The Journal of Supercomputing (2024) 80:19433–19461
https://doi.org/10.1007/s11227-024-06221-7

A novel target item-based similarity function in privacy-preserving collaborative filtering

Emre Yalcin[1] · Alper Bilge[2]

Accepted: 10 May 2024 / Published online: 27 May 2024

Abstract

Memory-based collaborative filtering schemes are among the most effective recommendation technologies in terms of prediction quality, despite commonly facing issues related to accuracy, scalability, and privacy. A prominent approach suggests an intuitively reasonable modification to the similarity function, which has been proven to provide more accurate recommendations than those generated by state-of-the-art memory-based collaborative filtering methods. However, this scheme exacerbates the scalability problem due to additional computational costs and fails to protect individual privacy. In this study, we recommend using a preprocessing method to eliminate relatively dissimilar items from the prediction estimation process, thereby enhancing the scalability of the proposed approach. We explore how to provide recommendations based on the previously proposed similarity function while preserving privacy and propose privacy-preserving schemes to accomplish this task. Additionally, we apply our preprocessing approach to our proposed privacy-preserving schemes to improve both scalability and accuracy. After analyzing our schemes with respect to privacy and additional costs, we conduct experiments with real data to examine the impact of our schemes on scalability and accuracy. The empirical outcomes indicate that our preprocessing scheme significantly alleviates scalability issues in both conventional and privacy-preserving environments and enhances accuracy within privacy-preserving frameworks.

Keywords Privacy · Accuracy · Scalability · Collaborative filtering · Preprocessing · Prediction

✉ Alper Bilge
abilge@akdeniz.edu.tr

Emre Yalcin
eyalcin@cumhuriyet.edu.tr

[1] Computer Engineering Department, Sivas Cumhuriyet University, 58140 Sivas, Turkey

[2] Computer Engineering Department, Akdeniz University, 07058 Antalya, Turkey

1 Introduction

The seamless integration of Internet services into our daily lives has revolutionized how we consume entertainment, shop, and interact with the world. This transformation has led to a significant shift from traditional means of entertainment and shopping to digital platforms. People now prefer watching movies and listening to music on renowned streaming services, and online shopping has become a more convenient alternative to brick-and-mortar stores. This shift is largely due to the convenience, variety, and personalized experiences that these digital services offer. However, the proliferation of these services has also introduced a new challenge: information overload, or as it is increasingly known, *infobesity* [1]. In response to this challenge, recommender systems have become indispensable. These sophisticated systems sift through vast amounts of data to provide tailored recommendations, guiding customers to products and services that align with their unique preferences and interests.

Among the various technologies powering recommender systems, Collaborative Filtering (CF) stands out for its effectiveness and widespread adoption. CF distinguishes itself by leveraging collective user behavior to filter and rank products, drawing on the “wisdom of crowds” and the “law of large numbers” [2]. These principles enable CF systems to predict what users might like in the future based on past preferences, a feature that has proven beneficial across various domains. From e-commerce giants like Amazon.com to streaming platforms like YouTube, Spotify, and Booking.com, CF has been instrumental in enhancing user engagement and driving sales [3–5]. The success of CF is not just anecdotal; a wealth of research corroborates its effectiveness in delivering personalized content and recommendations [6, 7].

However, the data-driven nature of CF systems raises significant privacy concerns. The collection and analysis of personal preferences necessary for CF to function can lead to invasive user profiling, threatening individual privacy [8]. As public awareness of these privacy risks has increased, a tension has emerged between the desire for personalized experiences and the imperative to protect personal information. This has led to a growing interest in Privacy-Preserving Collaborative Filtering (PPCF), which aims to maintain the benefits of CF while safeguarding user privacy [9, 10].

PPCF systems, despite their considerable appeal, confront significant challenges related to efficiency and scalability. These systems are required to function in real-time, and the continual growth of user bases and product catalogs places immense pressure on their scalability capabilities [11]. Scalability issues extend beyond mere technical obstacles; they crucially affect the efficiency, accuracy, and viability of the recommendations provided [12]. With the user and item numbers on the rise, CF systems increasingly struggle to process extensive datasets efficiently, resulting in delays and potentially diminished recommendation quality. Such growth not only poses a challenge to system performance but also heightens privacy risks, as larger datasets increase the likelihood of privacy infringements unless carefully managed. Moreover, the intricate task of managing vast user-item matrices calls for advanced computational resources, presenting a substantial barrier for smaller entities aiming to adopt CF technologies. Addressing these scalability challenges is thus essential for enhancing CF system performance and ensuring their ability to meet real-world demands while safeguarding user privacy.

In addition to scalability concerns, the integration of privacy measures introduces added complexity and necessitates greater resource allocation, which could impair the system's responsiveness and the accuracy of its predictions. Therefore, maintaining the timeliness and relevance of predictions becomes a delicate balancing act. Any deviation from accuracy can lead to user dissatisfaction and, in certain instances, notable financial losses.

The quest for improved predictive accuracy has led to innovations in similarity functions, such as the one proposed by [13], which refines the way items are compared and ranked within CF systems [14, 15]. By adjusting the weighting of item similarities, this function aims to enhance the relevance of recommendations. However, this approach increases the computational burden, exacerbating existing scalability challenges without addressing the underlying privacy concerns. In this study, we examine the method proposed by [13] and introduce a preprocessing approach to speed up the prediction production process, which filters out items not strongly correlated to the target item. We also implement privacy-preserving measures on both the original and enhanced schemes to assess the effects of the new similarity function and preprocessing on the scalability and accuracy of both non-private and privacy-preserving frameworks. The major contributions of our work can be summarized as follows:

- We propose a novel preprocessing method aimed at eliminating relatively dissimilar items from the prediction process using the similarity function, thereby addressing scalability challenges. This approach significantly enhances the system's efficiency without compromising on prediction accuracy.
- In our investigation of the formerly proposed similarity function within a privacy-preserving environment, we integrate our preprocessing scheme to further optimize both scalability and accuracy. This dual application demonstrates the method's adaptability and effectiveness in maintaining high performance under privacy constraints, underscoring our contribution toward achieving a balance between computational efficiency and data privacy in predictive modeling.
- We perform different sets of experiments using real data sets to evaluate our proposed schemes with respect to scalability and accuracy.

The rest of the paper is organized, as follows. Related research is summarized in Sect. 2 and preliminaries are given in Sect. 3. We deeply analyze applicability of similarity function in privacy-preserving environment and propose preprocessing method to enhance scalability in Sect. 4. After analyzing the proposed schemes by means of overhead costs and privacy in Sect. 5, we present real data-based experimental evaluations and their results in Sect. 6. The gained insights from the experimental studies and the limitations regarding the study are given in Sect. 7. We finally conclude the paper and give future research directions in Sect. 8.

2 Related work

Recommender systems primarily utilize content-based filtering or CF techniques [6, 16]. Additionally, there are hybrid approaches that combine content-based and collaborative methods to enhance accuracy [17]. CF systems can be categorized into memory-based or model-based approaches. Memory-based systems rely on the entire collection to calculate similarities using metrics such as Pearson's correlation coefficient, cosine similarity, or distance-based similarity [13]. For practical deployments covering a large number of products, item-based similarity calculations are preferred, whereas user-based similarities are employed for better accuracy in other instances [18]. Model-based techniques employ various data processing methods like clustering [19], dimensionality reduction [20], and classifiers [11, 21] to create a model of the collection. This model aids in making faster prediction estimates. However, these methods typically involve many parameters that require tuning and periodic updates. Hybrid applications also exist, performing different parts of the prediction process to achieve improved performance [22, 23].

Although today's recommender systems are very popular and widely used, they face with various challenges. Firstly, there is a need to make them more robust to rapidly growing nature of data. Thus, researchers have studied scalability problem in recommender systems and proposed solutions adopting data reduction techniques. For example, a principal component analysis through standard singular value decomposition-based and hierarchical nonlinear methods is applied to improve scalability of CF systems [24]. Also, by employing Singular Value Decomposition for dimensionality reduction and ontology for enhancing recommendation accuracy, a recent study effectively improves the trade-off between accuracy and computational efficiency [25]. Additionally, a recent method leverages dimensionality reduction and clustering techniques, specifically employing the *k*-means algorithm and Singular Value Decomposition, to cluster similar users and reduce dimensionality [26]. It proposes a two-stage recommender system designed to generate accurate and efficient recommendations, aiming to enhance the performance of existing algorithms within the bustling ecosystem of e-commerce and Internet-based companies. Another work proposes a novel collaborative filtering method, CBE-CF, which integrates information entropy and bi-clustering to address the challenges of data sparsity and computational efficiency in traditional recommendation algorithms [27]. By clustering both rows and columns of the user-item-rating matrix, CBE-CF identifies dense rating modules, using information entropy to quantify similarity between new users and these modules, thereby optimizing predictions through a blend of item-based global generalization and local module similarity. Also, a novel bio-inspired clustering ensemble is introduced in [28], which combines swarm intelligence and fuzzy clustering models, to enhance user-based collaborative filtering in recommender systems, addressing the issue of information overload in the digital age. Another method incorporates a time decay function for preprocessing user ratings, utilizes project attribute and user interest vectors for characterization, and employs clustering and enhanced similarity measures to identify users' nearest neighbors and recommend project candidates [29].

