Complex & Intelligent Systems (2025) 11:107
https://doi.org/10.1007/s40747-024-01758-9

ORIGINAL ARTICLE

FSPPCFs: a privacy-preserving collaborative filtering recommendation scheme based on fuzzy C-means and Shapley value

Weiwei Wang[1] · Wenping Ma[1] · Kun Yan[1]

Received: 23 April 2024 / Accepted: 15 December 2024 / Published online: 30 December 2024

Abstract
Collaborative filtering recommendation systems generate personalized recommendation results by analyzing and collaboratively processing a large numerous of user ratings or behavior data. The widespread use of recommendation systems in daily decision-making also brings potential risks of privacy leakage. Recent literature predominantly employs differential privacy to achieve privacy protection, however, many schemes struggle to balance user privacy and recommendation performance effectively. In this work, we present a practical privacy-preserving scheme for user-based collaborative filtering recommendation that utilizes fuzzy C-means clustering and Shapley value, FSPPCFs, aiming to enhance the recommendation performance while ensuring privacy protection. Specifically, (i) we have modified the traditional recommendation scheme by introducing a similarity balance factor integrated into the Pearson similarity algorithm, enhancing recommendation system performance; (ii) FSPPCFs first clusters the dataset through fuzzy C-means clustering and Shapley value, grouping users with similar interests and attributes into the same cluster, thereby providing more accurate data support for recommendations. Then, differential privacy is used to achieve the user's personal privacy protection when selecting the neighbor set from the target cluster. Finally, it is theoretically proved that our scheme satisfies differential privacy. Experimental results illustrate that our scheme significantly outperforms existing methods.

Keywords Privacy protection · Collaborative filtering · Fuzzy C-means clustering · Shapley value · Recommendation system

Introduction

Big data has become ubiquitous in all aspects of daily life due to the high-speed advancement of information technology. To solve the issue of information overload, the recommendation algorithm was launched in the Internet industry, which can collect an enormous amount of data to predict a particular user's preferences for various items or content [1–3]. Advances in recommendation systems have greatly motivated the steady flourishing of countless applications. With the rise of e-commerce and self-media platforms, which rely heavily on recommendation systems to improve user engagement and satisfaction, collaborative filtering has received increasing attention in recent years. Hence, numerous researchers have dedicated their efforts to developing novel and inventive collaborative filtering algorithms, along with exploring their potential applications across diverse domains [4–7]. In recent years, many novel technologies have emerged in the research of recommendation algorithms in new fields, especially the application of deep learning, which has made significant contributions. Bhatia et al. [8] combined deep learning with semantic fusion methods to improve the effect of recommendation systems. Li et al. [9] explored user preference data through deep learning and proposed a movie recommendation method. Fu et al. [10] proposed a new collaborative filtering model based on deep learning. However, the excellent performance of deep learning models usually depends on a lot of tuning and may need to be retrained to maintain the effect when applied across fields. If the data can be preprocessed using methods such as clustering, it will be more suitable for diverse recommendation scenarios.

Despite already having remarkable ability in our daily decision-making, recommendation systems still suffer from major threats of privacy and security issues [11, 12] due to the massive collection of personal data. To address the issue of privacy protection, the concept of differential privacy, initially proposed by Dwork et al. in [13], serves as a robust

✉ Weiwei Wang
wweiwei@stu.xidian.edu.cn

[1] School of Telecommunication Engineering, Xidian University, Xi'an 710071, China

mechanism for ensuring privacy and boasts rigorous mathematical proofs, thereby significantly reducing the risk of private information leakage. Numerous research papers have been published in recent years to explore the use of differential privacy in recommendation systems to fulfill the goal of privacy protection [14–16]. However, differential privacy generally makes quite a lot of noise, which has an effect on the quality of recommendations. As a result, existing approaches suffer from significant accuracy loss even when providing appropriate privacy guarantees. It is evident that the incorporation of differential privacy leads to a degradation in recommendation performance. Hence, striking a win-win between individual privacy and performance in recommendation systems remains a crucial and widely discussed topic.

It is very challenging to ensure the accuracy of recommendations while protecting the privacy of users' personal information through differential privacy [16]. To address the issue of user privacy leakage, differential privacy was effectively incorporated into the collaborative filtering recommendation systems by Zhu et al. in [17]. They proposed DP-UR/DP-IR schemes for user/item-based collaborative filtering, respectively, using exponential mechanisms to realize privacy protection. In order to achieve a better compromise between user privacy and recommendation performance, Chen et al. [16] proposed a privacy-preserving collaborative filtering system utilizing K-means clustering is used as a means of data preprocessing, and an exponential mechanism is used to achieve privacy protection. However, Koohi et al. [18] applied fuzzy C-means clustering to the collaborative filtering recommendation system and compared it with the recommendation scheme utilizing K-means clustering. The final experiment demonstrated that the recommendation performance utilizing fuzzy C-means clustering methods was better than K-means clustering-based methods. Unfortunately, Koohi et al. [18] does not consider privacy protection in the whole paper.

Based on the preceding analysis and inspired by the work of [18], we employ fuzzy C-means clustering in conjunction with the Shapley value to develop a privacy-preserving collaborative filtering recommendation scheme. Additionally, we introduce the concept of a similarity balance factor and propose an adjusted Pearson similarity calculation method to improve recommendation performance. In summary, we present a practical privacy-preserving user-based collaborative filtering recommendation scheme that integrates fuzzy C-means clustering and the Shapley value.

Contributions. The main contributions of our work are summarized as follows:

- Taking into account the influence of different user behaviors in the recommendation process, we introduce the concept of similarity balance factor and integrate it into the traditional Pearson similarity algorithm to obtain an adjusted Pearson similarity algorithm, which can improve the accuracy of the recommendation.
- We introduce the Shapley value as features of user intrinsic attributes to enable it to more accurately capture the internal relationships and preferences between users. Specifically, in the data preprocessing stage, the Shapley value is used to quantify the contribution of each feature, which can significantly improve the prediction accuracy of the recommendation system, thereby enhancing the user experience.
- We propose a novel privacy-preserving scheme for user-based collaborative filtering recommendation utilizing fuzzy C-means clustering and Shapley value. It is theoretically proved that the proposed scheme achieves ϵ-differential privacy.
- To assess the superiority of the proposed scheme, we conducted simulation experiments on the MovieLen 100K and FilmTrust datasets. The final data results demonstrate that our proposed scheme is better than the others, and can also provide robust privacy protection.

