim(u,c), \tag{11}$$

where U_i are all users that rated item i. Similar to *UserKNN*, *UserKNN+Reuse* assumes that no users are vulnerable ($V = \emptyset$). Thus, no users are protected with DP.

For $ReuseKNN_{DP}$, i.e., neighborhood reuse with DP, we use three DP baselines:

(1) $UserKNN_{DP}$: A variant of *UserKNN*, but DP is applied to vulnerable users $V = \{u \in U : \text{DataUsage@}k(u) > \tau\}$. See Section 5.5 for the exact τ values.
(2) $UserKNN{+}Reuse_{DP}$: A variant of *UserKNN+Reuse*, but DP is applied to vulnerable users $V = \{u \in U : \text{DataUsage@}k(u) > \tau\}$. See Section 5.5 for the exact τ values.
(3) $UserKNN^{full}_{DP}$: Traditional differentially private *UserKNN*, where DP is applied to the full set of users, i.e., $V = \{u \in U : \text{DataUsage@}k(u) \geq 0\}$ (similar to the rating perturbation in [65]).

To evaluate $ReuseKNN_{DP}$, we use the three DP baselines, as well as non-DP *UserKNN*. With this, we can compare $ReuseKNN_{DP}$ to two contrastive baselines: $UserKNN^{full}_{DP}$, which protects all users with DP, and *UserKNN*, which does not apply DP at all.

5.2 Evaluation Metrics

We test the proposed approach in two evaluation stages using the following evaluation criteria and metrics (see Table 2 for an overview):

5.2.1 Neighborhood Reuse. To evaluate the degree to which *ReuseKNN* can reuse neighbors from previous rating queries, we measure the size of a target user's neighborhood after multiple queries. Plus, we study whether the reused neighborhoods are capable of generating meaningful

Table 2. Overview of the Seven Evaluation Metrics Used in this Work

Evaluation Criterion	Evaluation Metric	Objective	Short Description	Evaluation Stage	
				ReuseKNN	*ReuseKNN*$_{DP}$
Neighborhood Reuse	Neighbors@q	↘	Neighborhood size	•	
	CoRatings@q	↗	No. of co-rated items	•	
Accuracy	MAE@k	↘	Mean absolute error	•	•
Privacy	$\|V\|$	↘	Percentage of vulnerable users	•	
	PrivacyRisk@k	↘	Privacy risk of users		•
Popularity Bias	PP-Corr@k	↘	Positivity–popularity correlation		•
	Coverage@k	↗	Percentage of item coverage		•

↘ indicates that lower values are better and ↗ indicates that higher values are better. q is the number of queries and k is the number of neighbors. With •, we indicate the evaluation stage in which the metric is used.

recommendations via measuring the number of co-rated items between the neighborhood and the target user.

Neighborhood Size. For every rating query of a target user u, k neighbors are required to generate the recommendation. In the worst case, no neighbors from previous rating queries can be reused. Thus, after q queries, $|N_u| = \min\{q \cdot k, |U| - 1\}$ for U being the set of all users. In the best case, u reuses the same k neighbors for all q queries, i.e., $|N_u| = k$. To quantify how many of u's neighbors are reused, we measure the size of u's neighborhood after q rating queries:

$$\text{Neighbors@}q(u) = \left|N_u^{(q)}\right|, \tag{12}$$

where $N_u^{(q)}$ is u's set of neighbors after q rating queries. With that, we test how well our neighborhood reuse strategies of *ReuseKNN*, i.e., neighborhood reuse only, can reuse a target user's neighbors for multiple rating queries.

Number of Co-Ratings. The utilization of fewer neighbors across many rating queries might impact the accuracy of recommendations. Therefore, we test whether a target user's neighbors are beneficial for recommendation accuracy, i.e., "reliable". One important characteristic of these reliable neighbors is the number of co-rated items with the target user [2, 16]. Thus, we measure the average number of co-rated items between a target user u and its neighbors $n \in N_u$ after q rating queries:

$$\text{CoRatings@}q(u) = \frac{1}{|N_u^{(q)}|} \sum_{n \in N_u^{(q)}} |I_u \cap I_n|, \tag{13}$$

where I_u are the items rated by target user u and I_n are the items rated by neighbor n. With this, we test how beneficial the neighborhoods are for generating accurate recommendations.

5.2.2 Accuracy. To quantify the accuracy of a target user's recommendations, we rely on the widely used mean absolute error metric (MAE). We use MAE to measure how accurate the rating scores can be predicted, because of the way in which we apply DP, i.e., via adding noise to the neighbors' rating values in order to protect against the disclosure of these ratings (see Section 3). According to Herlocker et al. [30], the number of neighbors k has an impact on the recommendation accuracy. Thus, we test the accuracy of u's recommendations for $k \in \{5, 10, 15, 20, 25, 30\}$. Therefore, MAE@$k(u)$ quantifies the accuracy of u's recommendations when k neighbors are used to generate a recommendation. More formally:

$$\text{MAE@}k(u) = \frac{1}{|R_u^{test}|} \sum_{r_{u,i} \in R_u^{test}} |r_{u,i} - \mathcal{R}^k(u,i)|, \tag{14}$$

where the predicted rating score $\mathcal{R}^k(u,i)$ is compared with the real rating scores $r_{u,i} \in R_u^{test}$ in u's test set. We note that the items for which R_u^{test} comprises ratings are the ones that are in u's set of rating queries Q_u. We use the MAE@$k(u)$ metric for evaluating both, *ReuseKNN*, i.e., neighborhood reuse only, and *ReuseKNN*$_{DP}$, i.e., neighborhood reuse with DP.

5.2.3 Privacy. Liu and Terzi [42] provide a framework to measure a user's privacy risk in computational systems, such as recommender systems based on the visibility of the user's data. In our work, we relate this visibility to how often a user's rating data was utilized in the recommendation process. As such, the DataUsage@$k(u)$ metric counts for how many rating queries a user u was used as a neighbor. Similar to MAE@$k(u)$, we also relate the usage of u's data to the number of neighbors k used to generate recommendations. Formally:

$$\text{DataUsage@}k(u) = \sum_{v \in U} \sum_{i \in Q_v} \mathbb{1}_{N_{v,i}^k}(u), \tag{15}$$

where U is the set of all users, Q_v is the set of items for which user v queries estimated ratings, and $\mathbb{1}_{N_{v,i}}(u)$ is the indicator function of user u being in v's set of neighbors $N_{v,i}$ for an item i.

Percentage of Vulnerable Users. As mentioned earlier, the main goal of neighborhood reuse is to reduce the number of users that need to be protected with DP. The DataUsage@k definition allows us to identify these vulnerable users V, i.e., the set of users whose data is utilized more often than the adjustable privacy risk threshold τ allows:

$$V = \{u \in U : \text{DataUsage@}k(u) > \tau\}, \tag{16}$$

where U is the set of all users. Thus, the percentage of vulnerable users relates to what fraction of users DP has to be applied to (i.e., $|V|/|U|$). We use this metric to evaluate *ReuseKNN*, i.e., neighborhood reuse only.

Privacy Risk. We apply DP to a user u's data if DataUsage@$k(u) > \tau$. This way, only the first τ utilized ratings contribute to u's privacy risk, since for the remaining ratings that are utilized, privacy is guaranteed via the DP framework (see Section 4.3):

$$\text{PrivacyRisk@}k(u) = \min[\tau, \text{DataUsage@}k(u)]. \tag{17}$$

We use PrivacyRisk@k to measure the users' privacy risk when neighborhood reuse is combined with DP, i.e., *ReuseKNN*$_{DP}$.

5.2.4 Item Popularity Bias. One might be concerned that neighborhood reuse could lead to exploiting users as neighbors that rated many popular items, which could result in more positive estimated rating scores for popular items. To test for this item popularity bias, we analyze all items for which the recommender system estimates high rating scores, i.e., "top items". For a recommender system model $\mathcal{R}$ and k neighbors, a user u's set of top items is given by $\mathcal{R}_{top}^k(u) = \arg\max_{i \in Q_u}^{n} \mathcal{R}^k(u,i)$, where Q_u are the items in u's query set. In our case, we set $n = 10$.

Positivity-Popularity Correlation. To study whether higher estimated rating scores are given to popular items, we follow Kowald et al. [39] and correlate an item's popularity with its occurrences in users' sets of top items: ItemFreq$^+$@$k(i) = \sum_{u \in U} \mathbb{1}_{\mathcal{R}_{top}^k(u)}(i)$, where $\mathbb{1}_{\mathcal{R}_{top}^k(u)}(i)$ indicates whether item i is in user u's set of top items $\mathcal{R}_{top}^k(u)$. Plus, an item i's popularity is given by ItemPop(i) $= |U_i|/|U|$, where U is the set of all users and U_i are the users that rated i. We compute the Pearson correlation coefficient [6] between the two variables ItemFreq$^+$ and ItemPop to identify item popularity bias:

$$\text{PP-Corr@}k = \frac{\sigma_{\text{ItemFreq}^+\text{@}k,\,\text{ItemPop@}k}}{\sigma_{\text{ItemFreq}^+\text{@}k} \cdot \sigma_{\text{ItemPop@}k}}, \tag{18}$$

Table 3. Descriptive Statistics of the Five Datasets

Dataset	Domain	Rating range	$\lvert U\rvert$	$\lvert I\rvert$	$\lvert R\rvert$	$\lvert R\rvert/\lvert U\rvert$	$\lvert U\rvert/\lvert I\rvert$	Density
ML 1M	Movies	{1…5}	6,040	3,706	1,000,209	165.60	1.6298	4.47%
Douban	Movies	{1…5}	2,509	39,576	893,575	356.15	0.0634	0.90%
LastFM	Music	{1…1,000}	3,000	352,805	1,755,361	585.12	0.0085	0.17%
Ciao	Movies	{1…5}	7,375	105,096	282,619	38.32	0.0702	0.04%
Goodreads	Books	{1…5}	20,000	508,696	2,569,177	128.46	0.0394	0.03%

$\lvert U\rvert$ is the number of users, $\lvert I\rvert$ is the number of items, $\lvert R\rvert$ is the number of ratings, $\lvert R\rvert/\lvert U\rvert$ is the ratings-to-users ratio, $\lvert U\rvert/\lvert I\rvert$ is the users-to-items ratio, and Density is given by $\lvert R\rvert/(\lvert U\rvert \cdot \lvert I\rvert)$.

where $\sigma_{\text{ItemFreq}^+@k,\text{ItemPop}@k}$ is the sample covariance between ItemFreq$^+$@k and ItemPop@k. The sample standard deviations are given by $\sigma_{\text{ItemFreq}^+@k}$ and $\sigma_{\text{ItemPop}@k}$.