However, these approaches are computationally inefficient, which limits their benefits. Furthermore, they tend to be ineffective with extremely large data sets. Secondly, another challenge of recommender systems is to produce accurate predictions. Inaccurate referrals may get customers angry and reduce sales of online vendors. The most successful implementations are memory-based solutions in which whole database is utilized supporting wisdom of crowds principle. A prominent study [13] propose a new similarity function to improve quality of predictions by ranking item votes according to their similarity to the target item. Thirdly, CF systems are expected to preserve confidentiality as online amenities become more popular [10, 30]. This motivates providing accurate predictions without violating individual privacy.

Initial solutions were first proposed by [31, 32] relying on a distributed architecture to form an aggregate data via cryptographic techniques. However, today's recommender systems mostly operate on a central server. Therefore, rather than concealing preference data, researchers proposed users to submit obfuscated vectors. Cryptographic methods are often employed in such systems to safeguard individual privacy [33, 34]. For instance, a recent study developed an unsynchronized secure multi-party computation protocol that allows for the incremental computation of item similarities without the need for individuals to participate in the computation process [35]. Similarly, a distributed recommendation system is introduced by [36] where encrypted information is exchanged between vendors and a mediator using secure protocols, enabling the generation of recommendations or item rankings without compromising security. Additionally, recent research has explored the use of homomorphic encryption to securely encrypt Quality of Service metrics, facilitating the recommendation of personalized, high-quality web services that respect user and location privacy [37]. A recent study also introduced an effective strategy for group recommender systems that protects user privacy during the recommendation process [34]. Also the computation of item similarities is treated as a probabilistic inference problem and presented a semi-distributed belief propagation network approach for item-based PPCF systems, which safeguards user preferences by only involving a select group of individuals in the network at any given time [38].

Anonymization stands out as another essential technique for ensuring user privacy by severing the identifiable link between users and their rating data. For instance, a recent study advocates for a PPCF approach that employs microaggregation to create k-anonymity masks, thereby safeguarding user privacy [39]. Another work delves into achieving k-anonymity for databases that are both large-scale and sparse, such as recommender systems, proposing a clustering-based k-anonymity heuristic method for privacy protection in data exchange [40]. Another work introduces a (p, l, a)-diversification strategy to enhance the efficacy of traditional k-anonymity techniques, where p represents prior knowledge of users' rating profiles and (l, a) symbolizes user diversity to augment privacy levels [41]. Finally, as an exemplar of anonymization in practice, a prominent study implements this approach within the context of privacy-conscious smart cities, framing urban privacy concerns as a PPCF challenge [42].

Differential privacy stands as a key effective strategy utilized in CF-based methods to evaluate the potential privacy loss incurred when incorporating a user's personal preferences into generating recommendations. The seminal work introduced the concept of differential privacy [43], which was later applied in the realm of recommender systems, where it is employed randomization of actual user preferences via a private covariance matrix [44]. A Subsequent research developed a distance-based differential privacy approach, modifying real user profiles by swapping elements at certain distances within the profile [45]. Additionally, an alternative technique focused on noise calibration to balance privacy and utility, enabling the fine-tuning of noise levels injected into users' preference profiles [46]. In a more specialized application [47], a dual-phase strategy is introduced for ensuring differential privacy in neighborhood-based recommendations, particularly within the medical recommendation sphere, through the secure selection of private neighbors.

In CF systems, strategies such as data obfuscation and randomized perturbation are prominently employed for privacy preservation, as extensively discussed in the literature. Specifically, the technique of data obfuscation seeks to shield sensitive details while permitting access to necessary data for recommendation generation. For example, [48] detail a permutation-based data obfuscation method designed for use in CF applications that depend on a central server. Furthermore, [49] describe a user-based PPCF recommender that involves semi-honest intermediaries to carry out supplementary computations for enhancing privacy. For scenarios involving distributed recommendations, [50] advocate for the use of obfuscated user profiles with coarse granularity as a means to protect personal information. Moreover, [51] improve upon traditional single-level obfuscation methods by employing variable obfuscation degrees to more effectively pinpoint target users.

Randomized Perturbation Techniques (RPTs) offer an alternative method for preserving user privacy by altering actual user preferences before their transmission to the recommendation server, as discussed by [30, 52, 53]. For instance, [54] introduce a scalable PPCF approach that employs bisecting k-means clustering along with randomized perturbations. In a separate study by [55], the combination of RPTs with secure multi-party computation methods is used to secure the privacy of user profiles distributed across various databases during the recommendation process. Polatidis et al. [56] propose the use of varying levels and spectrums of random values to improve the process of RPT-based PPCF. Additionally, [57] present a hybrid model that merges RPTs with differential privacy, providing a more comprehensive protection of user preferences compared to existing RPT methods. Lastly, [58] demonstrate the application of RPTs in safeguarding privacy within the multi-criteria recommender systems sphere.

The literature review highlights various privacy protection methods for CF recommenders, including anonymization, homomorphic encryption, differential privacy, and RPTs. In this study, we have chosen to implement RPTs to safeguard user privacy for several reasons outlined below. Each privacy protection method is suited to particular types of recommendation scenarios and comes with its own set of limitations. For instance, methods other than RPTs, notably homomorphic encryption and secure multi-party computation mechanisms, are typically tailored for distributed systems. Given our focus on a centralized recommendation framework, these approaches are not apt for our purposes. Additionally, methods based on cryptography tend to demand more computational resources due to the complexity of their encryption processes [37]. Conversely, while differential privacy offers a commendable level of privacy, it often compromises the accuracy of recommendations by introducing noise that can mask crucial patterns needed for precise recommendation generation [46].

With the primary goal of enhancing system efficiency in privacy without sacrificing predictive accuracy, these alternative methods do not align with our objectives. Moreover, although various perturbation schemes exist to protect individual privacy in CF systems, opportunities remain to mitigate accuracy degradation resulting from perturbation. Thus, our research aims to improve the scalability of the CF framework proposed by [13], incorporating privacy safeguards through randomization techniques, all the while ensuring minimal impact on the accuracy of the recommendations.

3 Preliminaries

3.1 Collaborative filtering and similarity function

CF systems collect ratings from users and form a user-item matrix $U_{n\times m}$, which consists of n users' votes about m items. A typical CF prediction estimation process includes three steps: (i) An active user (a) sends her available ratings and requests a prediction on a target item (q). (ii) Her neighbors are determined by calculating similarities between a and all other users. (iii) A weighted average of neighbors' ratings on q, referred to as p_{aq}, is estimated and returned to a.

According to results presented in [13], the best similarity measure to calculate weight between users a and u is Pearson's correlation coefficient (PCC) given in Eq. (1).

$$\mathrm{PCC}_{au} = \frac{\sum_{i=1}^{m'}[(v_{ai} - \overline{v_a})(v_{ui} - \overline{v_u})]}{\sqrt{\sum_{i=1}^{m'}(v_{ai} - \overline{v_a})^2}\sqrt{\sum_{i=1}^{m'}(v_{ui} - \overline{v_u})^2}}, \tag{1}$$

where v_{ai} and v_{ui} are the votes for item i by users a and u, respectively. Similarly, $\overline{v_a}$ and $\overline{v_u}$ are the average votes of users a and u, respectively and m' is the number of co-rated items by both a and u. After calculating similarities, k of the users are marked as neighbors according to calculated similarity values. Finally, p_{aq} can then be estimated using Eq. (2) [59].

$$p_{aq} = \overline{v_a} + \frac{\sum_{u=1}^{k}[(v_{uq} - \overline{v_u}) \times w_{au}]}{\sum_{u=1}^{k}|w_{au}|} \tag{2}$$

in which w_{au} is the similarity weight between a and u.

As seen from Eq. (1), each co-rated item has equal effect on similarity measure. [13] propose to modify the similarity metric so that each co-rated item's effect is also ranked with the item similarity between corresponding item and the target item.

In other words, if an item is very similar to the target item, then it will have a superior influence on estimated prediction. Therefore, they propose to apply adjustments on several similarity metrics. We present the best performing user-user similarity measure in Eq. (3) [13].

$$\mathrm{PCC}_{au}^{q} = \frac{\sum_{i=1}^{m'}[IS_{iq}^{2} \times (v_{ai} - \overline{v_a}) \times (v_{ui} - \overline{v_u})]}{\sqrt{\sum_{i=1}^{m'}[IS_{iq} \times (v_{ai} - \overline{v_a})]^2}\sqrt{\sum_{i=1}^{m'}[IS_{iq} \times (v_{ui} - \overline{v_u})]^2}}, \tag{3}$$

where PCC_{au}^{q} denotes PCC between users a and u for q and IS_{iq} denotes the item similarity between co-rated items i and q. The best measure to calculate IS_{iq} might vary for different data sets; however, PCC (Eq. 1) and Cosine similarity (Eq. 4) are shown to be the most feasible ones [13].

$$\mathrm{Cosine}_{iq} = \frac{\sum_{u=1}^{n'}(v_{ui} \times v_{uq})}{\sqrt{\sum_{u=1}^{n'} v_{ui}^{2}}\sqrt{\sum_{u=1}^{n'} v_{uq}^{2}}}. \tag{4}$$

3.2 Preserving individual privacy by randomized perturbation

Widespread usage of the Internet services also leads misusage of personal data. People are getting more conscious about privacy risks like profiling, price discrimination, unsolicited marketing, and so on. Therefore, they refuse to submit authentic preferences in order to avoid privacy violations. However, accurate predictions can only be produced through qualified collections. PPCF schemes aim to dissipate worries of users' by collecting disguised ratings instead of pure votes and produce dependable predictions upon such perturbed preferences. Due to distortion on input data, accuracy losses are inevitable. Therefore, such systems must be well-tuned to keep such losses insignificant.