Organization. The organizational structure of the paper is shown in Fig. 1. The current section is the Introduction. We briefly introduce some existing work in Sect. "Literature overview". Section "Technical preliminaries" presents the relevant basic knowledge involved in this work, which includes fuzzy C-means clustering, differential privacy, and Shapley value, and it is also a brief review of collaborative filtering recommendations. Section "Proposed method" mainly introduces the relevant details of our scheme. We primarily did a lot of comparative experiments and carried out detailed discussion and analysis in Sect. "Experiment and evaluation". In the end, we summarize the conclusions of this paper and some ideas for future research in Sect. "Conclusion and future works".

Literature overview

Clustering-based recommendation systems

Traditional recommendation systems often encounter performance limitations, which can be addressed by employing clustering techniques to classify user datasets into relatively dense groups of similar users, thereby enabling more accurate recommendations. This section briefly reviews existing clustering-based recommendation system approaches in three categories: Fuzzy C-means clustering(FCM)-based methods, K-means clustering (KM)-based methods, and other clustering-based methods.

FCM-based methods. Koohi et al. [18] initially proposed the FCM method for collaborative filtering recommendation. In their study, the authors sought to apply the FCM

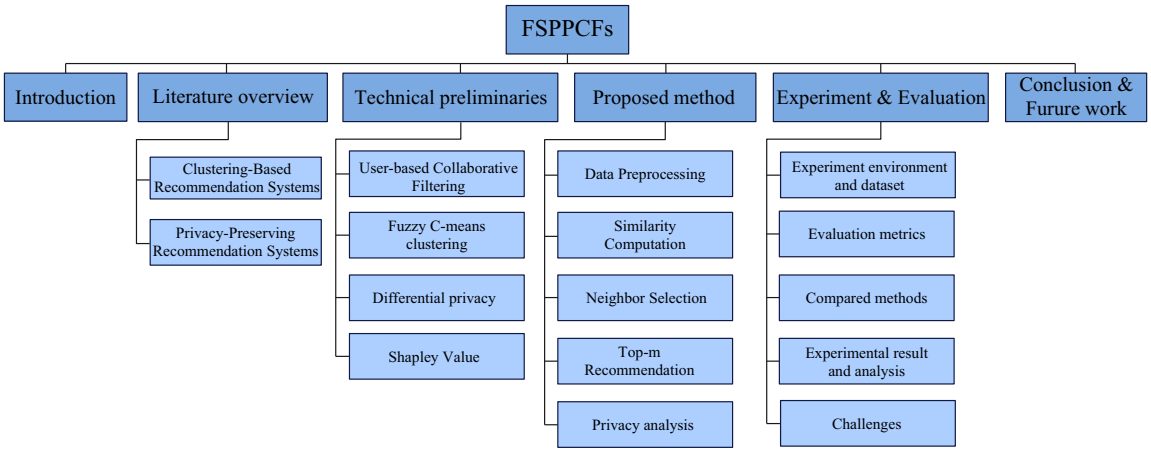

Fig. 1 The organizational structure of the paper

algorithm in user-based collaborative filtering recommender systems and compared it with the KM and self-organizing map (SOM) approaches. The experimental results clearly demonstrate that the FCM algorithm outperforms both the KM and SOM methods, delivering superior outcomes. However, the authors did not deeply explore the intrinsic social connections among users nor address user privacy protection. Similarly, Zhong et al. [19] have further proposed a collaborative filtering scheme based on the FCM algorithm, which is applied to items to enhance recommendation accuracy. In contrast to [18], however, this paper introduced the time weight function to recommendation results achieve better. To tackle the cold start issue, Duan et al. [20] proposed a hybrid recommender system based on FCM clustering and supervised learning. Although their experimental results show that the proposed scheme outperforms the baseline algorithm in both recommendation and prediction accuracy, their method does not consider the defuzzification of FCM, which limits its practical applicability. It is worth noting that we employ the FCM algorithm for collaborative filtering recommendations in this study. Unlike the methods proposed in [18, 19], our approach primarily integrates the FCM algorithm with the adjusted Pearson similarity algorithm. We introduce a similarity balance factor into the traditional Pearson similarity, which significantly enhances the performance of the recommendation system.

KM-based methods. Several researchers have proposed collaborative filtering recommendation algorithms based on KM to address the challenges of data sparsity and expansibility. Liu et al. [21] developed a collaborative filtering recommendation algorithm using bisecting KM. In a similar vein, Chen et al. [22] proposed a more effective collaborative filtering recommendation scheme utilizing user attributes and KM. To further enhance the performance of recommendation systems, Zarzour et al. [23] presented a KM ensemble-based method to promote recommendation systems. From the viewpoint of item probability, Deng et al. [24] proposed an innovative K-medoids clustering recommendation algorithm, which can better find the cluster center to achieve a better recommendation effect. Although the KM method can effectively improve the recommendation performance, the experimental results of [18] show that it is inferior to the FCM method.

Others clustering-based methods. Chen et al. [25] used user correlation and evolutionary clustering to put forward an innovative collaborative filtering recommendation method. It works by preprocessing the scoring matrix, generating clustering principles, and applying dynamic evolutionary clustering to find the nearest neighbors with the highest similar interest. Jiang et al. [26] introduced a collaborative filtering recommendation algorithm based on the combination of double filtering and information quotient, which can overcome the issues of data sparsity and heterogeneity in recommendation systems. To tackle the problem of soft clustering, the SKAP algorithm and a recommendation method based on soft co-clustering are proposed in [27]. This method incorporates item type as additional information in the recommendation process to address soft clustering issues with high dimensions. However, due to the broad process of this method, it may lead to the inclusion of some irrelevant users. Additionally, the experimental results in [27] demonstrate that the SKAP algorithm outperforms FCM. However, the article does not specify whether defuzzification was employed in the application of the FCM algorithm.

Privacy-preserving recommendation systems

Generic cryptography technology is the mainstream method to solve the privacy protection problem in recommendation systems. Here we briefly introduce the existing work from three aspects: (a) fully homomorphic encryption(FHE)-

based techniques, (b) differential privacy(DP)-based techniques, and (c) federated learning(FL)-based techniques.