Item Coverage. In addition to evaluating the correlation between an item's estimated rating score and its popularity, we measure the fraction of items that are a top item for at least one user. For this, we use the Item Coverage metric [31] given by

$$\text{Coverage}@k = \frac{1}{|I|} \left| \bigcup_{u \in U} \mathcal{R}^k_{top}(u) \right|, \tag{19}$$

where k is the number of neighbors, I is the set of items, U is the set of users, and $\mathcal{R}^k_{top}(u)$ is the set of top items for user u. This way, we can test whether parts of the item catalog always receive low estimated rating scores. We use PP-Corr@k and Coverage@k to evaluate *ReuseKNN*$_{DP}$. Additionally, we use these metrics to evaluate *UserKNN* to explore the impact of DP [21].

5.3 Datasets

In this work, we conduct experiments on five different datasets: *MovieLens 1M* (*ML 1M*) [28], *Douban* [34], *LastFM User Groups* (*LastFM*) [39], *Ciao* [26], and *Goodreads* [57, 58].

All five datasets exhibit different properties, as illustrated in Table 3. For example, the movie rating dataset *ML 1M* (integer ratings in $\{1 \ldots 5\}$) is the densest dataset. Similarly, *Douban* (integer ratings in $\{1 \ldots 5\}$) and *Ciao* (integer ratings in $\{1 \ldots 5\}$) are movie rating datasets. Moreover, in *Ciao*, users have the smallest number of ratings per user (i.e., $|R|/|U|$) on average. *LastFM* includes implicit feedback data (i.e., listening counts) from the online music streaming service Last.fm. However, in this dataset, Kowald et al. [39] transfer the implicit feedback to decimal ratings in $\{1 \ldots 1,000\}$. Plus, users have the largest number of ratings per users. The book rating dataset *Goodreads* (integer ratings in $\{1 \ldots 5\}$), for which we use a random sample of 20,000 users, is the largest and least dense dataset.

Overall, the datasets cover (i) the movie, music, and book domain; (ii) implicit and explicit feedback; and (iii) different descriptive statistics.

5.4 Evaluation Protocol and Statistical Tests

We perform all experiments using 5-fold cross-validation, and randomly split all folds into 80% training sets R^{train} and 20% test sets R^{test}. The ratings in R^{train} are used to train the recommendation algorithms, and the ratings in R^{test} represent the rating queries used for evaluation. Also, we test the statistical significance of our results. Specifically, after close inspection of our results, we resort to the Mann-Whitney-U-Test. For the query-based metrics Neighbors@q and CoRatings@q, we evaluate significance for all rating queries $q \in [2; 100]$ when utilizing $k = 10$ neighbors. For other metrics, i.e., MAE@k, PrivacyRisk@k, PP-Corr@k, and Coverage@k, we evaluate significance after all queries have been processed by the recommender system. Again,

here, we utilize $k = 10$ neighbors to generate recommendations. Importantly, throughout this work, we only report statistical significance if we observe significance for each of the five folds.

5.5 Parameter Settings

The proposed approach relies on two adjustable hyperparameters: (i) the number of neighbors k used in the recommendation process and (ii) the data usage threshold τ. To test the performance of *ReuseKNN* and $ReuseKNN_{DP}$ for different values of k, we vary $k \in \{5, 10, 15, 20, 25, 30\}$. Plus, we set τ to the approximate starting value of the tail of *UserKNN*'s data usage distribution DataUsage@k, which is given by its maximal second derivative (see Figure 1). This way, we assume that only the tail's small privacy risk (as a result of the rare data usage) is tolerable and give an example of how τ can be defined by the recommender system provider. Also, τ is the same for all users. This leads to the following τ values for $k = 10$: 92.89 (ML 1M), 91.54 (Douban), 104.32 (LastFM), 95.79 (Ciao), and 94.90 (Goodreads). For the similarity function *sim*, we use cosine similarity.

6 RESULTS AND DISCUSSION

We structure our results into two parts: (i) neighborhood reuse only (*ReuseKNN*), and (ii) neighborhood reuse with DP ($ReuseKNN_{DP}$).

6.1 ReuseKNN

In this section, we present our evaluation results for *ReuseKNN*, i.e., neighborhood reuse only.

6.1.1 Neighborhood Reuse. As the first step in this evaluation stage, neighborhood reuse only, we investigate the neighborhoods generated by *ReuseKNN*. Specifically, we compare our neighborhood reuse strategies to our *UserKNN* baseline with respect to the neighborhood size and the number of co-ratings with the target user. Moreover, we test for statistical significant differences to *UserKNN* after multiple rating queries, i.e., for all $q \in [2; 100]$.

We investigate the average size of target users' neighborhood after q rating queries for a model with $k = 10$ neighbors in Figure 3. For all of our five datasets, the size of a user's neighborhood increases more strongly for traditional *UserKNN* than for our neighborhood reuse strategies. For ML 1M, Douban, LastFM, and Goodreads, a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) shows that all our neighborhood reuse strategies yield significantly smaller neighborhoods than traditional *UserKNN* for $q \in [2; 100]$ rating queries. This means that *ReuseKNN* can already reuse neighbors after an initial neighborhood is generated for the very first rating query.

However, for Ciao, multiple initial rating queries are needed to generate reusable neighborhoods. Our neighborhood reuse strategies tend to yield significantly smaller neighborhoods only for a few rating queries. For *Gain*, we do not observe significant differences. We attribute this to the on average small user profiles in Ciao (see Table 3). Reusable neighbors are scarce and, thus, *ReuseKNN* cannot reduce the neighborhood size as effectively as in the case of the other datasets.

In addition to the neighborhood size, we also investigate the number of co-rated items between the target user and its neighbors after querying q rating queries (see Figure 4). Note that, as before, the statistical significance is evaluated after multiple rating queries, i.e., for all $q \in [2; 100]$. For all of our five datasets, our neighborhood reuse strategies can substantially increase the number of co-ratings over traditional *UserKNN*. A one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) reveals that our neighborhood reuse strategies generate neighborhoods with significantly more co-ratings with the target user than *UserKNN* for $q \in [2; 100]$ rating queries. This indicates that *ReuseKNN* generates neighborhoods with fewer neighbors that have more co-ratings with the target user than in the case of traditional *UserKNN*, which can foster recommendation accuracy [2, 16].

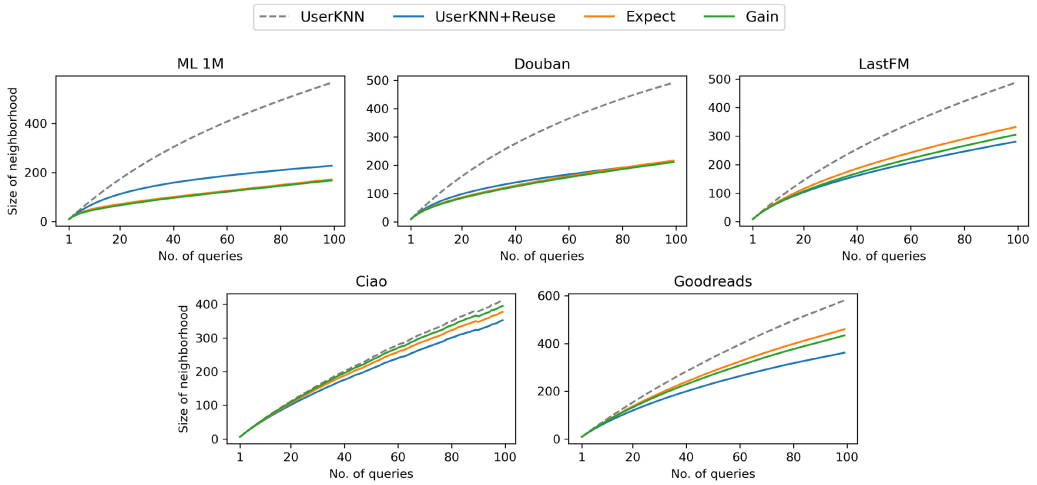

Fig. 3. Average number of neighbors per target user after q rating queries. Our neighborhood reuse strategies utilized in *ReuseKNN*, i.e., *Expect* and *Gain*, generate smaller neighborhoods than *UserKNN*.