According to [30], privacy in the context of CF systems has two key features; keeping (i) true ratings and (ii) the exact list of rated products private. In order to preserve such information, randomized perturbation techniques (RPTs) are useful tools to adjust required privacy levels and letting the server to calculate qualified predictions without jeopardizing privacy of the individual much. Data disguising protocol employing RPTs is described in Procedure 1 as applied in [30].

Procedure 1 Data Disguising Protocol

```
Require: User vector ($u_{1\times m}$), $\sigma_{max}$, $\beta_{max}$
    Estimate z-scores ($\rightarrow Z$):
    $\overline{u} \leftarrow$ MEAN($u$); $\sigma_u \leftarrow$ STD($u$)
    for all items in $u$ ($j \leftarrow 1$ to $m$) do
        $z_j = (u_j - \overline{u})/\sigma_u$
    end for
    Determine privacy parameters:
    $\beta \leftarrow$ RND($0, \beta_{max}$); $\sigma \leftarrow$ RND($0, \sigma_{max}$); $\alpha \leftarrow \sqrt{3}\sigma$
    $e \leftarrow$ # of empty cells; $g \leftarrow$ # of genuine ratings
    $F \leftarrow e \times \beta\%$                          ▷ # of empty cells to be filled
    Select distribution & generate random numbers:
    $dist \leftarrow$ RANDOM(uniform, Gaussian)
    $R \leftarrow dist(g + F; \mu = 0, \sigma|\alpha)$
    Disguise z-scores ($\rightarrow Z'$):
    for all items in $u$ ($j \leftarrow 1$ to $m$) do
        $z_j^{'} = (z_j + R_j)$
    end for
    return $Z'$
```

According to Procedure 1, a genuine vote value v is disguised by adding a random number r onto it, where r is drawn from either uniform or Gaussian distribution with zero mean (μ) and a standard deviation (σ). Note that $\sigma_{\max}$ is one of the controlling parameters for provided confidentiality level because the higher the $\sigma_{\max}$ is, the more the distortion on input data users have. Additionally, users forge some fake ratings into their profiles to conceal truly rated items. The number of unrated items in profiles to be forged can be associated with $\beta_{\max}$ parameter, which is another controlling parameter for privacy. After the central server sets $\sigma_{\max}$ and $\beta_{\max}$ values, users choose individual σ and β values from the permitted interval and disguise each of their votes according to Procedure 1. The central server collects such perturbed values and forms disguised user-item matrix, $U'_{n\times m}$, and operates on this matrix to produce recommendations.

4 A more precise and scalable PPCF scheme

In this section, we describe our schemes, which are proposed to enhance scalability of CF and PPCF schemes based on formerly proposed similarity function. We first define our preprocessing scheme to eliminate dissimilar items from prediction process to alleviate scalability issues and possibly enhance accuracy of referrals in non-private scheme. Then, we introduce modifications to apply the similarity function on masked data to improve recommendation quality. Finally, we discuss applying proposed preprocessing scheme onto the proposed privacy-enhanced environment to further improve scalability and accuracy of privacy-preserving scheme.

4.1 A preprocessing scheme to eliminate irrelevant items

The similarity function proposed by [13] utilizes item similarities between a commonly rated item and the target item in user similarity function. Since user-similarity calculation becomes more complicated due to new function, such computational overhead also affects online performance of the recommender system. Moreover, memory-based CF applications already suffer from scalability issue due to constantly growing size of data. Inspired from the idea of ranking preferences according to item similarity, we propose an additional preprocessing step to eliminate relatively dissimilar items from prediction estimation process. Such preprocessing scheme is aimed to enhance scalability of CF system because it focuses on reducing online response time significantly by eliminating irrelevant items to each corresponding target item.

According to the scheme proposed in [13], similarities between items are calculated using either Eq. (1) or Eq. (4) in off-line time can be used in the online prediction estimation process, as outlined in Procedure 2. Item similarities rank each rating's effect on the estimation of prediction. This way co-rated items resembling more similarity to the target item have a superior effect on the estimation. However, all items still join to the process no matter they have a dissimilar manner to the target item. Therefore, we propose to eliminate such items from the prediction estimation process so that the process speeds up due to the reduction of dimensions in original user-item matrix.

Procedure 2 Off-line Item Similarity Calculation

```
Require: User-item matrix ($U_{n \times m}$)
Initialize: $IS_{m \times m} \leftarrow 0$                                    ▷ item similarities matrix
Calculate and sort item similarities:
for all $item_i$ in $U$ ($i \leftarrow 1$ to $m$) do
    for all $item_j$ in $U$ ($j \leftarrow i$ to $m$) do
        $IS(i, j) = \text{SIMILARITY}(U(item_i), U(item_j))$    ▷ either using Eq. 1 or Eq. 4
    end for
end for
$[IS\_values, IS\_index] = \text{SORT}(IS, \text{descending})$        ▷ to be used in online process
return $IS\_values$ and $IS\_index$
```

According to the CF recommendation estimation process explained in Sect. 3.1, the bottleneck in the process is the calculation of similarities between *a* and all users in the system. Therefore, we base our proposed preprocessing scheme to handle such bottleneck. If relatively dissimilar items are removed from the matrix for that particular target item, then a significant reduction can be obtained in dimensions of the original user-item matrix. Such dissimilarity can be determined relying on a predetermined similarity threshold value (τ), so that the items having smaller similarity than the threshold value are eliminated. Then, a temporary user-item matrix can be formed for each corresponding target item, which is to be used in neighborhood formation process. Since similarity calculations are performed in the compact and reduced form of original user-item matrix, it will take much less time to calculate user similarities online. Pseudo code of the prediction estimation process relying on our preprocessing scheme is given in Procedure 3.

Procedure 3 Online Prediction Estimation via Preprocessing

```
Require: User-item matrix ($U_{n \times m}$), active user ($a$), neighbor count ($k$),
    target item ($q$), threshold value ($\tau$)
    Decide items to eliminate:
    $remaining\_items \leftarrow IS\_index(q, IS\_values(q) > \tau)$   ▷ locate items more similar than $\tau$
    Reduce dimensions of matrices:
    $U \leftarrow U(remaining\_items); a \leftarrow a(remaining\_items); IS \leftarrow IS(remaining\_items)$
    Initialize: $US_{1 \times n} \leftarrow 0$   ▷ user similarities vector
    Calculate and sort user similarities:
    for all $user_i$ in $U$ ($i \leftarrow 1$ to $n$) do
        $US(i, j) =$ SIMILARITY($U(user_i), q$)   ▷ using Eq. 3
    end for
    [$sim\_val, neighbor\_idx$] = SORT($US$,descending)   ▷ to be used in prediction estimation
    Estimate prediction:
    $p_{aq} \leftarrow$ PREDICTION($U, sim\_val, neighbor\_idx, k$)   ▷ estimated using Eq. 2
    return $p_{aq}$
```

4.2 A PPCF scheme using target item-based similarity function

Due to privacy concerns, people prefer to submit their disguised vectors instead of explicit expressions. Therefore, the central server needs to estimate predictions based on such disguised collections with decent accuracy. We explain how formerly proposed target item-based similarity function can be applied onto private prediction generation algorithm in the following.

4.2.1 Neighborhood formation

The PCC equation (Eq. 1) can be represented as the covariance of two z-score transformed user vectors [59]. PPCF schemes typically employ such similarity calculation method due to perturbation scheme. To employ target item-based similarity function onto such similarity calculation method, we propose to utilize item similarities as a factor to covariance calculation, as shown in Eq. (5).

$$w_{au}^{q} = \frac{\sum_{i=1}^{m} \mathrm{IS}_{iq} \times z_{ai} \times z_{ui}}{m}, \tag{5}$$

where w_{au}^{q} denotes covariance-based PCC weight between a and u for q, z_{ai} and z_{ui} represent z-score transformations of users' ratings on item i, respectively. However, as explained in Sect. 3.2, users submit their disguised z-scores, Z', due to privacy concerns. Hence, similarities between users in an RPT-based PPCF scheme are estimated on masked data, as in Eq. (6).

$$\begin{aligned} w_{au}^{q'} &= \frac{\mathrm{IS}(q)\cdot Z'_a \cdot Z'_u}{m} = \frac{\sum_{i=1}^{m} IS_{iq}\times z'_{ai}\times z'_{ui}}{m} = \frac{\sum_{i=1}^{m} IS_{iq}(z_{ai}+r_{ai})(z_{ui}+r_{ui})}{m} \\ &= \frac{\sum_{i=1}^{m}\mathrm{IS}_{iq}z_{ai}z_{ui} + \sum_{i=1}^{m}\mathrm{IS}_{iq}z_{ai}r_{ui} + \sum_{i=1}^{m}\mathrm{IS}_{iq}z_{ui}r_{ai} + \sum_{i=1}^{m}\mathrm{IS}_{iq}r_{ai}r_{ui}}{m} \\ &\approx \frac{\sum_{i=1}^{m}\mathrm{IS}_{iq}\times z_{ai}\times z_{ui}}{m}. \end{aligned} \tag{6}$$

Notice that R_a and R_u vectors are noise data drawn from a zero-mean distribution, which are generated to disguise original z-score values. Similarly, the expected means of z-scores are zero, as well. Thus, the expected value of the last three summations in Eq. (6) converges to zero, which helps the server estimate similarities with decent accuracy relying on perturbed aggregate data.