FHE-based techniques. BGV-CF [28] is the latest FHE-based privacy protection recommendation scheme, which mainly utilizes BGV encryption and SV packing to present a privacy-preserving recommendation scheme under the semi-honest model. Kim et al. [29] introduced a unique privacy-preserving matrix factorization for recommendation utilizing FHE. TMFH-DEM is a novel cryptographic primitive that is a multikey fully homomorphic data encapsulation scheme based on tags. Using TMFH-DEM as a building block, Zhou et al. [30] designed a lightweight privacy-preserving recommendation scheme. CryptoRec [31] is a novel collaborative filtering recommendation that relies only on addition and multiplication operations, which are directly compatible with operations of homomorphic encryption schemes. CryptoRec enables a better compromise between privacy and utility. Although FHE technology can achieve strong privacy protection, it is not realistic in the actual application of recommendation systems.

DP-based techniques. PNCF [32] is an influential neighborhood-based privacy-preserving recommendation algorithm, which mainly covers two aspects: one is private neighbor selection, and the other is a perturbation. PNCF algorithm based on differential privacy can obtain strong privacy protection while reducing recommendation accuracy loss. PrivateRS [33] is a novel recommendation system framework, which can enjoy accurate recommendation services on untrusted servers while achieving privacy protection. Chen et al. [16] designed a privacy-preserving collaborative filtering recommendation scheme utilizing KM, known as KDPCF, which aims to enhance the recommendation performance by reducing the frequency of utilization of the exponential mechanism. While the experimental results show that KDPCF performs better than PNCF, it is constrained by the previous discussion on KM and its performance is not as good as FCM. Hence, there remains significant potential for improvement.

FL-based techniques. The emerging technology of federated learning is of great significance in terms of privacy and security protection [34]. Feng et al. [35] designed a privacy-preserving multimodal recommendation system framework based on federated learning technology, aiming to solve the model convergence problem existing in current single-modal learning. In addition, the literature [36] explores the challenges and difficulties faced by federated recommendation systems in the latest models, especially technical issues such as network costs and performance requirements that need to be addressed in practical application scenarios. It is worth noting that recommendation systems are vulnerable to external attacks and threats in practical applications, and the introduction of federated learning technology can effectively alleviate these problems [37–39]. Furthermore, Yan et al. [40] utilized evolutionary algorithms to enable resource-sharing recommendations in communications and networks while preserving user privacy, providing a new direction for privacy-preserving recommendations.

Table 1 Rating matrix

	Item v_1	$\cdots$	Item v_j	$\cdots$	Item v_m
User u_1	$r_{1,1}$	$\cdots$	$r_{1,j}$	$\cdots$	$r_{1,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
User u_i	$r_{i,1}$	$\cdots$	$r_{i,j}$	$\cdots$	$r_{i,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
User u_n	$r_{n,1}$	$\cdots$	$r_{n,j}$	$\cdots$	$r_{n,m}$

Technical preliminaries

In this part, we mainly introduce the relevant basic knowledge in this work, which includes fuzzy C-means clustering, differential privacy, and Shapley value, and it is also a brief review of collaborative filtering recommendation.

User-based collaborative filtering

A widely used recommendation approach, user-based collaborative filtering [16, 41], analyzes user behavior history data to identify similar users and recommends items to the target user based on their behavioral patterns. A user's preference for a specific item is expressed through a rating, which is a non-negative integer. Assume that $U \triangleq \{u_1, u_2, \ldots, u_n\}$ be a set of n users, and $I \triangleq \{v_1, v_2, \ldots, v_m\}$ be a set of m items. We define $r_{i,j}$ as user u_i's rating of item v_j. Then, the set of item ratings in recommender systems is represented by an $n \times m$ matrix, as illustrated in Table 1.

The basic idea underlying user-based collaborative filtering is that users who have shown similar preferences for goods in the past are likely to have similar preferences for similar items in the future. Hence, the algorithm begins by calculating user similarities. User-based collaborative filtering with the Pearson correlation coefficient is a specific type of recommendation algorithm that employs the Pearson correlation coefficient to quantify the similarity between users [42]. The Pearson similarity between users u_a and u_b can be computed as follows:

$$\mathrm{sim}\,(u_a, u_b) = \frac{\sum_{i \in I_{a,b}} \left(R_{a,i} - \overline{R_a}\right)\left(R_{b,i} - \overline{R_b}\right)}{\sqrt{\sum_{i \in I_{a,b}} \left(R_{a,i} - \overline{R_a}\right)^2 \left(R_{a,i} - \overline{R_a}\right)^2}}, \tag{1}$$

where sim (u_a, u_b) is the similarity score between users u_a and u_b, $R_{a,i}$ and $R_{b,i}$ are the ratings of item i by users u_a and u_b, respectively. $\overline{R_a}$ and $\overline{R_b}$ are the average rating scores of users u_a and u_b, respectively. $I_{a,b}$ represents the set of items rated by both users u_a and u_b.

Then, identify the target user's neighbor set by selecting the Top-m users based on similarity scores. The scores for unrated items can be predicted using the following formula:

$$p_{a,i} = \overline{r_{u_a}} + \frac{\sum_{u_b \in N} \text{sim}\,(u_a, u_b) \times \left(R_{b,i} - \overline{R_b}\right)}{\sum_{u_b \in N} \text{sim}\,(u_a, u_b)}, \tag{2}$$

where N is the nearest neighbor set of user u_a. Finally, the top-m recommended items for user u_a can be selected based on the predicted ratings $p_{a,i}$.

Fuzzy C-means clustering

One of the numerous clustering techniques is FCM [43], which is a way of soft clustering, allowing the value of a single attribute value of data to belong to at least two clusters. Generally speaking, it is impossible to separate the objects in the data set into separate clusters. The rigidity of assigning an object to a certain cluster can result in mistakes. As a result, each object and each cluster are assigned a weight to represent how much the object is related to each cluster. However, finding an appropriate statistical model might be challenging at times. As a result, it is preferable to utilize the FCM with natural and non-probabilistic characteristics. FCM constructs fuzzy partitions consisting of C clusters, in which each element becomes a member of multiple clusters. The primary goal of FCM is to minimize the membership value of elements, as defined by the following objective function: $J = \sum_{k=1}^{n} \sum_{i=1}^{n} (u_{ik})^m (d_{ik})^2$ and $\sum_{k=1}^{n} u_{ik} = 1$, where $(d_{ik})^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)^T A (x_k - v_i)$, and $A_{n \times n}$ is a norm matrix. The fuzzy C-means clustering algorithm is listed in Algorithm 1.