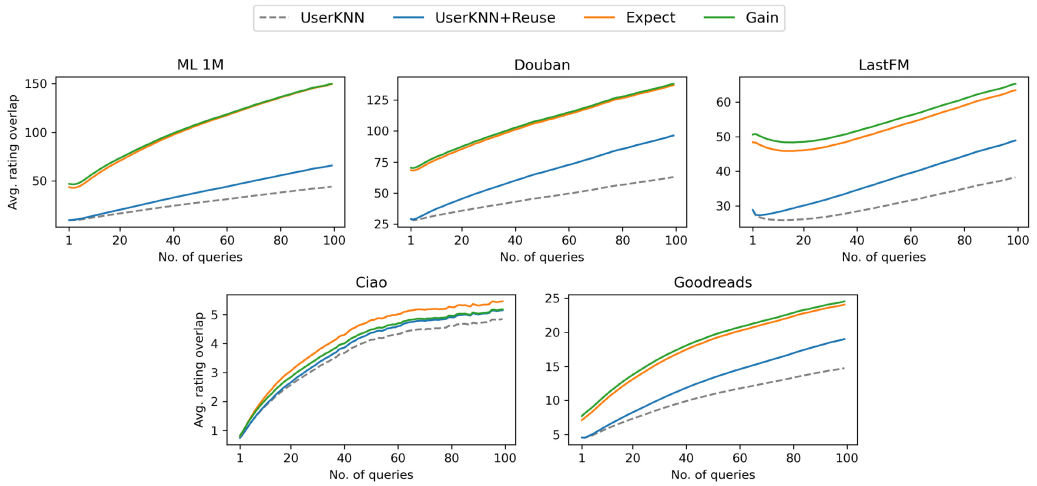

Fig. 4. Avg. number of co-rated items between the target user and its neighbors. Our neighborhood reuse strategies for *ReuseKNN*, i.e., *Expect* and *Gain*, generate neighborhoods, in which the neighbors' rated items overlap more with the target users' than in the case of *UserKNN*. With this, neighbors are beneficial for generating accurate recommendations.

However, for Ciao, our neighborhood reuse strategies tend to generate neighborhoods with significantly more co-ratings for only a few rating queries. As in our neighborhood size experiment, we attribute this to the small user profiles in Ciao, which makes neighborhood reuse less effective due to the scarcity of reusable neighbors.

6.1.2 Accuracy. Next, we compare *ReuseKNN* with traditional *UserKNN* in terms of recommendation accuracy (see Figure 5). Specifically, we test for statistically significant differences between our neighborhood reuse strategies and the *UserKNN* baseline.

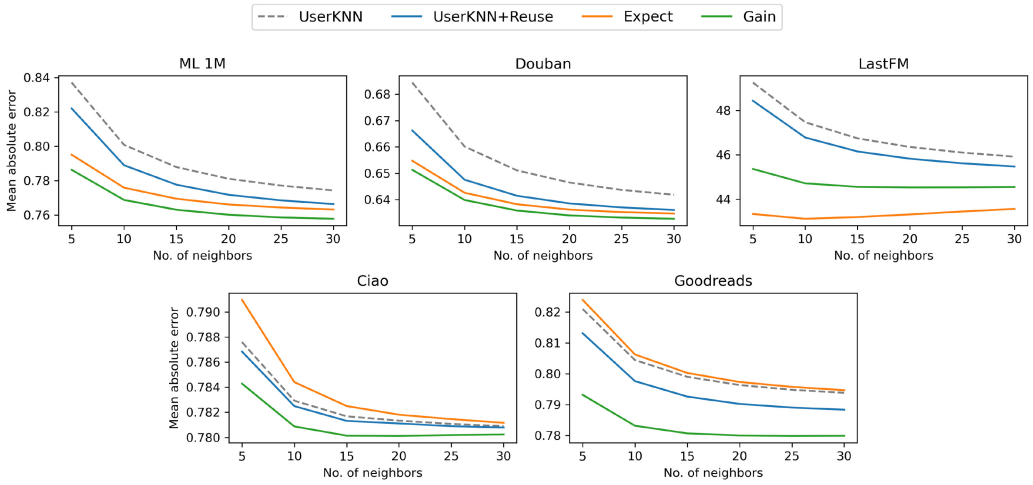

Fig. 5. Comparison of the recommendation accuracy between *ReuseKNN* and *UserKNN*. *ReuseKNN*'s neighborhood reuse strategies generate more accurate recommendations than *UserKNN*. For sparse datasets (i.e., Ciao and Goodreads), personalized neighborhood reuse (i.e., *Gain*) works better. In contrast, unpersonalized neighborhood reuse (i.e., *Expect*) works better for datasets, in which neighbors are scarce (i.e., LastFM).

We find that our neighborhood reuse strategies can generate more accurate recommendations than *UserKNN*. This shows that reusing neighbors that have already been used in the past can also lead to meaningful (accurate) recommendations in the future. Specifically, for ML 1M, Douban, and LastFM, a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) indicates that our neighborhood reuse strategies significantly increase recommendation accuracy for a model with $k = 10$ neighbors. Due to personalization, *Gain* performs best across most datasets.

For LastFM, unpersonalized neighborhood reuse (i.e., *Expect*) outperforms personalized neighborhood reuse (i.e., *Gain*). We attribute this to LastFM's small users-to-items ratio as compared with the other datasets (see Table 3), which makes it hard to identify neighbors, similar to an item–cold start scenario [52]. Concretely, in the case of personalized neighborhood reuse, selecting reusable neighbors for a specific target user reduces the pool of potential neighbors per item to a personalized subset and leads to a worse performance compared with unpersonalized neighborhood reuse. In contrast, unpersonalized neighborhood reuse allows using the entire pool of potential neighbors and, thus, achieves a higher accuracy for LastFM.

In the case of our least dense datasets Ciao and Goodreads, we observe that our personalized neighborhood reuse strategy *Gain* can handle these datasets better than our unpersonalized neighborhood reuse strategy *Expect*. *Gain* selects neighbors whose rating data could have been used by the target user in the past (see Equation (6)). This way, *Gain* creates a neighborhood for a given target user with sufficient rating data even in sparse datasets.

Plus, we highlight that *Gain* significantly increases recommendation accuracy for Goodreads despite the dataset's low density. In the case of Ciao, a two-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) reveals no significant differences between our neighborhood reuse strategies and *UserKNN* for $k = 10$, which suggests that all our neighborhood reuse strategies can preserve recommendation accuracy. As shown in our previous experiments (see Section 6.1.1), neighborhood reuse is less effective for Ciao due to the small user profiles. Thus, it makes sense that for Ciao, the recommendation accuracy cannot be improved as effectively as for the remaining datasets.

Table 4. Percentage of Vulnerable Users for a Model with $k = 10$ Neighbors

Method	ML 1M	Douban	LastFM	Ciao	Goodreads
UserKNN	80.39%	96.68%	99.89%	8.02%	65.00%
UserKNN+Reuse	84.64%	87.37%	98.90%	7.91%	52.29%
Expect	**24.13%**	**34.40%**	**68.20%**	**7.88%**	**29.12%**
Gain	25.09%	37.43%	80.28%	8.19%	40.51%

Best results, i.e., lowest values, are in **bold**. For all datasets, *ReuseKNN*'s *Expect* neighborhood reuse strategy leads to fewer vulnerable users than *UserKNN*. For Ciao, our neighborhood reuse strategies can achieve only minor improvements, as already *UserKNN* yields a small percentage of vulnerable users.

6.1.3 Percentage of Vulnerable Users. In Section 6.1.1, we found that neighborhood reuse can significantly reduce the number of neighbors that are utilized in the recommendation process. Now, however, we analyze how many neighbors are utilized for more than τ rating queries (i.e., the usage of their data exceeds threshold τ) and, thus, need to be protected with DP (see Table 4). Specifically, we compare our neighborhood reuse strategies to the *UserKNN* baseline.

For all of our five datasets, our neighborhood reuse strategies lead to less vulnerable users than traditional *UserKNN*. Especially, *Except* shows the best (i.e., lowest) percentage of vulnerable users. For example, for the ML 1M dataset, *UserKNN* leads to 80.39% of users that are vulnerable, since their data usage exceeds threshold $\tau = 92.89$ (see Section 5.5), whereas *Expect* leads to only 24.13% vulnerable users and, thus, fewer users need to be protected with DP.

For Ciao, our neighborhood reuse strategies achieve only minor improvements over *UserKNN*. The reason is that *UserKNN* already yields a small percentage of vulnerable users and, as such, *ReuseKNN* leads to only small improvements. Additionally, our previous findings show that the effect of neighborhood reuse on Ciao is smaller than on the remaining datasets due to the small average user profile size (see Table 3). This leads to a lack of reusable neighbors and, thus, also limits the effect that neighborhood reuse has on the percentage of vulnerable users.

6.1.4 Summary. Overall, we find that through neighborhood reuse, *ReuseKNN* can significantly reduce the size of target users' neighborhoods as compared with traditional *UserKNN*. Despite the much smaller neighborhoods, *ReuseKNN* identifies neighbors that have many more co-rated items with the target user than in the case of *UserKNN*. As related work suggests, these neighbors are more "reliable" and can be crucial for recommendation accuracy [2, 16].

Based on the much smaller but more reliable neighborhoods, *ReuseKNN* can provide significantly higher recommendation accuracy than traditional *UserKNN*. For sparse datasets, personalized neighborhood reuse seems to be a better solution than unpersonalized neighborhood reuse.

Plus, *ReuseKNN* can substantially reduce the percentage of vulnerable users, and in general, our *Except* neighborhood reuse method yields the fewest vulnerable users.

6.2 ReuseKNN$_{DP}$

Next, we present our results on *ReuseKNN*$_{DP}$, i.e., neighborhood reuse with DP.