Off-line item similarity calculations are also performed on perturbed data. Without privacy concerns, it is trivial to calculate such similarities using PCC or cosine similarity. However, since users send their disguised z-scores, the data collector should be able to estimate weights between items from masked data, as well. Due to disguising mechanism, the server can utilize covariance-based PCC similarly for item similarities, as shown in Eq. (6). The server can estimate similarities between co-rated item i and target item q, as explained in the following: In Eq. (4), the nominator part performs multiplication between co-rated users, which can be treated as a dot product in privacy-preserving scheme because unrated items have a zero rating value. With increasing number of users in the system, as can be followed from Eq. (5), such dot product calculations can be performed with sufficient accuracy due to zero-mean nature of random number distribution. The denominator holds magnitude calculation of two vectors. The server can estimate such magnitudes for an item vector, as shown in Eq. (7).

$$||Z'||_2 = ||(Z+R)||_2 = \sqrt{\sum_{u\in T}(z_u + r_u)^2}, \tag{7}$$

where T is the set of users who rated corresponding item and R represents the distribution of such users' random perturbing factors added onto genuine ratings. Equation (7) can be rewritten without square roots, as follows:

$$\sum_{u\in T}(z_u + r_u)^2 = \sum_{u\in T} z_u^2 + 2\sum_{u\in T} z_u r_u + \sum_{u\in T} r_u^2 \approx \sum_{u\in T} z_u^2 + \sum_{u\in T} r_u^2. \tag{8}$$

Equation (8) holds as number of users submitting a vote for the item increases due to generated random numbers distribution with zero mean. However, to get rid of the second summation, the server can subtract its contribution relying on the maximum allowed standard deviation of the random numbers, as follows:

$$\sum_{u\in T}(z_u + r_u)^2 \approx \sum_{u\in T} z_u^2 + \sum_{u\in T} r_u^2 - T\sigma_{max}^2 \approx \sum_{u\in T} z_u^2. \tag{9}$$

After computing the summation, the server can take the square root and estimate magnitudes of vectors and similarity weights between items based on masked data. Then, the most similar k of such users are labeled as neighbors to be used in prediction production process.

4.2.2 Prediction estimation

The server estimates a prediction based on masked data and replies back to a with such estimation. Each active user de-normalizes received prediction via her ratings mean and standard deviation. Since predictions are generated relying on masked z-scores data, Eq. (2) can be rewritten for producing a private prediction for a on q, as follows [60]:

$$p'_{aq} = \overline{v_a} + \sigma_a \times \frac{\sum_{u=1}^{k} z'_{uq} w_{au}^{q'}}{\sum_{u=1}^{k} |w_{au}^{q'}|} = \overline{v_a} + \sigma_a \times P'_{aq}, \tag{10}$$

where k is the number of neighbors utilized in the prediction production process, $\overline{v_a}$ and σ_a represent a's mean vote and standard deviation, respectively. Therefore, the server estimates P'_{aq} and sends it back to a, where she de-normalizes provided aggregation and obtains the final prediction. The server can estimate P'_{aq} based on masked data, as follows:

$$\begin{aligned} P'_{aq} &= \frac{\sum_{u=1}^{k}(w_{au}^{q} + V_{au}^{q})(z_{uq}r_{uq})}{\sum_{u=1}^{k} |w_{au}^{q} + V_{au}^{q}|} \\ &= \frac{\sum_{u=1}^{k} w_{au}^{q} z_{uq} + \sum_{u=1}^{k} w_{au}^{q} r_{uq} + \sum_{u=1}^{k} V_{au}^{q} z_{uq} + \sum_{u=1}^{k} V_{au}^{q} r_{uq}}{|\sum_{u=1}^{k} w_{au}^{q} + \sum_{u=1}^{k} V_{au}^{q}|} \\ &\approx \frac{\sum_{u=1}^{k} w_{au}^{q} z_{uq}}{\sum_{u=1}^{k} |w_{au}^{q}|}. \end{aligned} \tag{11}$$

Equation (11) holds because expected values of the last three summations in nominator and the second one in denominator converge to zero due to zero-mean random number distributions. In other words, the server can estimate P'_{aq} on masked data and still can produce accurate predictions.

4.3 Improving PPCF referrals via preprocessing

We finally propose to employ the preprocessing idea proposed in Sect. 4.1 onto PPCF framework described in Sect. 4.2, which utilizes the target item-based similarity function. In this proposed framework, PPCF referrals are aimed to be produced in less amount of time to enhance scalability. In addition, it is possibly expected to obtain more qualified private referrals. This way, accuracy-enhanced traditional CF scheme via target item-based similarity function will be further improved to provide private referrals with better accuracy and in scalable time. To do so, the private framework defined in Sect. 4.2 utilizes Procedures 2 and 3 along with privacy-preserving similarity calculation (Eqs. 5 and 9) and prediction estimation (Eqs. 10 and 11) equations.

5 Performance and privacy analysis

It is imperative to analyze our preprocessing scheme, which is employed in both non-private and privacy-preserving CF schemes with respect to off-line and online costs. In addition, a detailed privacy analysis is provided in order to evaluate the privacy preservation procedure to investigate how and to what extent it is effective. During such analysis and experimental examination, we denote traditional recommendation approach as **CF**, similarity function-enhanced CF method as **CF+** proposed by [13], and preprocessing applied ultimate model as **CF++**.

5.1 Overhead costs analysis

Overhead costs due to introduced preprocessing scheme must be analyzed by means of three cost-related phases, i.e., (i) communication, (ii) storage, (iii) and computational phases. An overview of the analysis is presented in Table 1.

Compared to the traditional **CF** approach, **CF+** approach and our item reduction preprocessing-based **CF++** approach scheme does not cause any extra communication overheads. All three schemes require a transfer of 1-by-m user vector, which introduces an $O(1)$ complexity in terms of communications costs. Hence, we can conclude that both the number of communications and amount of data to be transmitted in online and off-line phases remain the same for all three schemes.

CF scheme requires a storage cost in the order of $O(nm)$ to record preferences of n users on m items. However, **CF+** and **CF++** schemes utilize item similarities in user-user similarity computations. Therefore, **CF+** requires a total storage area in the order of $O(nm) + O(m^2)$ to also hold item similarities. However, **CF++**

Table 1 Overview of overhead costs

	Communication	Storage costs	Computational costs	
			Off-line	Online
CF	$O(1)$	$O(nm)$	–	$O(k + nmP)$
CF+	$O(1)$	$O(nm) + O(m^2)$	$O(nm^2)$	$O(k + nmP')$
CF++	$O(1)$	$O(nm) + O(2m^2)$	$O(nm^2) + O(m \log m)$	$O(k + n\overline{m}P')$

P and P': number of calculations in Eq. (1) and Eq. (3), respectively

$\overline{m}$: reduced number of items for corresponding target item approach also eliminates some items from the collection relying on item similarities, which requires to hold sorted item similarity index values in addition to item similarities that results in a storage cost of $O(nm) + O(2m^2)$ in total.

Computation costs should be analyzed separately for off-line and online phases. Although off-line computations are not critical for recommender systems, it is better to provide a report on off-line work overload. Traditional **CF** scheme solely runs online and does not perform any off-line computations. However, **CF+** and **CF++** schemes calculate item-item similarities in off-line phase in the order of $O(nm^2)$, where **CF++** scheme also sorts such similarities, which requires an additional $O(m \log m)$ time using quick sort algorithm.

The important component of recommender systems' performance is determined by how fast queries are responded online. **CF** scheme runs in $O(k + nmP)$ time, where k represents the number of neighbors to be utilized and P is the complexity of computations performed in user-user similarity calculation via a similarity measure. **CF+** scheme also produces predictions in a similar manner; however, drawback of **CF+** scheme is that it further complicates online similarity calculation step by assembling item-similarity factors into similarity formulas. Such increased computational complexity is denoted with $O(k + nmP')$ in Table 1, where P' represents the increased complexity of calculations and $P' > P$ all the time. Our proposed preprocessing, on the other hand, reduces the number of items to be utilized in similarity calculation step and requires an online computation time in the order of $O(k + n\overline{m}P')$, where the size of $\overline{m}$ is determined for each corresponding target item separately, but making sure that $\overline{m} \ll m$ to relieve scalability issues. Data disguising procedure allows PPCF systems to collect and store preference data similar to the non-private schemes and produce predictions in an identical way, as well. Therefore, such overhead costs analysis is also valid for privacy-preserving conjugates of **CF**, **CF+**, and **CF++**.

5.2 Privacy analysis

Data disguising protocol focuses on preventing the central server to deduce about (i) if a rating is genuine or forged and (ii) actual values of the genuine ratings. Accordingly, these two considerations are analyzed to evaluate the privacy level provided by the system.

User profiles contain fake ratings about $\beta\%$ of empty cells, where such value is chosen uniformly randomly from the interval $(0, \beta_{\max}]$. Let us denote the probability of guessing β over $(0, \beta_{\max}]$ with $\Pr(\beta)$. Uncertainty caused by $\Pr(\beta)$ can be measured using Shannon entropy [61] of masked vector. Recall that the entropy of a random variable $X = \{x_1, x_2, \dots, x_n\}$ distributed by a probability mass function p is defined as $H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$. We can model users of a PPCF system as random variables associated with normal distribution $\mathcal{N}(x;0,\sigma^2)$ or uniform distribution $\mathcal{U}(x;2\sigma\sqrt{3})$. Let g and e represent the numbers of genuine votes and empty cells of a user, respectively. Correspondingly, let $P = \{p_1, p_2, \dots, p_g\}$ and $R = \{r_1, r_2, \dots, r_{e\times\beta}\}$ define the probability distribution of genuine ratings and forged items, respectively.