Algorithm 1 Classic FCM algorithm [44]

Require: X: data set; m: fuzzy index; C: number of clusters; ξ: threshold value

Ensure: Membership matrix U; Cluster center matrix V;

1: Initialize the membership matrix $U = [u_{ik}]$;

2: Compute the cluster center $V = [v_i]$ using the membership matrix $U = [u_{ik}]$, where

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}$$

3: Update the membership matrix U by computing

$$u_{ik} = \frac{1}{\sum_{j=1}^{n} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|}\right)^{\frac{2}{m-1}}}$$

4: Repeat 2 and 3 until the termination condition $\|v^{k+1} - v^k\| < \xi$ and stop the iteration; otherwise, return to step 2.

Differential privacy

Differential privacy [13, 45] is a widely used privacy protection technique that aims to analyze and mine personal data while minimizing the risk of personal privacy leakage. Differential privacy makes it difficult for an attacker to infer any sensitive information about a specific individual from the results by introducing a certain level of noise or perturbation into the data set. In recommendation systems, differential privacy can help protect users' personal information, encourage data sharing and collaboration, and promote a balance between privacy protection and data utilization.

Definition 1 (*ϵ-Differential Privacy* [13]) A randomized algorithm $\mathcal{M}$ is said to satisfy ϵ-differential privacy if for any pair of neighboring datasets D and D' that differ on at most one element, and for any set of outcomes $\mathcal{S} \subseteq Range(\mathcal{M})$, the algorithm $\mathcal{M}$ satisfies the following property:

$$P[\mathcal{M}(D) \in \mathcal{S}] \leq exp(\epsilon) \cdot P\left[\mathcal{M}\left(D'\right) \in \mathcal{S}\right], \tag{3}$$

where ϵ is called the privacy budget, which determines the algorithm's level of privacy. Generally speaking, the smaller ϵ is, the more noise is added, which means that it has a good privacy protection effect.

Exponential Mechanism [45] is a differential privacy protection algorithm for limited selection or sorting tasks on a given data set. It balances the privacy protection and practicality of the results by introducing randomness and probability distribution.

Definition 2 (*Exponential Mechanism* [45]) The exponential mechanism $\mathcal{M}_E(x, u, \mathcal{R})$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $exp(\frac{\epsilon u(x,r)}{2\triangle u})$, where $u(x, r)$ is a quality function and the sensitivity of the quality function is denoted by Δu.

Definition 3 (*Sensitivity* [45]) The exponential mechanism's sensitivity is described as follows

$$\Delta u = \max_{r \in \mathcal{R}} \max_{x, x' : \|x - x'\| \leq 1} \left|u(x, r) - u(x', r)\right|, \tag{4}$$

where $r \in \mathcal{R}$ is a valid exponential mechanism output and $u(x, r)$ is a quality function.

Shapley value

Shapley value belongs to the category of cooperative game theory [46], which is a distribution method based on contribution. A cooperative game theory can be defined as the pair (N, v), where N defines the set of players and $v : 2^N \to \mathbb{R}$ is a real-valued function with $v(\emptyset) = 0$, where $\emptyset$ denotes the empty set. The v is named a characteristic function. In

the realm of cooperative game theory, an analysis of a cooperative game entails the application of a solution concept that offers a systematic approach to apportioning the game's overall value among the participating individual players. The Shapley value is the most significant of the numerous original solutions that have come to light with the development of the cooperative game.

The Shapley value [47] offers a distinctive solution for allocating expected payoffs in a particular coalitional game (N, v). It presents a practical framework for achieving an equitable distribution of gains resulting from collaboration among players in cooperative games. It is essential to ensure a fair distribution of these advantages among the participants because certain players may make a greater overall value contribution than others. The Shapley value concept addresses this concern by considering the corresponding significance of each participant's contribution to the game when determining the allocation of payoffs to the individuals involved. The following formula is used to compute the Shapley value of player i:

$$\phi_i(N, v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \{v(S \cup \{i\}) - v(S)\}, \tag{5}$$

where $i \in N$ and $S \subseteq N$. The number of participants in the game is denoted here by $|S|$. So the Shapley value can be written as

$$\phi(N, v) = (\phi_1(N, v), \phi_2(N, v), \cdots, \phi_n(N, v)).$$

Proposed method

To enhance the performance of the recommendation process and ensure the privacy of users, we present a privacy-preserving user-based collaborative filtering recommendation scheme utilizing FCM, FPPCFs. Additionally, we propose the inclusion of the Shapley value as a feature in users' intrinsic attributes to enhance personalization and accuracy in the recommendation system. Specifically, during the data processing stage, the Shapley value is utilized to comprehensively consider the internal relationships among users. Building upon these principles, we present a privacy-preserving user-based collaborative filtering recommendation scheme using FCM and Shapley value, FSPPCFs. The main flowchart of the scheme is shown in Fig. 2, which consists primarily of four steps: (a) Data preprocessing, (b) Similarity computation, (c) Neighbor selection, and (d) Top-m recommendation. The detailed process of the scheme is outlined as follows.

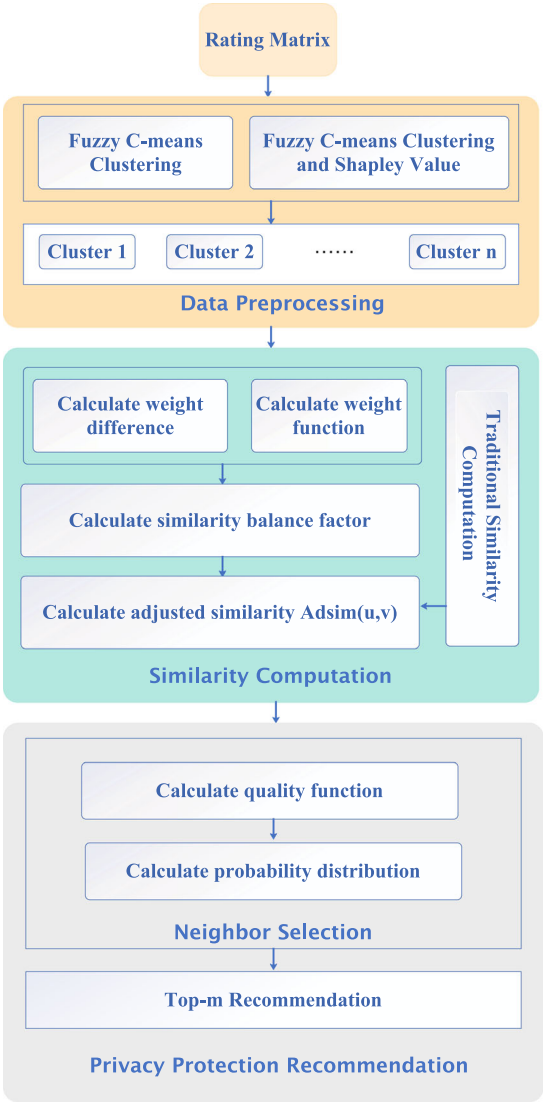

Fig. 2 FSPPCFs/FPPCFs scheme flowchart

Data preprocessing

Fuzzy C-means clustering. With the exponential growth of information, the rating matrix of recommendation systems often contains a vast number of users. Consequently, certain users may have minimal impact on the current recommendations, while only a subset of users significantly contributes to the quality of the recommendations. Therefore, it becomes necessary to preprocess the data before making recommendations. To enhance the performance and accuracy of subsequent recommendations, we propose utilizing FCM as a preprocessing technique for improved target classification in the rating matrix. This approach aims to identify meaningful user clusters that can aid in achieving more effective recommendations.