6.2.1 Accuracy. First and foremost, we note that in our experiments without DP (see Figure 5), *UserKNN* could be outperformed by *ReuseKNN*. In our experiments with DP, however (see Figure 6), it is apparent that all evaluated DP methods do not reach the accuracy of non-DP *UserKNN*. This means that in general, due to DP, drops in recommendation accuracy have to be expected. However, we will investigate next whether *ReuseKNN*$_{DP}$ can make this accuracy drop less severe compared with using the baselines. In detail, we compare our neighborhood reuse strategies to the *UserKNN*$_{DP}$ baseline and test for statistically significant differences.

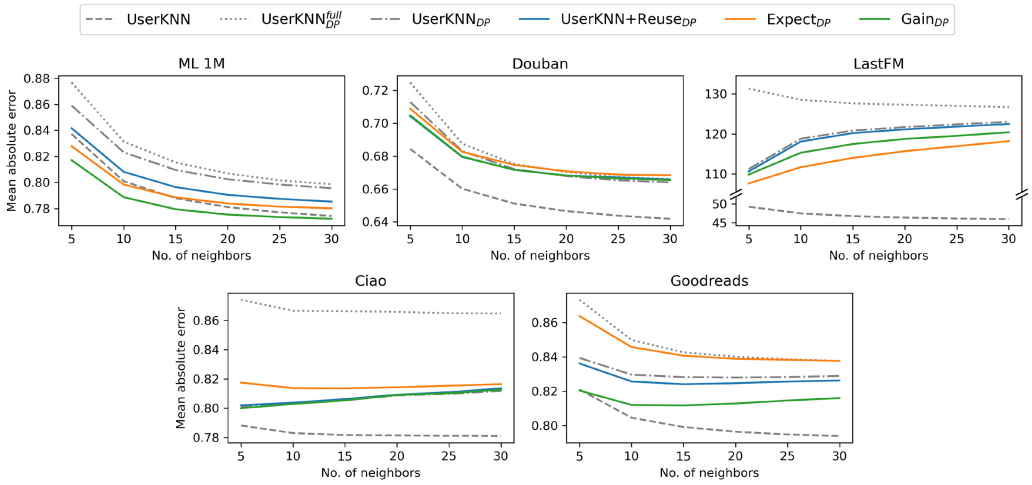

Fig. 6. Comparison of the recommendation accuracy between *ReuseKNN*$_{DP}$ and *UserKNN*$_{DP}$. We find that *ReuseKNN*$_{DP}$'s neighborhood reuse strategies, *Expect*$_{DP}$ and *Gain*$_{DP}$, can preserve or even improve recommendation accuracy in terms of lower MAE. This shows that reducing the number of users to which DP has to be applied can help to increase recommendation accuracy.

Furthermore, we incorporate *UserKNN* without DP and *UserKNN*$_{DP}^{full}$ as additional baselines for our experiments.

In general, for our neighborhood reuse strategies, DP does not cause an accuracy drop as severe as in case of *UserKNN*$_{DP}$ (see Figure 6). Plus, as expected, *UserKNN*$_{DP}^{full}$ performs worst due to the randomness that is added via DP to the rating data of all users. This shows that our neighborhood reuse concept helps to generate accurate recommendations in differentially private KNN-based recommender systems. For ML 1M and LastFM, a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) indicates that our neighborhood reuse strategies significantly increase recommendation accuracy over *UserKNN*$_{DP}$ for a model with $k = 10$ neighbors. Additionally, for ML 1M, *Gain*$_{DP}$ performs better than our non-DP baseline *UserKNN*.

Moreover, we observe that LastFM is highly sensitive to the incorporation of DP, since the mean absolute error magnitudes differ substantially between our non-DP experiment in Figure 5 and our DP experiment in Figure 6. In line with our previous results on non-DP *ReuseKNN*, *ReuseKNN*$_{DP}$'s unpersonalized neighborhood reuse strategy *Except*$_{DP}$ also cannot increase recommendation accuracy for Ciao and Goodreads, which are our two sparsest datasets. However, our personalized neighborhood reuse strategy *Gain*$_{DP}$ generates recommendations with significantly higher accuracy for Goodreads. For Ciao, no significant differences are found according to a two-tailed Mann-Whitney-U-Test ($\alpha = 0.01$). Thus, *Gain*$_{DP}$ can preserve recommendation accuracy.

For Douban, we observe no significant differences between our neighborhood reuse strategies and *UserKNN*$_{DP}$. We found empirically that for Douban, *UserKNN*$_{DP}$ and *ReuseKNN*$_{DP}$ utilize more rating data from vulnerable users than in the case of our remaining datasets. Thus, we measure the fraction of rating data; each user contributes to the dataset, i.e., $|R_u|/|R|$, where R are all users' ratings and R_u are user u's ratings. We find that for Douban, the 5% of users with the largest user profiles contribute substantially more ratings to the dataset than for our other datasets: 0.0008 (ML 1M), 0.0022 (Douban), 0.0012 (LastFM), 0.0009 (Ciao), and 0.0003 (Goodreads). This suggests that in the case of Douban, the recommendation process more often utilizes these users due to their abundance of rating data. This, however, makes these users more vulnerable. Therefore, we

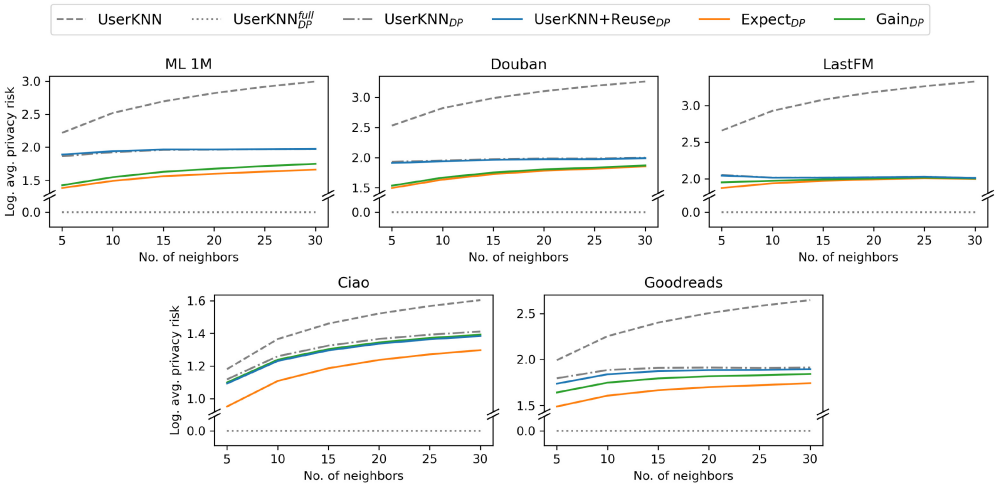

Fig. 7. Logarithm (base 10) of the privacy risk averaged over all users. *ReuseKNN*$_{DP}$'s neighborhood reuse strategies yield lower privacy risk than *UserKNN*$_{DP}$. This is due to the fact that *ReuseKNN*$_{DP}$ reduces the percentage of users with a privacy risk of τ (i.e., vulnerables) and simultaneously decreases the privacy risk of the remaining users (i.e., secures). Overall, we find that our unpersonalized neighborhood reuse strategy *Expect*$_{DP}$ achieves the best user privacy, i.e., the lowest privacy risk.

suppose that this strong utilization of DP-protected rating data from vulnerable users leads to no significant differences in accuracy between *UserKNN*$_{DP}$ and *ReuseKNN*$_{DP}$.

For Douban, we additionally compare *ReuseKNN*$_{DP}$ to *UserKNN*$^{full}_{DP}$. Our results suggest that our personalized reuse strategy *Gain*$_{DP}$ generates recommendations with significantly higher accuracy, wher *Except*$_{DP}$ show no significant differences. Thus, all our neighborhood reuse strategies can preserve recommendation accuracy for this dataset.

6.2.2 Privacy Risk. In *ReuseKNN*$_{DP}$, vulnerable users with high data usage are protected with DP and as such, their privacy risk is set to threshold τ. Moreover, secure users' privacy risk is also reduced since they are rarely exploited as neighbors in the recommendation process, i.e., low data usage (see Figure 1). Specifically, we compare our neighborhood reuse strategies to *UserKNN*$_{DP}$ and test for statistically significant differences. Furthermore, we use *UserKNN* without DP and *Full*$_{DP}$ as additional baselines.

We visualize the privacy risk of *ReuseKNN*$_{DP}$ and our three baselines *UserKNN*, *UserKNN*$_{DP}$, and *UserKNN*$^{full}_{DP}$ in Figure 7. We find that our neighborhood reuse strategies combined with DP can improve user privacy over *UserKNN*$_{DP}$. Specifically, a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) reveals that for our neighborhood reuse strategies on all datasets and for $k = 10$, users have significantly less privacy risk than in *UserKNN*$_{DP}$.

However, for LastFM, this privacy improvement is smaller than for the other datasets. Due to the large percentage of vulnerable users for all approaches (see Table 4), most users' privacy risk is set to τ due to the application of DP. Thus, the small percentage of secure users is insufficient to reduce the average privacy risk via neighborhood reuse in the case of LastFM.

Across all datasets, we observe that our unpersonalized neighborhood reuse strategy *Expect*$_{DP}$ yields the best (lowest) privacy risk. This finding is in line with our previous results in Table 4, which show that *Expect*$_{DP}$ performs best with respect to minimizing the percentage of vulnerable users. Thus, only a few users have a privacy risk of τ, and the high number of secure users enables

a drastic reduction of the average privacy risk. For example, the average privacy risk of secure users for a model with $k = 10$ neighbors for $Expect_{DP}$ is 11.45 for ML 1M, 18.34 for Douban, 49.92 for LastFM, 15.29 for Ciao, and 18.99 for Goodreads compared with the privacy risk of secure users for $UserKNN_{DP}$, which is 50.83 for ML 1M, 62.13 for Douban, 73.42 for LastFM, 21.76 for Ciao, and 41.13 for Goodreads. Additionally, a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) reveals that for ML 1M, Douban, Ciao, and Goodreads, these differences are significant. Thus, for secure users, $Expect_{DP}$ yields a substantially smaller privacy risk than $UserKNN_{DP}$.