Hereafter, a user's distribution can be modeled as $\mathcal{S} = (\#P + \#R)/(g + (e \times \beta_{\max}))$ and provided privacy can be quantified by $H(\mathcal{S})$. An example scenario is presented in Fig. 1, where the user is assumed to have 50 genuine ratings on 500, 1000, and 2000 ratable items and $\beta_{\max}$ is varied from 5 to 100%.

We also present the unlikelihood of locating genuine votes besides their values with a probabilistic approach. The server does not know the numbers of actual ratings (g) and empty cells (e) in a user profile. Instead, it has g' and e' due to the Procedure 1. However, g can be calculated as $g = m - e$, where $e = e'/(1 - \beta\%)$. Thus, combining the possibilities of (i) choosing β over $(0, \beta_{\max}]$ and (ii) determining the exact list of g out of g' disguised values, we can conclude that discriminating genuine ratings out of a disguised user vector is $\Pr(\beta) \times \genfrac{}{}{0pt}{}{g'}{g}$, where $\genfrac{}{}{0pt}{}{g'}{g}$ stands for the number of combinations of g' objects chosen g at a time.

5.2.1 Privacy obtained by individual perturbations of elements

Even if the central server distinguishes genuine votes from forged entries, it still needs to extract real values from their masked z-score forms. Additionally, the privacy obtained by adding random noise on ratings must also be quantified. Agrawal and Aggarwal [62] propose a differential entropy-based metric to quantify privacy of an additive noise-based perturbed variable, where such metric is utilized in PPCF context by [30, 60, 63]. Let random variables P and R represent the original user vector and perturbing random data, respectively yielding $D = P + R$. Then *average conditional privacy* of P is defined as $\Pi(P|D) = 2^{H(P|D)}$, where $2^{H(P|D)}$ represents *conditional differential entropy* of P given D. Recall that P and R are independent random variables. Thus, privacy level of P after disclosing D is given by $\Pi(P|D) = \Pi(P) \times (1 - Pr(P|D))$, where $\Pr(P|D) = 1 - 2^{H(D|P) - H(D)}$. Assuming that

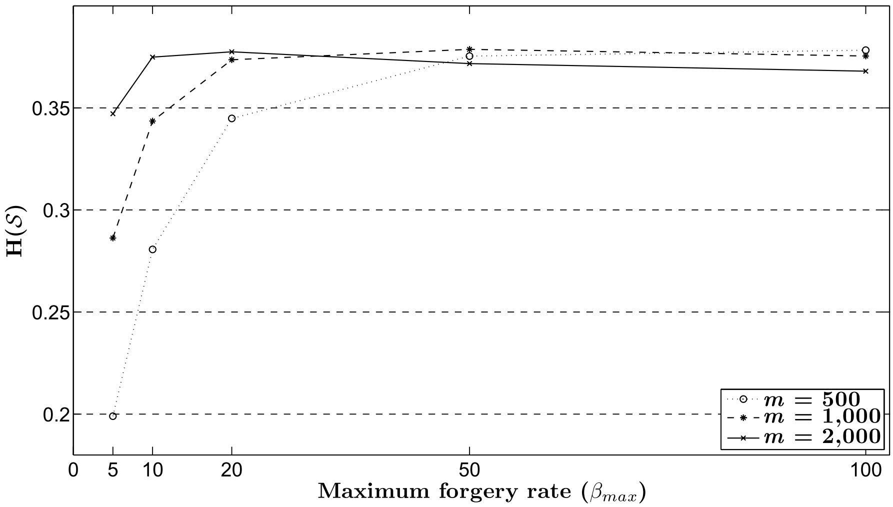

Fig. 1 Privacy levels for varying $\beta_{\max}$ values

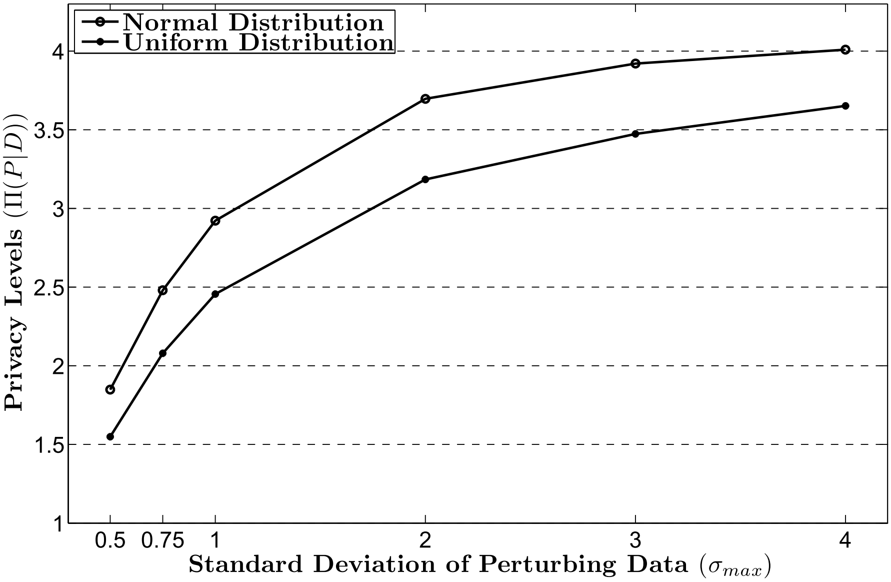

Fig. 2 Privacy levels for varying $\sigma_{\max}$ values

P distributes normally, privacy levels, $\Pi(P|D)$, for various perturbation levels are presented in Fig. 2. Recall that the distribution of R is determined by coin tosses. As seen from Fig. 2, provided privacy levels enhance with increasing level of perturbation as expected and Gaussian distribution provides slightly better privacy.

Finally, as Eq. (10) demonstrates, the server needs to de-normalize extracted z-score values, which requires deducing mean and standard deviation of each user's original rating profiles.

6 Experiments

We performed several experiments on two benchmark data sets to scrutinize the effects of applying the similarity function in privacy-preserving systems and employing our proposed preprocessing scheme on non-private and privacy-preserving schemes.

6.1 Data sets and evaluation criteria

Experiments were performed on two well-known benchmark data sets. MovieLens data set (ML) was collected by GroupLens at the University of Minnesota (http://www.grouplens.org) and Netflix provided a training data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. We used a subset of Netflix data set (NF), where we sampled 10,000 users from differing density ranges. Data sets are suitable

Table 2 Descriptions of data sets

Name	User × Item	Rating scale	Total votes	Density (%)
ML	6040 × 3952	5-star	1 M	4.25
NF	10,000 × 17,700	5-star	2,337,295	1.32

to show effects of preprocessing schemes as they both are extremely sparse and large. Detailed information about the data sets is given in Table 2.

Like [13] did, we utilized mean absolute error (MAE) to measure quality of predictions. MAE basically measures how precise the predictions are compared to the actual ratings as an average of absolute errors, i.e., $\sum_N (e_i/n) = \sum_N (|p_i - r_i|/N)$, where p_i is the estimated prediction, r_i is the actual rating value, and N is the number of produced predictions. Thus, the smaller the MAE is, the better the results are. Since the proposed preprocessing method aims to improve scalability, total elapsed time (T) in seconds spent on producing online recommendations is also recorded.

6.2 Experimentation methodology

Experiments were realized on uniformly randomly chosen train and test sets. The original data set (U or U') was divided into two, where uniformly randomly chosen 30% of all users are assigned to be test users and remaining ones as train users. After training and test sets were constructed, five rated items' actual votes were withheld for each test (active) user. Such entries were replaced with null, their values were tried to be predicted, and estimations were compared with actual values. User-user similarities were computed via PCC on both data sets and item-item similarities are calculated by PCC in ML and by Cosine similarity in NF as shown by [13] to the best performing measures on those data sets. We also set k to 10. Trials were performed in MATLAB 8.0 environment using a computer with an Intel Core i7 2.8 GHz dual-core processor and 4 GB RAM.

6.3 Results and discussion

We utilized four sets of experiments. Firstly, we evaluated how the similarity values between users differ when random error is introduced into similarity calculations. Secondly, we experimented on **CF++** scheme by employing proposed preprocessing method to see its effects on accuracy and scalability compared to **CF** and **CF+** schemes. Thirdly, we derived **PPCF+** scheme by implementing the similarity function onto privacy-preserving scheme and investigated how it performs in terms of accuracy. Finally, we obtain **PPCF++** scheme by preprocessing **PPCF+** and examined its performance in terms of quality of predictions and online performance. Details of experimental procedures and results of conducted tests are explained in the following.

6.3.1 Evaluating the impact of random error on similarity calculation accuracy

In this section, we examine the impact of introducing random error into similarity calculations. Specifically, we investigate the extent to which the manipulation of vectors, through the addition of noise, affects the accuracy of these calculations. To this end, we compute the similarities between a designated active user and all other users across various levels of induced noise. These computed values are then compared with the similarity calculations obtained in the absence of perturbation. This process is repeated for each user, and the resulting average deviations from the original similarity values are quantified as percentages and presented in Table 3.

The results in Table 3 clearly demonstrates that as the level of random error increases, so too does the deviation in similarity values. These deviations, while unavoidable, are a necessary compromise to achieve the desired level of privacy through the implementation of perturbations. It's important to note that although these deviations tend to reduce toward zero over a series of trials, the results from a single trial are highlighted in this study. This is done to provide a clear perspective on the immediate impacts of such perturbations on the accuracy of similarity assessments, an aspect crucial for evaluating the trade-offs between privacy enhancement and computational accuracy in real-world scenarios.