The FCM algorithm used in this section is described in Algorithm 1. We choose the traditional FCM method with a fuzzy index of 2, but it is worth noting that the result of FCM is still a fuzzy set, as we all know, and we always want to have a clearer bound in solving real problems, so we need to defuzzify, the advantage of this is that fuzzy clustering can be transformed into a deterministic classification by some rules. Here, we use the maximum degree of membership method to defuzzify [48], that is, take the element with the largest degree of membership as the output value. Furthermore, previous research [18, 41] has demonstrated that the recommendation performance tends to decline with an increasing number of clusters. Considering this finding, we have chosen to set the number of clusters to 2 in this work.

Fuzzy C-means and Shapley value. The Shapley value is a metric from game theory that quantifies the influence of participants on game outcomes. It can be employed in our system to assess each user's contribution to the recommendation results. Incorporating the Shapley value into the privacy-preserving collaborative filtering recommendation system utilizing FCM will enhance our understanding of each user's contribution to the recommendation results. This approach allows for user grouping and the generation of personalized recommendations for various user combinations, thus improving the precision and accuracy of the recommendation system.

Assume that $U \triangleq \{u_1, u_2, \ldots, u_n\}$ be a set of n users. Given the user set U, define a function, $\rho : U \times U \to [0, 1]$, and the function ρ in our model is defined as: $\rho(u_i, u_j) = 1 - \left|\text{sim}(u_i, u_j)\right|$, where $\text{sim}(u_i, u_j)$ can be obtained by Eq. (1) and $u_i, u_j \in U$. Next, we need to specify a monotonic non-increasing similarity function $\mathcal{S}$ with respect to ρ. Specifically, the similarity function $\mathcal{S} : [0, 1] \to (0, 1]$ is defined as follows:

$$\mathcal{S}\left(\rho\left(u_i, u_j\right)\right) = 1 - \frac{\rho\left(u_i, u_j\right)}{\rho_{max}\left(u_i, u_j\right) + 1}, \tag{6}$$

where $\rho_{max}(u_i, u_j)$ is the maximum similarity between u_i and u_j. According to cooperative game theory, in our model, each u_i can be regarded as a player and the total number of players can be the size of the user set U: $|U| = n$. Therefore, we can set up a cooperative game (N, v) among the data points. Then, we need to introduce a value function v, intrinsically linked to forming cooperative games. In a cooperative game, participants collaborate to optimize the collective value represented by v. It is explicitly stated that an individual coalition (consisting of a single player) is not allowed, denoted as $v(u_i) = 0$. More specifically, with the above definition of similar functions, we can define the value function of alliance C through the following formula:

$$v(C) = \frac{1}{2} \sum_{\substack{u_i, u_j \in U \\ u_i \neq u_j}} \mathcal{S}\left(\rho\left(u_i, u_j\right)\right). \tag{7}$$

Then we know from [47] that the Shapley value of our model can be calculated by Eq. (9). Therefore, the Shapley value of user u_i in our model can be given by:

$$\varphi_i = \frac{1}{2} \sum_{\substack{u_j \in U \\ j \neq i}} \mathcal{S}\left(\rho\left(u_i, u_j\right)\right). \tag{8}$$

Algorithm 2 Data preprocessing using FCM and Shapley value

Require: User-Item Rating Matrix; number of clusters C
Ensure: Cluster results to which each user belongs;
1: Extract the set of users from the rating matrix, defined as $U \triangleq \{u_1, u_2, \ldots, u_n\}$;
2: Calculate the SV for each $u_i \in U$:
3: **for all** $u_i \in U$ **do**
4: Calculate $\varphi_i = \frac{1}{2} \sum_{\substack{u_j \in U \\ j \neq i}} \mathcal{S}(\rho(u_i, u_j))$;
5: **end for**
6: Call Algorithm 1 to perform FCM on Shapley value;
7: Get the cluster labels and assign each user to the corresponding cluster;
8: Output the cluster to which each user belongs to obtain the final clustering result;

After calculating the Shapley value of each user, perform FCM on the Shapley value, assign each user to the corresponding cluster, and finally output the cluster to which each user belongs. Based on this, we use FCM and Shapley value to preprocess the data. The corresponding algorithm is shown in Algorithm 2.

Similarity computation

This step is mainly to compute the adjusted similarity proposed in this article. We first compute a conventional similarity between users in the target cluster. In collaborative filtering, items given to the target user will be selected according to the preferences of neighboring users. In order to identify such neighbors, one prevalent approach utilized in collaborative filtering is the application of similarity metrics. The most common measures to compute the similarity between users u_a and u_b are the Pearson correlation coefficient, cosine, and so on [42]. Due to the superiority of the Pearson correlation coefficient in collaborative filtering, we only introduce the calculation of the similarity between users u_a and u_b using the Pearson correlation coefficient in this paper as shown in Eq.(1). We know that the value range of

$\mathrm{sim}(u_a, u_b)$ in Eq. (1) is $[-1, 1]$, where -1 means completely dissimilar and 1 means completely similar. Next, we calculate the weight differences between users u_a and u_b using the formula below:

$$\mathrm{w}_d(u_a, u_b) = \sqrt{\frac{\sum_{i \in I_{a,b}} w_i \left(R_{a,i} - R_{b,i}\right)^2}{\sum_{i \in I_{a,b}} w_i}}, \tag{9}$$

where w_i is the weight of item i, defined as $w_i = \ln\left(1 + \frac{t}{n_i}\right)$, where t denotes the total number of items and n_i represents the number of times item i appears in all users' ratings. The purpose of adding weights is to give a greater weight to those relatively unpopular items, because these items appear less frequently in all user ratings. After weighing, the contribution of items with fewer ratings to the similarity will be suppressed. Items with more ratings will contribute more to the similarity, increasing the recommendation's accuracy.