6.2.3 Item Popularity Bias. We test for item popularity bias in $ReuseKNN_{DP}$'s recommendations via comparing $ReuseKNN_{DP}$ to our $UserKNN_{DP}$ baseline with respect to two metrics: Positivity-Popularity Correlation (PP-Corr) and Item Coverage (Coverage). Plus, we use *UserKNN* without DP and $UserKNN_{DP}^{full}$ as additional baselines. Moreover, in the case of PP-Corr, we test for statistically significant differences between our neighborhood reuse strategies and $UserKNN_{DP}$ (see Table 3). First and foremost, for ML 1M, Douban, LastFM, and Ciao, the non-DP baseline *UserKNN* yields lower PP-Corr values than all remaining methods that use DP. Similarly, applying DP to only vulnerable users yields lower PP-Corr values than applying DP to all users in the case of ML 1M, Douban, Ciao, and Goodreads. This fits well to related research [21] arguing that popularity bias can arise due to the recommender system's inability to personalize recommendations when DP is applied.

However, $ReuseKNN_{DP}$ can make the impact of DP on popularity bias less severe, since our neighborhood reuse strategies yield a lower PP-Corr than the DP baseline $UserKNN_{DP}$. No notable differences can be observed for Ciao only. We investigate this in more detail and find that the neighbors identified by $ReuseKNN_{DP}$ rated more distinct items than the neighbors identified by $UserKNN_{DP}$. As shown by related work on item popularity bias in recommender systems (e.g., [1, 39]), users with a larger user profile size tend to consume less popular items, which leads to less popularity bias. Due to the small number of ratings per user in Ciao (see Table 3), which is similar to a user cold-start setting [40], no noteworthy effects on popularity bias can be observed.

In addition to PP-Corr, we also evaluate Coverage, i.e., the percentage of items from the entire item catalog that occur within users' sets of top items. In general, $UserKNN_{DP}^{full}$ tends to give the highest item coverage and non-DP *UserKNN* yields the lowest item coverage. This makes sense since $UserKNN_{DP}^{full}$ protects all rating data with DP and, thus, the estimated rating scores are more random than for the remaining approaches. This leads to more randomized recommendations, and, therefore, to high item coverage [22]. These randomized recommendations also lead to the fact that, in Table 5, $ReuseKNN_{DP}$ cannot reach the item coverage of $UserKNN_{DP}^{full}$. However, more randomized recommendations lead to poorer accuracy than our previous results in Figure 6 show.

Our neighborhood reuse strategies cover fewer items than $UserKNN_{DP}$ only in the case of LastFM. We underline that these item coverage values are negatively correlated with our accuracy results in Figure 6. This indicates that for LastFM, there is a trade-off between recommendation accuracy and item coverage similar to the well-known trade-off between precision and recall [8].

6.2.4 Summary. Overall, our results are in line with the previously presented results for our non-DP *ReuseKNN*. Through neighborhood reuse and, thus, reducing the number of users that need to be protected with DP, recommendation accuracy can be preserved and, in many cases, even significantly improved over $UserKNN_{DP}$.

Also, our neighborhood reuse strategies used in $ReuseKNN_{DP}$ lead to significantly smaller privacy risk than $UserKNN_{DP}$. In particular, unpersonalized neighborhood reuse (i.e., $Except_{DP}$) performs best in increasing user privacy. This shows that the combination of neighborhood reuse and DP provides higher privacy than $UserKNN_{DP}$.

Table 5. PP-Corr and Item Coverage for a Model with $k = 10$ Neighbors

	ML 1M		Douban		LastFM		Ciao		Goodreads	
	PP-Corr	Coverage	PP-Corr	Coverage	PP-Corr	Coverage	PP-Corr	Coverage	PP-Corr	Coverage
UserKNN	**0.8405**	87.94%	**0.6780**	23.50%	**0.7339**	6.11%	**0.9755**	63.19%	0.9318	29.56%
UserKNN_{DP}	0.8742	88.77%	0.7589	26.55%	0.8625	15.54%	0.9758	64.03%	0.9409	31.59%
$\text{UserKNN}_{DP}^{full}$	0.8800	**89.53%**	0.7675	27.65%	0.8597	**15.86%**	0.9778	**66.72%**	0.9523	**34.13%**
$\text{UserKNN+Reuse}_{DP}$	0.8750	88.37%	0.7523	27.67%	0.8779	15.46%	0.9759	64.26%	0.9407	31.74%
Expect_{DP}	0.8688	88.83%	**0.7400	**28.75%**	0.8773	14.32%	0.9767	64.58%	****0.9317**	34.69%
Gain_{DP}	0.8725	88.07%	**0.7428	28.61%	0.8621	14.77%	0.9769	64.01%	0.9454	31.46%

Best results, i.e., highest for PP-Corr and lowest for Coverage, are in **bold**. For PP-Corr, a z-Test [32] shows, with ** ($\alpha = 0.01$) that our neighborhood reuse strategies as utilized in $\textit{ReuseKNN}_{DP}$ lead to estimated rating scores that are significantly less correlated with item popularity than in case of $\textit{UserKNN}_{DP}$. With respect to item coverage, especially $\textit{Expect}_{DP}$ can cover a larger percentage of the item catalog than $\textit{UserKNN}_{DP}$. Overall, our results suggest that $\textit{ReuseKNN}_{DP}$ does not increase item popularity bias over $\textit{UserKNN}_{DP}$.

Table 6. Mean Absolute Error and Average Privacy Risk Values for our Neighborhood Reuse Strategies Used in $\textit{ReuseKNN}_{DP}$, i.e., $\textit{Expect}_{DP}$ and $\textit{Gain}_{DP}$ and for the $\textit{UserKNN}_{DP}$ Baseline ($k = 10$)

	ML 1M		Douban		LastFM		Ciao		Goodreads	
	MAE	Privacy R.	MAE	Privacy R.	MAE	Privacy R.	MAE	Privacy R.	MAE	Privacy R.
UserKNN	0.80	330.77	0.66	665.17	47.46	844.94	0.78	35.21	0.80	182.26
UserKNN_{DP}	0.82	84.39	0.68	89.86	118.80	103.77	0.81	27.61	0.83	75.71
$\text{UserKNN}_{DP}^{full}$	0.83	0.00	0.69	0.00	128.41	0.00	0.87	0.00	0.85	0.00
$\text{UserKNN+Reuse}_{DP}$	0.81	87.16	0.68	87.16	118.13	103.56	0.81	26.54	0.83	68.35
Expect_{DP}	**0.80	**31.03	0.68	**43.25	**111.78	**86.81	0.82	**21.53	0.85	**40.95
Gain_{DP}	**0.79	**35.30	0.68	**46.57	**115.31	**93.95	0.81	**26.74	**0.81	**55.90

Also, we perform a one-tailed Mann-Whitney-U-Test ($\alpha = 0.01$) and mark (with **) significantly better (i.e., Lower) values than $\textit{UserKNN}_{DP}$. Overall, personalized neighborhood reuse (i.e., $\textit{Gain}_{DP}$) yields the best accuracy and unpersonalized neighborhood reuse (i.e., $\textit{Expect}_{DP}$) gives the lowest privacy risk. For Douban and LastFM, $\textit{Expect}_{DP}$ is well-suited as it yields the highest accuracy and lowest privacy risk. For the remaining datasets, all neighborhood reuse strategies provide a less serious accuracy-privacy trade-off than $\textit{UserKNN}_{DP}$.

In addition, we find that for $\textit{ReuseKNN}_{DP}$, high estimated rating scores are weaker correlated to item popularity than in the case of $\textit{UserKNN}_{DP}$ and that $\textit{ReuseKNN}_{DP}$ can estimate high rating scores for more items than $\textit{UserKNN}_{DP}$. Thus, $\textit{ReuseKNN}_{DP}$ does not increase item popularity bias.

6.3 Discussion

We provide a condensed summary of experimental results (see Table 6) for all evaluated approaches and all five datasets. Specifically, we present the accuracy (i.e., MAE@k) and average privacy risk (i.e., PrivacyRisk@k) values for a model with $k = 10$ neighbors.

Overall, non-DP *UserKNN* results in low MAE but high privacy risk values. This shows that approaches without DP sacrifice a user's privacy for recommendation accuracy. However, our neighborhood reuse strategies with DP provide a less serious trade-off between recommendation accuracy and privacy. Thus, in the following, we briefly discuss advantages and disadvantages of our neighborhood reuse strategies for all five datasets.

Across our neighborhood reuse strategies that are utilized in $\textit{ReuseKNN}_{DP}$, in general, personalized neighborhood reuse ($\textit{Gain}_{DP}$) provides the best recommendation accuracy. Plus, unpersonalized neighborhood reuse ($\textit{Expect}_{DP}$) yields the lowest privacy risk. For Douban and LastFM, $\textit{Expect}_{DP}$ performs best in both accuracy and privacy risk. Thus, in this case, $\textit{Expect}_{DP}$ is well suited to provide accurate and private recommendations. For ML 1M, Ciao, and Goodreads, no neighborhood reuse strategy provides the best result in both evaluation criteria. Thus, it depends on the recommender system service provider to decide what strategy could be utilized.