6.3.2 Evaluating preprocessing technique in non-private schemes

In order to examine the effects of the proposed preprocessing scheme by means of scalability and accuracy, we applied such preprocessing on non-private scheme first. As [13] demonstrated, applying the similarity function onto traditional CF schemes improves accuracy. However, they do not perform any online performance test, which is vital for recommender systems. We demonstrated experimental results of accuracy and online performance for both similarity function enhanced **CF+** scheme and preprocessing enhanced **CF++** scheme. While utilizing τ for **CF++** scheme, although PCC takes values in the interval [− 1.0, +1.0] for item similarity calculations, we transformed such values to [0, +1.0] interval to form a common base with cosine similarity, which also takes values in the interval [0, +1.0]. Then, we varied τ from 0.05 to 0.5 in order to eliminate dissimilar items. To present a more clear comparison, we presented improvements of **CF+** and **CF++** schemes over traditional **CF** scheme in percentage. Comparison of the quality of predictions for ML and NF datasets is given in Figs. 3 and 4 respectively. Similarly, the comparison of online performance for ML and NF datasets is presented in Figs. 5 and 6, respectively.

Table 3 The percentages of average deviations in similarity values

Dataset	$\sigma_{\max}$ values					
	0.5	0.75	1	2	3	4
ML	− 1.95	− 7.57	− 11.52	− 15.60	− 17.58	− 20.39
NF	− 2.56	− 8.43	− 13.57	− 17.08	− 19.38	− 23.34

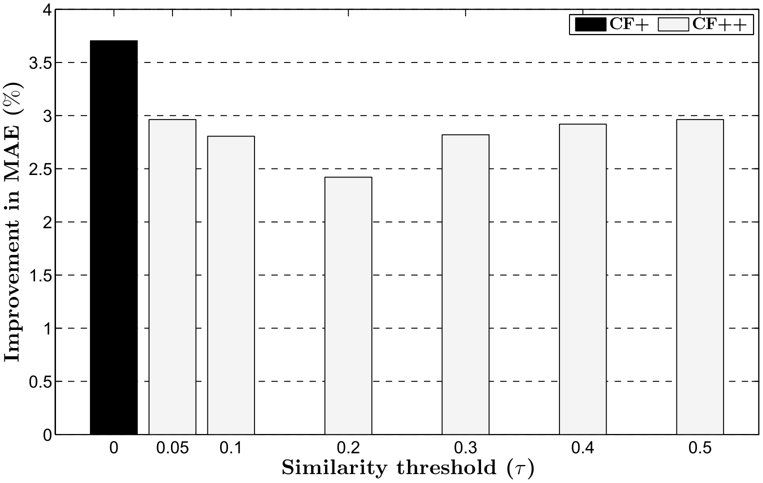

Fig. 3 Improvements on quality of predictions by varying τ values for **CF+** and **CF++** schemes on ML dataset

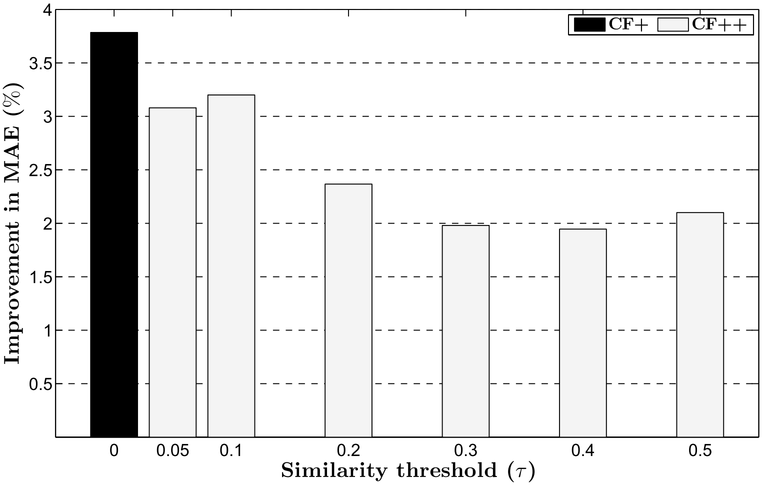

Fig. 4 Improvements on quality of predictions by varying τ values for **CF+** and **CF++** schemes on NF dataset

As seen from Figs. 3 and 4, **CF+** scheme achieves better accuracy improvements (around 4%) compared to our **CF++** scheme (around 3%). Especially for ML data set, such improvements can be obtained even for very high τ values. For ML data set, the best improvement is obtained at $\tau = 0.5$ with 2.96%, where **CF+** scheme achieves 3.7%. Due to its extreme sparse nature, increasing τ affects accuracy adversely for NF data set. Therefore, the best improvement for NF is obtained at $\tau = 0.1$ with 3.2%, where **CF+** scheme achieves 3.8%.

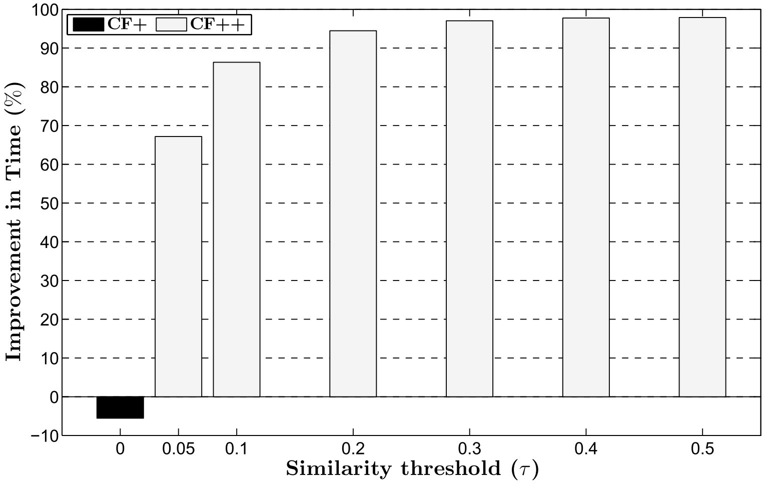

Fig. 5 Improvements on online performance by varying τ values for **CF+** and **CF++** schemes on ML dataset

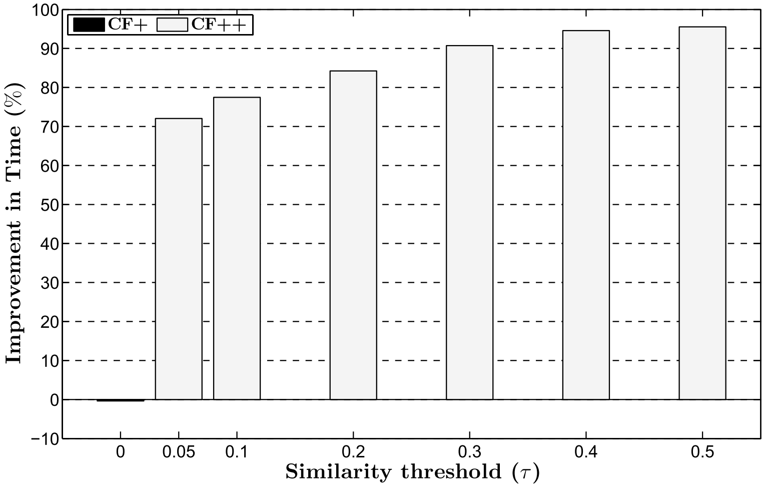

Fig. 6 Improvements on online performance by varying τ values for **CF+** and **CF++** schemes on NF dataset

Improvements in online performance, on the other hand, are vast. As can be followed from Figs. 5 and 6, **CF+** scheme introduces extra burden compared to traditional **CF** scheme and degrades performance by − 5.5% in ML data set. However, losses due to integration of the similarity function is negligible in NF data set due to extreme sparsity. Also, improvements due to preprocessing are significant for both data sets. As τ grows, online performances enhances for **CF++** scheme, as expected. Improvements are about 98 and 77% for ML and NF data sets, respectively, where the highest quality of predictions are obtained, i.e., $\tau = 0.5$ for ML and $\tau = 0.1$ for NF. We can conclude that employing the proposed preprocessing, at a level of satisfactory quality of predictions is obtained, is also beneficial for improving scalability.

6.3.3 Evaluation in privacy-preserving schemes

After examining the effects of the proposed preprocessing scheme on non-private CF schemes and determining optimum threshold values for different data sets, we experimented on privacy-preserving environment. We first investigated the effects of applying the similarity function onto traditional **PPCF** algorithm, which is called **PPCF+** scheme during experiments. We then implemented the preprocessing method onto **PPCF+** and derive **PPCF++** scheme. Similar to the experiments in non-private environment, we scrutinized the success of **PPCF+** and **PPCF++** schemes against traditional **PPCF** approach by varying τ from 0.05 to 0.5. For data disguising procedure, we kept standard deviation of produced random values ($\sigma_{\max}$) constant because there is no need to investigate such parameter's effects as it obviously deteriorates accuracy inline with the distortion amount, as shown in Fig. 2. Effects of different distortion values on quality of predictions is studied in [30, 60]. However, due to the results of [64, 65], utilizing $\sigma \leq 1$ may permit recovery of original data from perturbed values. Thus, we kept $\sigma_{\max} = 1.5$ during the experiments. Also, we kept maximum forgery rate ($\beta_{\max}$) constant at 25% as investigating its effects is out of scope of this study. Likewise, effects of varying $\beta_{\max}$ values on accuracy can be found in [60]. In addition, due to randomized selection of σ and β by each user, the experiments were repeated 100 times and average of the outcomes are demonstrated. Overall comparison results in privacy-preserving environment for ML and NF datasets are presented in Figs. 7 and 8, respectively.