With the analysis of the above weight difference formula, we propose a concept of similarity balance factor, which is specifically calculated as follows:

$$\mathrm{sim}_{BF}(u_a, u_b) = \tau(H)^{\mathrm{w}_d(u_a, u_b)}, \tag{10}$$

where $\tau(H)$ is the weight function of the similarity balance factor, and $\tau(H)$ can be calculated through:

$$\tau(H) = \frac{1}{\ln(2 + H)}, \tag{11}$$

where H represents the count of $I_{a,b}$, and $\tau(H)$ is within the interval (0, 1). The purpose of the similarity balance factor is to adjust the similarity measure based on the number of items evaluated together, thereby mitigating the issue of inaccurate similarity measures caused by data sparsity. The value of $\tau(H)$ gradually decreases with the increase of H, which is also to ensure the balance between the similarity measure and the number of items.

Next, we introduce our proposed adjusted Pearson correlation coefficient. It is important to highlight that our adjusted Pearson correlation coefficient is derived by augmenting the results of the traditional Pearson correlation coefficient with the inclusion of the similarity balance factor. Therefore, it can be gained by:

$$\mathrm{Adsim}(u_a, u_b) = \mathrm{sim}(u_a, u_b) \cdot \mathrm{sim}_{BF}(u_a, u_b). \tag{12}$$

Then, we calculate the similarity by using the adjusted Pearson correlation coefficient, that is, compute the adjusted similarity between the selected user and every other user in the cluster according to Eq. (12), the calculation of the above similarity can provide support for the selection of the neighbor set.

Table 2 An example of Rating matrix

	i_1	i_2	i_3	i_4	i_5
$\mathbf{u_1}$	5	–	4	2	–
$\mathbf{u_2}$	–	3	–	–	4
$\mathbf{u_3}$	–	–	4	3	–
$\mathbf{u_4}$	1	2	–	–	5

Table 2 shows a sparse rating matrix containing 4 users and 5 items. We can observe that items i_3 and i_4 are both rated by u_1 and u_3, then the similarity between u_1 and u_3 calculated by PCC is approximately 0.83, that is, $\mathrm{sim}(u_1, u_3) \approx 0.83$. Subsequently, the adjusted similarity between u_1 and u_3 is approximately 0.655 according to Eq. (12), namely, $\mathrm{Adsim}(u_1, u_3) \approx 0.655$, where the value of $\mathrm{sim}_{BF}(u_1, u_3)$ is determined by $\tau(H)$ and $\mathrm{w}_d(u_1, u_3)$. From the data in Table 2 and Eqs. (9) and (11), we can get the values of $\tau(H)$ and $\mathrm{w}_d(u_1, u_3)$ are approximately 0.721 and 0.632, respectively. Therefore, the value of $\mathrm{sim}_{BF}(u_1, u_3)$ is about 0.79. Obviously, from the sparsity of Table 2, it can be seen that the result of Eq. (12) is more reliable. Since the rating matrix in Table 2 is extremely sparse and u_3 only rates two items, the PCC result may overestimate the user similarity, while the adjusted Pearson correlation coefficient provides a more reasonable similarity. This shows that introducing the similarity balance factor can yield a more robust similarity measure.

Neighbor selection

Based on the above similarity, this step needs to choose a neighbor set in the target cluster for the target user, and this process ensures the realization of differential privacy. We next introduce the target user's quality function. Considering the target cluster $\mathcal{M}$, target users u_a and a specific set of users $N \subseteq \mathcal{M}/\{u_a\}$, so the quality function for choosing the set of neighbors of user u_a to be N is defined as follows:

$$q(\mathcal{M}, u, N) = \sum_{u_b \in N} \left|\mathrm{Adsim}(u_a, u_b)\right|. \tag{13}$$

For the quality function set above, we refer to the literatures [16, 17]. The quality function can be obtained by first calculating the absolute value of adjusted similarities $\mathrm{Adsim}(u_a, u_b)$, and then summing them up, which can be obtained from Eq. (13). Then, according to the concept of the exponential mechanism in Definition 2, we know that the probability that the set N is the neighbor set is obtained as follows:

$$Pr(N) = \frac{exp\left(\frac{\epsilon}{2\Delta q} q(\mathcal{M}, u, N)\right)}{\sum_{N' \in \mathfrak{N}} exp\left(\frac{\epsilon}{2\Delta q} q(\mathcal{M}, u, N')\right)}, \tag{14}$$

Algorithm 3 FSPPCFs/FPPCFs Algorithm

Require: M: user-item rating matrix; m: recommendation list length; u_a: target user; ϵ: privacy budget;
Ensure: results of Top-m recommendation;
1: **Step 1. Data Preprocessing.**
2: Execute Algorithm 1 or Algorithm 2 on matrix M to get clustering results $C_1, \cdots, C_k$, where k represents the number of clusters;
3: Select the target cluster through the target user's location to obtain the corresponding rating matrix $\mathcal{M} = C_i$, where $i = 1, \cdots, k$;
4: **Step 2. Similarity Computation.**
5: Compute the Adjusted similarities between u_a and $u_b \in \mathcal{M}$:
6: **for all** $u_b \in \mathcal{M}, u_a \neq u_b$ **do**
7: Calculate the similarity $\text{sim}(u_a, u_b)$ and similarity balance factor $\text{sim}_{BF}(u_a, u_b)$ between the target user and other users by Eqs. (1) and (10), respectively, and then compute the Adjusted similarities $\text{Adsim}(u_a, u_b)$ with Eq. (12);
8: **end for**
9: **Step 3. Neighbor selection.**
10: Compute the probability on the neighbor cluster $\mathfrak{N}$ of all neighbor sets of size N in the target cluster as follows:
11: **for all** $N \in \mathfrak{N}$ **do**
12: Calculate $Pr(N)$ by Eq. (13) and Eq. (14);
13: **end for**
14: Choose a neighbor set $N \in \mathfrak{N}$ of user u_a with the probability $Pr(N)$;
15: **Step 4. Top-m recommendation.**
16: Predict the scores of u_a on the unrated items by Eq. (16); and build a recommendation list for user u_a by choosing the Top-m item.

where Δq is defined to be the sensitivity of quality function q. In fact, the sensitivity of the user quality function is defined as follows:

$$\Delta q = \max_{N} \max_{\|\mathcal{M}_1 - \mathcal{M}_2\| \leq 1} |q(\mathcal{M}_1, u, N) - q(\mathcal{M}_2, u, N)| = 1, \tag{15}$$

where $\mathcal{M}_1$ and $\mathcal{M}_2$ are any pair of adjacent target cluster matrices.