6.4 Additional Considerations and Experiments

While our experiments reported so far considered a rating prediction task as motivated by our problem statement in Section 3 (accordingly, we measured accuracy using the MAE [51]), we perform additional experiments with regards to a ranking-based recommendation scenario and a neural-based recommender system. Due to space limitations, the results of these are detailed in the appendices of this article. First, we model a ranking-based recommendation scenario, which is very common today. Accordingly, we perform experiments using a ranking-based evaluation metric, nDCG [35], and report results in Appendix B. Given the widespread adoption of deep learning techniques in the latest recommender systems, we also incorporate neighborhood reuse into a popular neural-based approach, neural collaborative filtering (NeuCF) [29]. The approach and results are detailed in Appendix C.

Overall, our additional experiments reveal the same pattern of results as discussed above. That is, the combination of neighborhood reuse and DP can provide a better trade-off between accuracy and privacy than recommendation methods without neighborhood reuse. This shows the generalizability of the neighborhood reuse principle for other evaluation scenarios and recommendation algorithms.

7 CONCLUSION

In this work, we investigate the efficacy of neighborhood reuse for differentially private KNN-based recommendations. We discuss the proposed approach in a two-stage evaluation procedure: (i) neighborhood reuse only, *ReuseKNN*, to distill the impact of neighborhood reuse on recommendation accuracy and on the percentage of users that need to be protected with differential privacy; and (ii) neighborhood reuse with differential privacy, $ReuseKNN_{DP}$, to investigate the practical benefit of neighborhood reuse for differentially private KNN-based recommendations. We find that *ReuseKNN* and $ReuseKNN_{DP}$ can substantially reduce the number of users that need to be protected with DP while outperforming related approaches in terms of accuracy. Also, we highlight that $ReuseKNN_{DP}$ effectively mitigates users' privacy risk, as most users are rarely exploited in the recommendation process. Our work illustrates how to address privacy risks in recommender systems through neighborhood reuse combined with DP.

Limitations. We recognize two limitations of the proposed approach. To quantify the privacy risk, we assume that all pieces of data are equally sensitive. In reality, disclosing a particular piece of information could pose a different level of privacy risk than disclosing another piece of information [38, 45]. Also, we focus on a neighborhood-based recommender system, specifically user-based *KNN*, instead of neural-based recommender systems. The latter are popular due to their ability to extract and exploit rich user and item representations for generating recommendations. However, traditional algorithms, such as user-based *KNN*, have been shown to perform well in a variety of real-world use cases [15]. Plus, neighborhood-based recommender systems have the advantage of providing justifiable recommendations and they incorporate new rating data of users efficiently without requiring a complete retraining of the whole model from scratch [16]. Nonetheless, we demonstrate in Appendix C that neighborhood reuse can be generalized to neural-based recommender systems, e.g., NeuCF [29].

Future Work. In this work, we evaluated the proposed approach using datasets of three different domains (movies, books, and music). Future work will consider additional, more sensitive domains, such as medicine, finance, insurance, and recruiting. We will also incorporate neighborhood reuse into other neural-based recommendation models, e.g., BERT4Rec [54]. Plus, we plan to study the impact of the proposed approach, i.e., neighborhood reuse and differential privacy, on individual users' preferences towards long-tail items, e.g., by using the dataset from our previous work on

fairness in music recommender systems [39]. Hence, our long-term plan is to investigate the interaction between privacy and fairness, two key aspects of trustworthy recommender systems.

MATERIALS

The Python-based implementation of our work is publicly available.[1] Also, we provide the source code for generating our sample of the Goodreads dataset. All remaining datasets are publicly available as well (see Section 5.3).

APPENDICES

A DETAILED DIFFERENTIAL PRIVACY ANALYSIS

Our differential privacy analysis relies on the fact that, even if the adversary is able to infer the rating used in the recommendation process, it is unaware whether this rating is the neighbor's real rating or was randomly generated by our m_{DP} mechanism. Formally:

$$\frac{Pr[\text{Adversary's assumption: Real rating} \mid \text{Truth: Real rating}]}{Pr[\text{Adversary's assumption: Real rating} \mid \text{Truth: Random rating}]} = \tag{20}$$

$$\frac{Pr[\text{Non-DP rating}] + Pr[\text{Real rating} \mid \text{DP rating}] \cdot Pr[\text{DP rating}]}{Pr[\text{Random rating} \mid \text{DP rating}] \cdot Pr[\text{DP rating}]} = \tag{21}$$

$$\frac{Pr[\text{Non-DP rating}]}{Pr[\text{Random rating} \mid \text{DP rating}] \cdot Pr[\text{DP rating}]} + \underbrace{\frac{Pr[\text{Real rating} \mid \text{DP rating}]}{Pr[\text{Random rating} \mid \text{DP rating}]}}_{m_{DP}\text{ mechanism}} = \tag{22}$$

$$\frac{1}{0.25} \cdot \frac{Pr[\text{Non-DP rating}]}{Pr[\text{DP rating}]} + \frac{0.75}{0.25} = \tag{23}$$

$$4 \cdot \frac{\frac{\text{PrivacyRisk@}k(u)}{\text{DataUsage@}k(u)}}{\frac{\text{DataUsage@}k(u) - \text{PrivacyRisk@}k(u)}{\text{DataUsage@}k(u)}} + 3 = \tag{24}$$

$$4 \cdot \frac{\text{PrivacyRisk@}k(u)}{\text{DataUsage@}k(u) - \text{PrivacyRisk@}k(u)} + 3 \leq e^{\epsilon} \tag{25}$$

which leads to a privacy parameter of

$$\epsilon = \ln\left(3 + 4 \cdot \frac{\text{PrivacyRisk@}k(u)}{\text{DataUsage@}k(u) - \text{PrivacyRisk@}k(u)}\right). \tag{26}$$

In the case of $UserKNN_{DP}^{full}$, all ratings of a user u are protected with DP and, therefore, PrivacyRisk@$k(u) = 0$, which leads to $\epsilon = \ln 3$. In the case of *UserKNN*, no DP is applied at all and, thus, computing ϵ is not possible since ϵ is part of the DP framework. Therefore, we set $\epsilon = \infty$. In the case of $UserKNN_{DP}$ and $ReuseKNN_{DP}$, DP is applied to the rating data of users, for which the usage of their data exceeds threshold τ. Assuming that u is vulnerable, then DataUsage@$k(u) > \tau$ and PrivacyRisk@$k(u) = \min[\tau, \text{DataUsage@}k(u)]$. Therefore, it follows that $0 <$ PrivacyRisk@$k(u) <$ DataUsage@$k(u)$. Varying PrivacyRisk@$k(u)$ within these boundaries yields:

$$\ln 3 < \ln\left(3 + 4 \cdot \frac{1}{\text{DataUsage@}k(u) - 1}\right) \leq \epsilon \leq \ln\left(3 + 4 \cdot (\text{DataUsage@}k(u) - 1)\right) < \infty. \tag{27}$$

[1] https://github.com/pmuellner/ReuseKNN

This shows that $UserKNN_{DP}$ and $ReuseKNN_{DP}$ provide better privacy than *UserKNN*, but worse privacy than $UserKNN_{DP}^{full}$.

Moreover, via neighborhood reuse, $ReuseKNN_{DP}$ utilizes a vulnerable user u more often as neighbor (with DP-protected data) than $UserKNN_{DP}$ does. Also, note that the privacy risk of u is the same for $ReuseKNN_{DP}$ and $UserKNN_{DP}$. From these observations and Equation (26), we see that the ϵ value for $ReuseKNN_{DP}$ is smaller than the ϵ value for $UserKNN_{DP}$. Thus, for vulnerable users, our neighborhood reuse principle leads to $ReuseKNN_{DP}$ providing better privacy than $UserKNN_{DP}$.

B EVALUATION OF TOP-N RECOMMENDATIONS

In our article, we show that $ReuseKNN_{DP}$ can achieve better accuracy in terms of the rating prediction metric MAE than a traditional *KNN* recommender system with DP. In the following, we evaluate $ReuseKNN_{DP}$ in a top-n items recommendation setting via the ranking-aware metric *nDCG* (Normalized Discounted Cumulative Gain) [35].

B.1 Evaluation Process

To generate a list of recommended items that can be evaluated via *nDCG*, we select the $n = 10$ items with the highest predicted rating score for a given target user u [51]. Formally, for a recommender system model $\mathcal{R}$ and k neighbors, a user u's top-n items are given by:

$$\mathcal{R}_{top}^{k}(u) = \arg\max_{i \in Q_u}^{n} \mathcal{R}^{k}(u, i) \tag{28}$$

where Q_u are the items in u's query set. We consider items in the test set as relevant if their true rating exceeds the average rating in the training set of the given dataset.