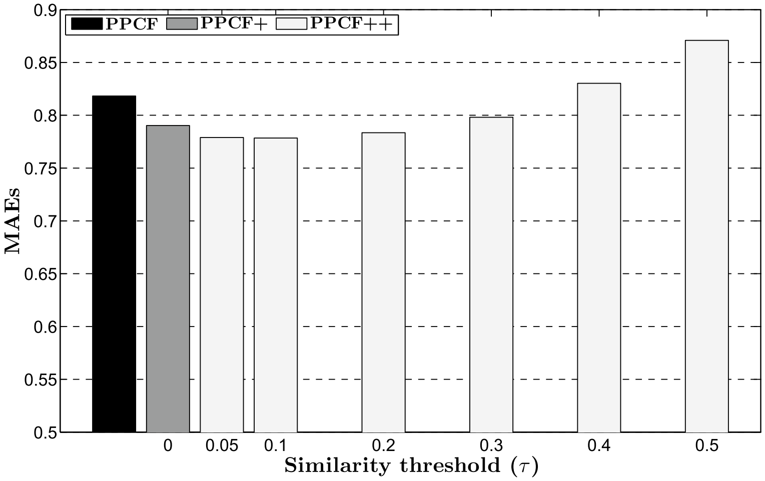

Fig. 7 Improvements on quality of predictions by varying τ values for **PPCF**, **PPCF+**, and **PPCF++** schemes on ML dataset

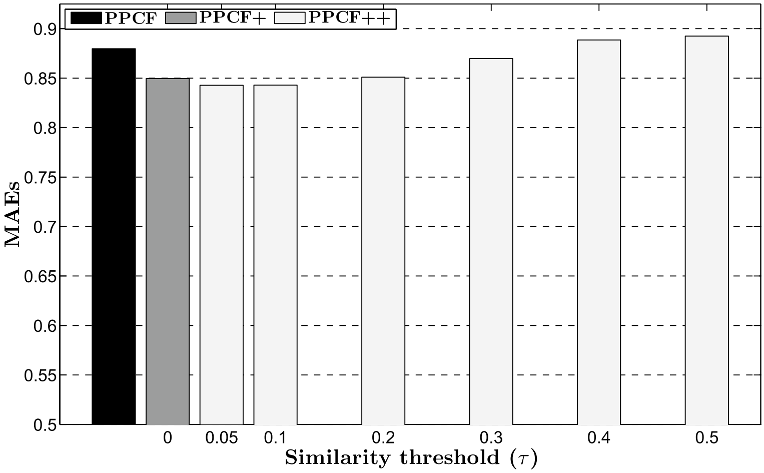

Fig. 8 Improvements on quality of predictions by varying τ values for **PPCF**, **PPCF+**, and **PPCF++** schemes on NF dataset

As demonstrated in Figs. 7 and 8, the similarity function is effective in privacy-preserving algorithms, as well. **PPCF+** scheme manages to reduce the error values by 3.42% (from 0.818 to 0.790) in ML data set and 3.53% (from 0.881 to 0.849) in NF data set compared to original **PPCF** scheme. Such improvements are similar to the ones achieved in non-private schemes. In addition, applying preprocessing causes further improvements in quality of predictions. For $\tau = 0.05$ and $\tau = 0.1$ values, **PPCF++** scheme performs better than **PPCF+** in both data sets. However, increased τ values like $\tau = 0.5$ cause too much loss of information; and therefore, accuracy diminishes. However, we can conclude that for $\tau <= 0.2$, **PPCF++** scheme is able to perform at least as good as **PPCF+** scheme in terms of accuracy for both data sets. Although improvements are similar to non-private schemes' experimental results, in order to present a clear overview, we demonstrated elapsed time to produce predictions in Table 4.

Table 4 presents elapsed time to produce five predictions for each active user in any data set, i.e, 9060 predictions for ML data set and 15,000 predictions for NF data set. It is clearly seen that **PPCF+** scheme brings extra online computational cost and the proposed **PPCF++** scheme enhances scalability with increasing τ values. Combining these experimental results with the ones presented in Figs. 7 and 8,

Table 4 Online performance by varying τ for **PPCF**, **PPCF+**, and **PPCF++** schemes

	PPCF	PPCF+	PPCF++					
			$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$
ML	1285	1351	416	171	70	37	28	26
NF	18,822	18,891	5247	4235	2964	1740	1019	838

we can conclude that applying the preprocessing technique with $\tau = 0.2$ provides an optimal performance in terms of both accuracy and scalability. Since in such arrangement, quality of predictions is very close to the values achieved by **PPCF+** scheme, yet online performance is enhanced by about 94% for ML and 84% for NF data sets.

7 Insights and limitations

Our study has introduced a novel preprocessing scheme integrated with a target item-based similarity function to enhance the scalability and accuracy of PPCF systems. The development of this approach was motivated by the challenges faced in memory-based collaborative filtering schemes, particularly those related to scalability, privacy, and the accuracy of recommendations. By applying a preprocessing method that eliminates relatively dissimilar items from the prediction estimation process, we achieved significant improvements in both conventional and privacy-preserving environments, enhancing system efficiency without compromising prediction accuracy. In a nutshell, the following summarizes the gained insights from the performed extensive set of experiments.

- The proposed preprocessing approach effectively addresses the scalability issues traditionally associated with memory-based collaborative filtering schemes by focusing on reducing the online response time significantly. This is achieved through the elimination of items that show relatively low similarity to the target item, thereby reducing the dimensions of the original user-item matrix and enhancing system performance.
- Our experiments, conducted on real datasets, have demonstrated that our preprocessing scheme significantly alleviates scalability problems in both conventional and privacy-preserving settings, while also improving accuracy within the privacy-preserving frameworks. These findings underscore the adaptability and effectiveness of our method in maintaining high performance under privacy constraints.
- The introduction of a target item-based similarity function, alongside our preprocessing scheme, showcases a promising direction for enhancing the quality of PPCF recommendations. This approach not only improves prediction accuracy by considering the relevance of items to the target item but also demonstrates the potential for scalable PPCF recommendations.

On the other hand, our study presents two significant limitations. Firstly, the experimental evaluation was carried out using specific datasets, and although the results are promising, the applicability of our preprocessing scheme and similarity function might differ across various domains and datasets. This necessitates further research to examine the generalizability of our approach to diverse CF applications and to evaluate its performance under different scenarios. Secondly, our study recognizes the challenge of balancing the trade-offs between privacy, accuracy, and computational efficiency. Although our approach contributes to the field by improving scalability and accuracy, finding an optimal equilibrium among these factors remains a complex challenge that warrants continued investigation.

In conclusion, our study provides valuable insights into enhancing the scalability and accuracy of PPCF systems through a novel preprocessing scheme and a target item-based similarity function. However, we recognize the limitations of our experimental evaluation and the need for further research to explore the full potential and applicability of our approach across different collaborative filtering scenarios.

8 Conclusions and future work

We investigated how to apply a target item-based similarity function on privacy-preserving collaborative recommender systems, which previously was adapted on non-private schemes and performed well. We theoretically examined how to integrate such similarity function onto privacy-preserving collaborative filtering architecture and showed its applicability. We also studied how individual privacy is preserved by following such scheme and quantified provided privacy levels due to the perturbation protocol. However, applying the similarity function introduces slight extra computational costs to the existing schemes. In order to alleviate this problem, we utilized item similarities, which were already computed for the similarity function. Motivating from the same idea of ranking item ratings with target item similarities, we proposed to eliminate relatively dissimilar items from the original matrix before calculating user similarities. Such elimination reduces the size of user-item matrix, which helps scaling the system. After analyzing our proposed preprocessing scheme with respect to overhead costs, we performed several experiments to scrutinize the effects of the scheme in both non-private and privacy-preserving environments. According to overall empirical outcomes, implementing the similarity function onto privacy-preserving framework results promising as the quality of predictions are enhanced like in non-private schemes. Moreover, the proposed preprocessing scheme achieves slightly better accuracy in privacy-preserving framework. On the other hand, improvements in terms of online performance are major, where traditional and accuracy-enhanced collaborative filtering and privacy-preserving collaborative filtering schemes are significantly outperformed.

Although improvements are similar in both MovieLens and Netflix data sets, the preprocessing scheme performs slightly better in MovieLens data set because it is more than three-times dense than Netflix data set. Since Netflix is extremely sparse, it is expected that there is already limited number of co-rated items between users, which makes harder to improve online response time by eliminating dissimilar items. Moreover, improvements in quality of predictions are also less than the values achieved in MovieLens data set. The same reason applies here, as well. Since the likelihood of finding co-rated items between users gets harder, item elimination causes too much loss of information, which result in worse accuracy. It was shown that combining the similarity function with the preprocessing achieves better accuracy and online performance in privacy-preserving framework.

In addition to numerical ratings, collaborative filtering systems also deal with binary preference data obtained by market-basket analysis and web logs. We are planning to modify the similarity function and the preprocessing method to be applied in binary rating-based systems, as well. Moreover, rather than memory-based prediction schemes, we are considering to build such improvement methods on model-based recommendation techniques and study their challenges in computing the similarity function as future research goals.