After using Algorithm 1 or Algorithm 2 to obtain clusters, we take the cluster where the target user is located as the target cluster. From the target cluster, we choose a neighbor set of size N as a candidate neighbor set and repeat this process until all neighbor sets of size N in the target cluster are selected. By combining all the candidate neighbor sets, we form the final neighbor cluster $\mathfrak{N}$. Subsequently, we compute the probability over the neighbor cluster $\mathfrak{N}$. Finally, using the probability obtained above, we select a neighbor set.

Top-m recommendation

For the selected set N, construct a recommendation table for user u_a using the following formula:

$$pred_{u_a i} = \overline{r_a} + \frac{\sum_{u_b \in N} \text{Adsim}(u_a, u_b)\left(r_{b,i} - \overline{r_b}\right)}{\sum_{u_b \in N} |\text{Adsim}(u_a, u_b)|}, \tag{16}$$

where $\overline{r_a}$ and $\overline{r_b}$ are average ratings of users u_a and u_b, respectively. i represents the unrated items of u_a. Finally, all prediction scores are ranked in descending order, and the Top-m items are chosen to provide a list of recommendations for the target user u_a.

The above details the design process of FSPPCFs, and its corresponding algorithm design is illustrated in Algorithm 3. In addition, the privacy analysis of FSPPCFs is given below.

Privacy analysis

Theorem 1 *The proposed FSPPCFs algorithm satisfies ϵ-differential privacy.*

Proof For any two neighboring datasets M_1 and M_2 and any $N \in \mathfrak{N}$,

$$\frac{\exp\left(\frac{\varepsilon q(M_1,u,N)}{2\Delta q}\right)}{\exp\left(\frac{\varepsilon q(M_2,u,N)}{2\Delta q}\right)} = \exp\left(\frac{\epsilon\,(q(M_1,u,N) - q(M_2,u,N))}{2\Delta q}\right) \leq \exp(\frac{\epsilon}{2})$$

Similarly, in the same way, the following can be obtained:

$$\exp\left(\frac{\varepsilon q\left(M_2,u,N'\right)}{2\Delta q}\right) \leq \exp(\frac{\epsilon}{2}) \cdot \exp\left(\frac{\varepsilon q\left(M_1,u,N'\right)}{2\Delta q}\right)$$

we consider the ratio of the probability that each output $N \in \mathfrak{N}$ on two neighboring datasets M_1 and M_2 as follows:

$$\begin{aligned}
&\frac{\Pr[\mathcal{M}_q^{\epsilon}(M_1) = N]}{\Pr[\mathcal{M}_q^{\epsilon}(M_2) = N]} \\
&= \frac{\frac{\exp(\frac{\varepsilon q(M_1,u,N)}{2\Delta q})}{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}}{\frac{\exp(\frac{\varepsilon q(M_2,u,N)}{2\Delta q})}{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_2,u,N')}{2\Delta q})}} \\
&= \left(\frac{\exp(\frac{\varepsilon q(M_1,u,N)}{2\Delta q})}{\exp(\frac{\varepsilon q(M_2,u,N)}{2\Delta q})}\right) \cdot \left(\frac{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_2,u,N')}{2\Delta q})}{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}\right) \\
&\leq \exp(\frac{\epsilon}{2}) \cdot \left(\frac{\sum_{N' \in \mathfrak{N}} \exp(\frac{\epsilon}{2}) \exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}\right) \\
&\leq \exp(\frac{\epsilon}{2}) \cdot \exp(\frac{\epsilon}{2}) \cdot \left(\frac{\exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}{\sum_{N' \in \mathfrak{N}} \exp(\frac{\varepsilon q(M_1,u,N')}{2\Delta q})}\right) \\
&= \exp(\epsilon).
\end{aligned} \tag{17}$$

Therefore, we fully show that the FSPPCFs scheme achieves ϵ-differential privacy. □

Table 3 Information description of the datasets

Dataset	#User	#Item	#Rating	Scale	Sparsity
ML-100K	943	1682	100000	[1,5]	6.30%
FilmTrust	1508	2071	35497	[0.5,4]	1.14%

Experiment and evaluation

Experiment environment and dataset

All experiments in this paper were conducted on a single computer using the PyCharm development platform and Python 3.8 as the development language. The hardware configuration is Intel(R) core(TM): i5-11400F, the CPU and RAM are 2.60 GHz and 32.0 GB respectively, and the GPU is NVIDIA GeForce RTX 3080Ti.

We assessed the effectiveness of our proposed approach using the MovieLens 100K[1] dataset, which contains around 100,000 ratings of 1682 movies by 943 users in which there are five (1–5) grade ratings, and each user rates at least 20 items. The sparsity level s of MovieLens 100K is 6.30% ($s = \frac{\sharp R \times 100\%}{\sharp U \times \sharp M}$), so it is easy to see that the dataset is very sparse. To further verify the generalizability of our scheme, we conducted simulation experiments again using the FilmTrust[2] dataset. The FilmTrust dataset consists of 1508 users and 2071 items, with a total of 35,497 ratings and the sparsity level of 1.14%. Detailed dataset information is presented in Table 3.

Evaluation metrics

To evaluate the disparity between the actual and predicted ratings in the test set, we employed widely-used accuracy metrics for prediction, namely mean absolute error (MAE) and root mean square error (RMSE) by [24]. A lower error value indicates higher prediction accuracy. The formulas for MAE and RMSE are as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|A_i - P_i|, \tag{18}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|A_i - P_i|^2}, \tag{19}$$

where n represents the number of predicted items and A_i is the actual rating value in the testing set and P_i is the predicted rating of a target user on an item i.

[1] https://grouplens.org/datasets/movielens/

[2] https://librec.net/datasets.html

In this paper, meanwhile, four important metrics are used to measure the accuracy of recommendation models: Accuracy, Precision, Recall, and F1-value, respectively [18, 25]. These four formulas are described as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{20}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{21}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{22}$$

$$\text{F1-value} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{23}$$

In the evaluation, TP, FP, TN, and FN as in Eqs. (20)–(23) above can be described in a confusion matrix, as illustrated in Table 4.

Compared methods

We conducted comparative experiments using ten distinct approaches, as outlined below, to assess the performance of our presented scheme. To facilitate the description of the similarities and differences among these ten comparison methods, we have listed them in Table 5.