B.2 Experiments

Our results reveal that $Expect_{DP}$ and $Gain_{DP}$ can yield higher *nDCG* scores than $UserKNN_{DP}^{full}$ (see Figure 8). In the case of the ML 1M dataset, $Expect_{DP}$ and $Gain_{DP}$ can even outperform the non-DP baseline *UserKNN*. Especially $Gain_{DP}$ yields high *nDCG* scores. Overall, this experiment validates

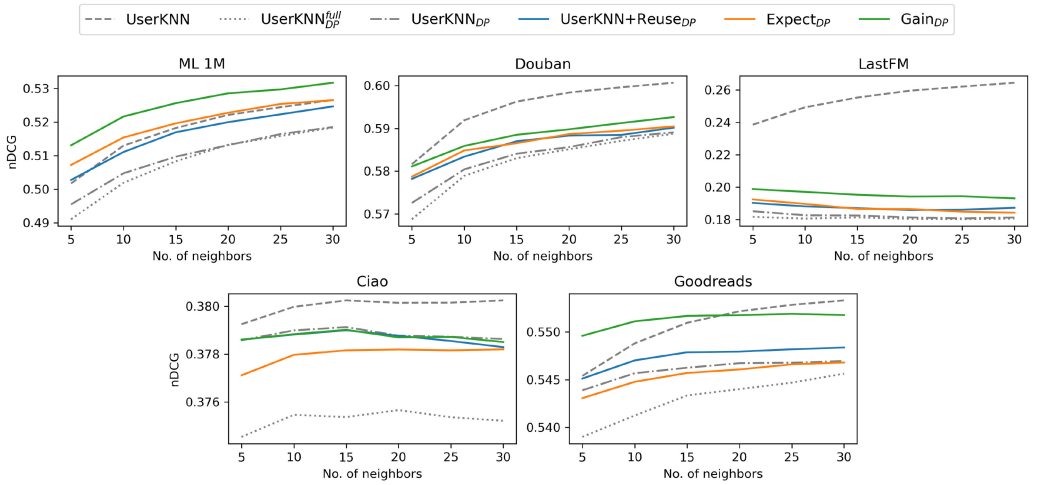

Fig. 8. nDCG values of each user's top 10 items. The pattern matches our results reported in Section 6, i.e., $ReuseKNN_{DP}$ can yield better accuracy than $UserKNN_{DP}$. Also, especially personalized neighborhood reuse (i.e., $Gain_{DP}$) can preserve accuracy well.

the results of our rating prediction evaluation setting also in a top-n items recommendation setting.

C EVALUATION OF NEURAL-BASED RECOMMENDATIONS

This work considers rating data as input to the recommender system. However, recommender systems can also use more complex representations of users and items, i.e., embeddings as generated by neural network architectures. Therefore, in the following, we demonstrate the generalizability of our approach for neural-based recommendation methods.

C.1 Generation of Embeddings

To generate user and item embeddings, we rely on a simple approach inspired by the NeuCF [29] architecture. Specifically, for user u and item i, the predicted rating score $y_{u,i}$ is given by:

$$y_{u,i} = b + ReLu\left(wx_u W_u^T x_i W_i\right), \tag{29}$$

where x_u is the id of user u, x_i is the id of item i, the size of the embedding layer is $d = 16$, $W_u, W_i \in \mathbb{R}^d$, $w, b \in \mathbb{R}$, and $ReLu$ is the activation function. We apply Adam [37] with a step size of $\alpha = 0.001$ to minimize the MAE between $y_{u,i}$ and the rating $r_{u,i}$. The parameters α and d are set to the values proposed in [29]. We train the network for 50 epochs and use a batch size of 128. We stop training if there is no improvement of the training objective for more than 10 epochs. After training, the user and item embeddings are given by $x_u W_u$ and $x_i W_i$ respectively.

C.2 Neural-Based Recommendations

For our neural-based variants of *UserKNN—NeuKNN* and $NeuKNN_{DP}$—we calculate the similarity between the target user and the candidate neighbors based on their user embeddings (see Equation (2)). For $NeuKNN{+}Reuse_{DP}$, i.e., an embedding-based variant of $ReuseKNN_{DP}$, we also use an embedding-based similarity. Plus, we employ a modified definition of $reusability$ that measures the reusability of a candidate neighbor c based on the previous $t-1$ rating queries of target user u:

$$reusability(c|u,i,t) = \sum_{j \in Q_u^{(t-1)}} \mathbb{1}_{N_{u,j}}(c) \cdot sim(i,j), \tag{30}$$

where $\mathbb{1}_{N_{u,j}}(c)$ is the indicator function of candidate neighbor c being in $N_{u,j}$. The item similarity sim is the cosine similarity between i's and j's item embeddings. Therefore, $reusability(c|u,i,t)$ is the summed-up item similarity between the target item i and all items $j \in Q_u^{(t-1)}$ (i.e., the previous $t-1$ rating queries of u) for which c has been used as neighbor.

C.3 Experiments

In our experiments, we perform evaluation according to the following procedure: First, we randomly split the dataset into 5 equally sized subsets: $D_{1 \le i \le 5}$. We select D_1 and equally partition it into the validation data that is used for validating the user and item embeddings and the test data that is used for evaluating the recommendations. The remaining data, $\bigcup_{2 \le i \le 5} D_i$, is used to train the user and item embeddings and to generate recommendations. Next, we select D_i and repeat this procedure for all $D_{2 \le i \le 5}$. Eventually, we compute the mean of our evaluation results.

Accuracy. For all datasets, $NeuKNN{+}Reuse_{DP}$ outperforms our baseline $NeuKNN_{DP}^{full}$ that applies DP to all users (see Figure 9). For completeness, we also visualize *NeuKNN* that does not apply DP at all and, thus, yields higher accuracy than both DP-based methods. Overall, the result for our embedding-based methods $NeuKNN_{DP}^{full}$ and $NeuKNN{+}Reuse_{DP}$ are in line with the results of

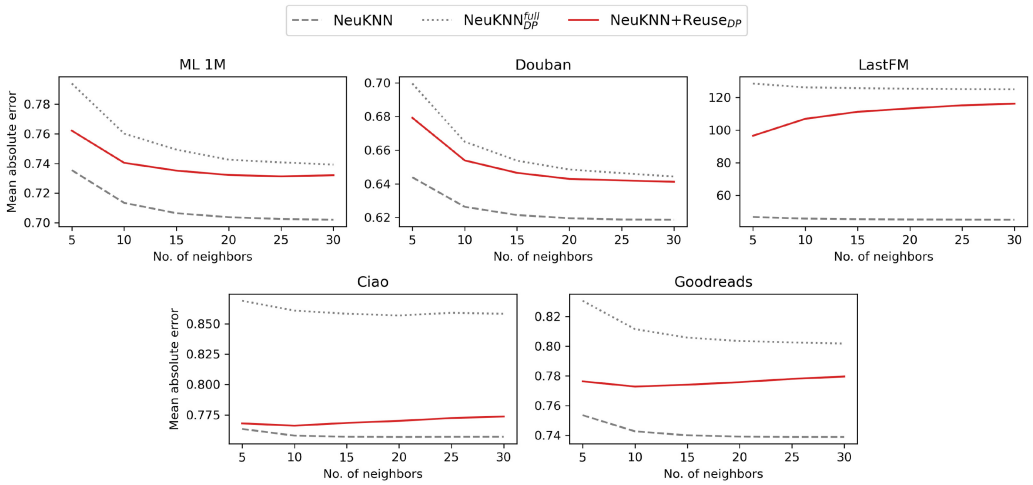

Fig. 9. Mean absolute error of our neural-based *KNN* recommender system variants. Our results indicate that combining neighborhood reuse with DP (i.e., *NeuKNN+Reuse*$_{DP}$) yields better accuracy (lower MAE) than neural-based methods that apply DP without neighborhood reuse (i.e., *NeuKNN*$^{full}_{DP}$).

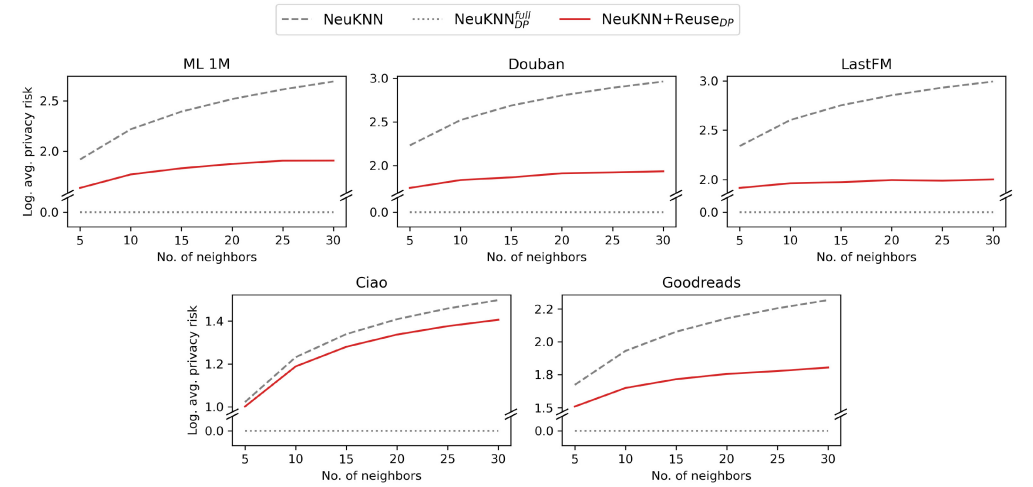

Fig. 10. Logarithmic (base 10) average privacy risk of our neural-based *KNN* recommender system variants. Via combining neighborhood reuse and DP, *NeuKNN+Reuse*$_{DP}$ decreases the users' average privacy risk compared with neural-based methods that do not apply DP (i.e., *NeuKNN*).

our rating-based methods, i.e., that the combination of neighborhood reuse and DP yields better accuracy on all five investigated datasets than traditional DP-based methods.

Privacy. Our baseline *NeuKNN* without DP yields the worst privacy risk, whereas *NeuKNN*$^{full}_{DP}$ yields a privacy risk of zero since all users are protected with DP (see Figure 10). *NeuKNN+Reuse*$_{DP}$ protects only vulnerable users with DP; in this way, its privacy risk lies between our two baselines. Therefore, also in terms of privacy risk, the results of our embedding-based experiments match the pattern of the results of our rating-based methods.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and the associate editor for their valuable remarks and suggestions.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. In *Proc. of the RMSE'19 Workshop, in Conjunction with ACM RecSys'19*.