Author contributions Emre Yalcin: Methodology, Experiments, Visualization, and Article original draft preparation. Alper Bilge: Methodology, Conceptualization, Experiments, Investigation, Supervision, and Article review and editing.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availibility Data are available upon request from the authors.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Belabbes MA, Ruthven I, Moshfeghi Y, Rasmussen Pennington D (2023) Information overload: a concept analysis. J Doc 79(1):144–159
2. Himmelstein M, Budescu DV, Han Y (2023) The wisdom of timely crowds. In: Judgment in Predictive Analytics. Springer, Berlin, pp 215–242
3. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
4. Madathil M (2017) Music recommendation system spotify-collaborative filtering. Reports in Computer Music. Aachen University, Germany
5. Covington P, Adams J, Sargin E (2016) Deep neural networks for Youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp 191–198
6. Koren Y, Rendle S, Bell R (2021) Advances in collaborative filtering. Recommender systems handbook, pp 91–142
7. Batmaz Z, Yurekli A, Bilge A, Kaleli C (2019) A review on deep learning for recommender systems: challenges and remedies. Artif Intell Rev 52:1–37

8. Calandrino JA, Kilzer A, Narayanan A, Felten EW, Shmatikov V (2011) “you might also like:” privacy risks of collaborative filtering. In: 2011 IEEE Symposium on Security and Privacy. IEEE, pp 231–246
9. Bilge A, Gunes I, Polat H (2014) Robustness analysis of privacy-preserving model-based recommendation schemes. Expert Syst Appl 41(8):3671–3681
10. Gulsoy M, Yalcin E, Bilge A (2023) Robustness of privacy-preserving collaborative recommenders against popularity bias problem. PeerJ Comput Sci 9:1438
11. Bilge A, Polat H (2010) Improving privacy-preserving NBC-based recommendations by preprocessing. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol 1. IEEE, pp 143–147
12. Singh M (2020) Scalability and sparsity issues in recommender datasets: a survey. Knowl Inf Syst 62(1):1–43
13. Choi K, Suh Y (2013) A new similarity function for selecting neighbors for each target item in collaborative filtering. Knowl Based Syst 37:146–153
14. Singh PK, Sinha M, Das S, Choudhury P (2020) Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. Appl Intell 50:4708–4731
15. Singh PK, Sinha S, Choudhury P (2022) An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user-user similarity as weight. Knowl Inf Syst 64(3):665–701
16. Aggarwal CC, Aggarwal CC (2016) Content-based recommender systems. Recommender systems: the textbook, pp 139–166
17. Seth R, Sharaff A (2022) A comparative overview of hybrid recommender systems: review, challenges, and prospects. In: Data Mining and Machine Learning Applications, pp 57–98
18. Lima GR, Mello CE, Lyra A, Zimbrao G (2020) Applying landmarks to enhance memory-based collaborative filtering. Inf Sci 513:412–428
19. Li M, Wen L, Chen F (2021) A novel collaborative filtering recommendation approach based on soft co-clustering. Physica A 561:125140
20. Yu M, Quan T, Peng Q, Yu X, Liu L (2022) A model-based collaborate filtering algorithm based on stacked autoencoder. Neural Comput Appl 1–9
21. Yalcin E, Bilge A (2020) Binary multicriteria collaborative filtering. Turk J Electr Eng Comput Sci 28(6):3419–3437
22. Dong X, Yu L, Wu Z, Sun Y, Yuan L, Zhang F (2017) A hybrid collaborative filtering model with deep structure for recommender systems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 31
23. Xiong R, Wang J, Zhang N, Ma Y (2018) Deep hybrid collaborative filtering for web service recommendation. Expert Syst Appl 110:191–205
24. Vozalis MG, Markos A, Margaritis KG (2010) Collaborative filtering through SVD-based and hierarchical nonlinear PCA. In: International Conference on Artificial Neural Networks. Springer, Berlin, pp 395–400
25. Nilashi M, Ibrahim O, Bagherifard K (2018) A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. Expert Syst Appl 92:507–520
26. Zarzour H, Al-Sharif Z, Al-Ayyoub M, Jararweh Y (2018) A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In: 2018 9th International Conference on Information and Communication Systems (ICICS). IEEE, pp 102–106
27. Jiang M, Zhang Z, Jiang J, Wang Q, Pei Z (2019) A collaborative filtering recommendation algorithm based on information theory and bi-clustering. Neural Comput Appl 31:8279–8287
28. Logesh R, Subramaniyaswamy V, Malathi D, Sivaramakrishnan N, Vijayakumar V (2020) Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. Neural Comput Appl 32:2141–2164
29. Xiaojun L (2017) An improved clustering-based collaborative filtering recommendation algorithm. Clust Comput 20:1281–1288
30. Polat H, Du W (2005) Privacy-preserving collaborative filtering. Int J Electron Commer 9(4):9–35
31. Canny J (2002) Collaborative filtering with privacy. In: Proceedings 2002 IEEE Symposium on Security and Privacy. IEEE, pp 45–57

32. Canny J (2002) Collaborative filtering with privacy via factor analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 238–245
33. Badsha S, Yi X, Khalil I (2016) A practical privacy-preserving recommender system. Data Sci Eng 1(3):161–177
34. Li D, Lv Q, Shang L, Gu N (2017) Efficient privacy-preserving content recommendation for online social communities. Neurocomputing 219:440–454
35. Li D, Chen C, Lv Q, Shang L, Zhao Y, Lu T, Gu N (2016) An algorithm for efficient privacy-preserving item-based collaborative filtering. Futur Gener Comput Syst 55:311–320
36. Shmueli E, Tassa T (2017) Secure multi-party protocols for item-based collaborative filtering. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp 89–97
37. Badsha S, Yi X, Khalil I, Liu D, Nepal S, Lam K-Y (2018) Privacy preserving user based web service recommendations. IEEE Access 6:56647–56657
38. Zou J, Fekri F (2015) A belief propagation approach to privacy-preserving item-based collaborative filtering. IEEE J Sel Top Signal Process 9(7):1306–1318
39. Casino F, Domingo-Ferrer J, Patsakis C, Puig D, Solanas A (2015) A k-anonymous approach to privacy preserving collaborative filtering. J Comput Syst Sci 81(6):1000–1011
40. Chen X, Huang V (2012) Privacy preserving data publishing for recommender system. In: 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops. IEEE, pp. 128–133
41. Wei R, Tian H, Shen H (2018) Improving k-anonymity based privacy preservation for collaborative filtering. Comput Electr Eng 67:509–519
42. Zhang F, Lee VE, Choo K-KR (2018) Jo-dpmf: differentially private matrix factorization learning through joint optimization. Inf Sci 467:271–281
43. Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. Springer, Berlin, pp 265–284
44. McSherry F, Mironov I (2009) Differentially private recommender systems: building privacy into the Netflix prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 627–636
45. Guerraoui R, Kermarrec A-M, Patra R, Taziki M (2015) D 2 p: distance-based differential privacy in recommenders. Proceedings of the VLDB Endowment 8(8):862–873
46. Shen Y, Jin H (2014) Privacy-preserving personalized recommendation: an instance-based approach via differential privacy. In: 2014 IEEE International Conference on Data Mining. IEEE, pp 540–549
47. Hou M, Wei R, Wang T, Cheng Y, Qian B (2018) Reliable medical recommendation based on privacy-preserving collaborative filtering. Comput Mater Continua 56(1):137–149
48. Parameswaran R, Blough DM (2007) Privacy preserving collaborative filtering using data obfuscation. In: 2007 IEEE International Conference on Granular Computing (GRC 2007. IEEE), pp 380–380
49. Badsha S, Yi X, Khalil I, Bertino E (2017) Privacy preserving user-based recommender system. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp 1074–1083
50. Boutet A, Frey D, Guerraoui R, Jégou A, Kermarrec A-M (2016) Privacy-preserving distributed collaborative filtering. Computing 98(8):827–846
51. Elmisery AM, Botvich D (2017) An enhanced middleware for collaborative privacy in IPTV recommender services. arXiv preprint arXiv:1711.07593
52. Polat H, Du W (2003) Privacy-preserving collaborative filtering using randomized perturbation techniques. In: Third IEEE International Conference on Data Mining. IEEE, pp 625–628
53. Polat H, Du W (2005) Privacy-preserving collaborative filtering on vertically partitioned data. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, pp 651–658
54. Bilge A, Polat H (2013) A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. Inf Process Manag 49(4):912–927
55. Gong S (2011) Privacy-preserving collaborative filtering based on randomized perturbation techniques and secure multiparty computation. Int J Adv Computg Technol 3(4):89–99
56. Polatidis N, Georgiadis CK, Pimenidis E, Mouratidis H (2017) Privacy-preserving collaborative recommendations based on random perturbations. Expert Syst Appl 71:18–25
57. Liu X, Liu A, Zhang X, Li Z, Liu G, Zhao L, Zhou X (2017) When differential privacy meets randomized perturbation: a hybrid approach for privacy-preserving recommender system. In: International Conference on Database Systems for Advanced Applications. Springer, Berlin, pp 576–591

58. Yargic A, Bilge A (2019) Privacy-preserving multi-criteria collaborative filtering. Inf Process Manag 56(3):994–1009
59. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst (TOIS) 22(1):5–53
60. Bilge A, Polat H (2012) An improved privacy-preserving dwt-based collaborative filtering scheme. Expert Syst Appl 39(3):3841–3854
61. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423
62. Agrawal D, Aggarwal CC (2001) On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp 247–255
63. Bilge A, Polat H (2013) A comparison of clustering-based privacy-preserving collaborative filtering schemes. Appl Soft Comput 13(5):2478–2489
64. Huang Z, Du W, Chen B (2005) Deriving private information from randomized data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp 37–48
65. Kargupta H, Datta S, Wang Q, Sivakumar K (2003) On the privacy preserving properties of random data perturbation techniques. In: Third IEEE International Conference on Data Mining. IEEE, pp 99–106

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.