[2] Gediminas Adomavicius and Jingjing Zhang. 2012. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems* 3, 1 (2012), 1–17.

[3] Sushant Agarwal. 2020. *Trade-offs between fairness, interpretability, and privacy in machine learning*. Master's thesis. University of Waterloo.

[4] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Antonio Ferrara, and Fedelucio Narducci. 2021. How to put users in control of their data in federated top-N recommendation with learning to rank. In *Proc. of SAC'21*.

[5] Ghazaleh Beigi and Huan Liu. 2020. A survey on privacy in social media: identification, mitigation, and applications. *ACM Transactions on Data Science* 1, 1 (2020), 1–38.

[6] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*. Springer, 37–40.

[7] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2012. The impact of data obfuscation on the accuracy of collaborative filtering. *Expert Systems with Applications* 39, 5 (2012), 5033–5042.

[8] Michael Buckland and Fredric Gey. 1994. The relationship between recall and precision. *Journal of the American Society for Information Science* 45, 1 (1994), 12–19.

[9] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. 2011. "You might also like:" Privacy risks of collaborative filtering. In *Proc. of S&P'11*. IEEE, 231–246.

[10] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2022. Efficient federated matrix factorization against inference attacks. *ACM Transactions on Intelligent Systems and Technology* 13, 4, Article 59 (Jun 2022), 20 pages.

[11] Chaochao Chen, Huiwen Wu, Jiajie Su, Lingjuan Lyu, Xiaolin Zheng, and Li Wang. 2022. Differential private knowledge transfer for privacy-preserving cross-domain recommendation. In *Proc. of ACM WWW'22*.

[12] Chaochao Chen, Jun Zhou, Bingzhe Wu, Wenjing Fang, Li Wang, Yuan Qi, and Xiaolin Zheng. 2020. Practical privacy preserving POI recommendation. *ACM Transactions on Intelligent Systems and Technology* 11, 5, Article 52 (Jul 2020), 20 pages.

[13] Xiaolin Chen, Xuemeng Song, Ruiyang Ren, Lei Zhu, Zhiyong Cheng, and Liqiang Nie. 2020. Fine-grained privacy detection with graph-regularized hierarchical attentive representation learning. *ACM Transactions on Information Systems* 38, 4 (2020), 1–26.

[14] Ziqian Chen, Fei Sun, Yifan Tang, Haokun Chen, Jinyang Gao, and Bolin Ding. 2022. Studying the impact of data disclosure mechanism in recommender systems via simulation. *ACM Transactions on Information Systems* (2022).

[15] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems* 39, 2, Article 20 (Jan 2021), 49 pages.

[16] Christian Desrosiers and George Karypis. 2010. A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook* (2010), 107–144.

[17] Josep Domingo-Ferrer. 2010. Rational privacy disclosure in social networks. In *Proc. of MDAI'10*. Springer, 255–265.

[18] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proc. of TAMC'08*. Springer, 1–19.

[19] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proc. of ITCS'12*. 214–226.

[20] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[21] Michael D. Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. 2018. Privacy for all: Ensuring fair and equitable privacy protections. In *Proc. of FAccT'18*. PMLR, 35–47.

[22] Jill Freyne and Shlomo Berkovsky. 2013. Evaluating recommender systems for supportive technologies. In *User Modeling and Adaptation for Daily Routines*. Springer, 195–217.

[23] Arik Friedman, Bart P. Knijnenburg, Kris Vanhecke, Luc Martens, and Shlomo Berkovsky. 2015. Privacy aspects of recommender systems. In *Recommender Systems Handbook*. Springer, 649–688.

[24] Chen Gao, Chao Huang, Dongsheng Lin, Depeng Jin, and Yong Li. 2020. DPLCF: Differentially private local collaborative filtering. In *Proc. of SIGIR'20*. 961–970.

[25] Craig Gentry et al. 2009. *A Fully Homomorphic Encryption Scheme*. Stanford University, Stanford, CA.

[26] Guibing Guo, Jie Zhang, Daniel Thalmann, and Neil Yorke-Smith. 2014. ETAF: An extended trust antecedents framework for trust prediction. In *Proc. of ASONAM'14*.
[27] Jialiang Han, Yun Ma, Qiaozhu Mei, and Xuanzhe Liu. 2021. DeepRec: On-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In *Proc. of WWW'21*. 900–911.
[28] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 1–19.
[29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proc. of WWW'17*. 173–182.
[30] Jonathan L. Herlocker, Joseph A. Konstan, A. l. Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proc. of SIGIR'99*. 230–237.
[31] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (Jan 2004), 5–53.
[32] Dennis E. Hinkle, William Wiersma, and Stephen G. Jurs. 2003. *Applied Statistics for the Behavioral Sciences*. Vol. 663. Houghton Mifflin College Division.
[33] T. Ryan Hoens, Marina Blanton, Aaron Steele, and Nitesh V. Chawla. 2013. Reliable medical recommendation systems with patient privacy. *ACM Transactions on Intelligent Systems and Technology* 4, 4, Article 67 (Oct 2013), 31 pages.
[34] Longke Hu, Aixin Sun, and Yong Liu. 2014. Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In *Proc. of SIGIR'14*.
[35] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.
[36] Arjan J. P. Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L. Lagendijk, and Qiang Tang. 2013. Privacy in recommender systems. In *Social Media Retrieval*. Springer, 263–281.
[37] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR'15*.
[38] Bart P. Knijnenburg and Alfred Kobsa. 2013. Making decisions about privacy: Information disclosure in context-aware recommender systems. *ACM Transactions on Interactive Intelligent Systems* 3, 3 (2013), 1–23.
[39] Dominik Kowald, Markus Schedl, and Elisabeth Lex. 2020. The unfairness of popularity bias in music recommendation: A reproducibility study. In *Proc. of ECIR'20*.
[40] Emanuel Lacic, Dominik Kowald, Matthias Traub, Granit Luzhnica, Jörg Peter Simon, and Elisabeth Lex. 2015. Tackling cold-start users in recommender systems with indoor positioning systems. In *Proc. of ACM RecSys'15*. ACM.
[41] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta matrix factorization for federated rating predictions. In *Proc. of SIGIR'20*.
[42] Kun Liu and Evimaria Terzi. 2010. A framework for computing the privacy scores of users in online social networks. *ACM Transactions on Knowledge Discovery from Data* 5, 1 (2010), 1–30.
[43] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. In *Proc. of CIKM'20*.
[44] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. of AISTATS'17*. PMLR, 1273–1282.
[45] A. K. M. Nuhil Mehdy, Michael D. Ekstrand, Bart P. Knijnenburg, and Hoda Mehrpouyan. 2021. Privacy as a planned behavior: Effects of situational factors on privacy perceptions and plans. In *Proc. of UMAP'21*.
[46] Peter Muellner, Dominik Kowald, and Elisabeth Lex. 2021. Robustness of meta matrix factorization against strict privacy constraints. In *Proc. of ECIR'21*.
[47] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proc. of S&P'19*. IEEE.
[48] Vasileios Perifanis and Pavlos S. Efraimidis. 2022. Federated neural collaborative filtering. *Knowledge-Based Systems* 242 (2022), 108441.
[49] Naren Ramakrishnan, Benjamin J. Keller, Batul J. Mirza, Ananth Y. Grama, and George Karypis. 2001. When being weak is brave: Privacy in recommender systems. *IEEE Internet Computing* (2001), 54–62.
[50] Hanchi Ren, Jingjing Deng, and Xianghua Xie. 2022. GRNN: Generative regression neural network–a data leakage attack for federated learning. *ACM Transactions on Intelligent Systems and Technology* 13, 4, Article 65 (May 2022), 24 pages.
[51] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proc. of ACM RecSys'14*. 129–136.
[52] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: Learning local collective embeddings. In *Proc. of ACM RecSys'14*. 89–96.
[53] Agrima Srivastava and G. Geethakumari. 2013. Measuring privacy leaks in online social networks. In *Proc. of ICACCI'13*. IEEE, 2095–2100.

[54] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from Transformer. In *Proc. of CIKM'19*. 1441–1450.
[55] Qiang Tang and Jun Wang. 2016. Privacy-preserving friendship-based recommender systems. *IEEE Transactions on Dependable and Secure Computing* 15, 5 (2016), 784–796.
[56] Isabel Wagner and David Eckhoff. 2018. Technical privacy metrics: A systematic survey. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 1–38.
[57] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proc. of ACM RecSys'18*. 86–94.
[58] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proc. of ACL'19*. Association for Computational Linguistics, 2605–2610.
[59] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.
[60] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. FedCTR: Federated native Ad CTR prediction with cross-platform user behavior data. *ACM TIST* 13, 4, Article 62 (Jun 2022), 19 pages.
[61] Yu Xin and Tommi Jaakkola. 2014. Controlling privacy in recommender systems. In *Proc. of NIPS'14*. 2618–2626.
[62] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proc. of ICML'13*. PMLR, 325–333.
[63] Mingwu Zhang, Yu Chen, and Jingqiang Lin. 2021. A privacy-preserving optimization of neighbourhood-based recommendation for medical-aided diagnosis and treatment. *IEEE Internet of Things Journal* (2021).
[64] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership inference attacks against recommender systems. In *Proc. of ACM SIGSAC'21*. 864–879.
[65] Tianqing Zhu, Gang Li, Yongli Ren, Wanlei Zhou, and Ping Xiong. 2013. Differential privacy for neighborhood-based collaborative filtering. In *Proc. of IEEE/ACM ASONAM'13*. 752–759.

Received 17 August 2022; revised 26 May 2023; accepted 6 June 2023