- **UCF:** A user-based collaborative filtering recommendation using cosine to calculate similarity.
- **EUCF:** An enhanced user-based collaborative filtering recommendation method addresses the sparsity of user ratings in the user similarity calculation and adjusts the similarity results by thoroughly considering the user's rating history.
- **HCF:** A hybrid recommendation approach that integrates both user-based and item-based collaborative filtering, as proposed by [49].
- **HUSMCF:** [50] proposed a hybrid user similarity model using KL divergence for collaborative filtering, which takes into account user preference factors and asymmetric factors to distinguish the rating preferences between different users.
- **KPPCF:** A scheme was presented by [16], employing the KM algorithm and differential privacy, referred to as KDPCF. Unlike KDPCF, KPPCF endeavors to replicate the results of the scheme proposed by [16] as closely as possible, although minor differences may persist. Hence, this approach is designated as KPPCF in the present study.
- **PPCFs:** In this scheme, differential privacy and the similarity balance factor are applied in collaborative filtering recommendations without any data preprocessing.
- **FPPCFs/FPPCF:** To reasonably compare KPPCFs and PPCFs, we design FPPCFs scheme, which combines

Table 4 Confusion matrix

		Correct result Positive	Negative
Obtained result	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 5 Similarities and differences among several algorithms

Algorithms	Clustering method	Similarity balance factor	Privacy protection
UCF	✗	✗	✗
EUCF	✗	✗	✗
HCF	✗	✗	✗
HUSMCF	✗	✗	✗
PPCFs	✗	✓	✓
KPPCF	K-means	✗	✓
FSPPCF	FCM, SV	✗	✓
FPPCF	FCM	✗	✓
FPPCFs	FCM	✓	✓
FSPPCFs	FCM, SV	✓	✓

FCM and adjusted similarity algorithms. Unlike FPPCFs, the similarity balance factor is not used in FPPCF.

- **FSPPCFs/FSPPCF:** FSPPCF, a privacy protection collaborative filtering scheme based on FCM and Shapley value in the paper, is distinct from its counterpart, FSPPCFs, in that it eliminates the usage of the similarity balance factor.

Experimental result and analysis

The experiment uses the dataset MovieLens 100K for comparison experiments with the other schemes. The dataset is separated into two parts: the training set and the test set with a ratio of 4:1 in the experiment. Then, the metrics proposed in Sect. "Evaluation metrics" are used to conduct experiments to evaluate the algorithm proposed in this paper. The key parameters involved in the experiment are presented in Table 6. In addition, we also verified the effectiveness of introducing similarity balance factor in FSPPCFs on the dataset FilmTrust.

Table 6 Setting of key parameters

Parameter	Value	Meaning
C	2	number of clusters
ξ	1e-5	threshold value
m	2	fuzzy index
ϵ	[0.1, 1]	privacy budget
Top-m	20	recommended list length
N	[10, 60]	numbers of neighbors
τ	0.91, 0.51, 0.25, 0.16	weight function values

In Figs. 3 and 4, we consider the influence of different numbers of neighbors on recommendation results. We set the number of neighbors to 10, 20, 30, 40, 50, and 60 in

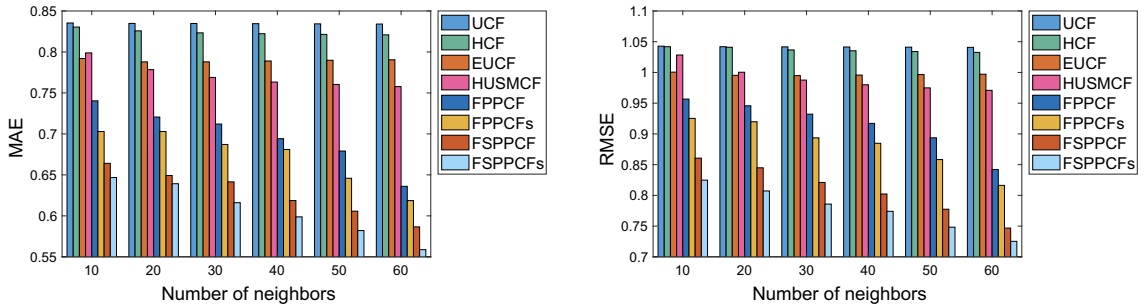

Fig. 3 The MAE and RMSE comparison with different numbers of neighbors on MovieLens 100K

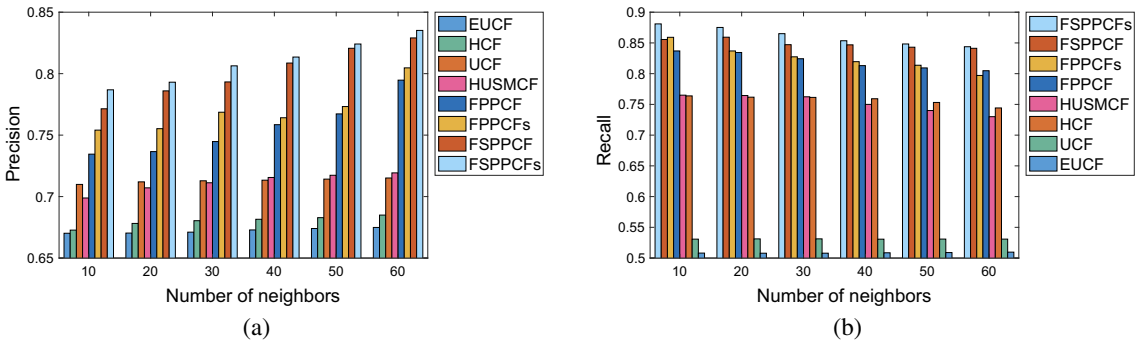

Fig. 4 The Precision and Recall comparison with different numbers of neighbors on MovieLens 100K

Table 7 Comparison results of F1-value of different algorithms on the dataset MovieLens 100K with different numbers of neighbors

Algorithm	No. of Neighbors					
	10	20	30	40	50	60
UCF	0.56649	0.56563	0.56567	0.56594	0.56635	0.56611
EUCF	0.57798	0.57792	0.57822	0.57930	0.57997	0.58074
HCF	0.70650	0.71753	0.71861	0.71828	0.71629	0.71335
HUSMCF	0.73044	0.73464	0.73591	0.73698	0.73684	0.73727
FPPCF	0.78239	0.78247	0.78251	0.78484	0.78774	0.79971
FPPCFs	0.80318	0.79399	0.79450	0.79325	0.79295	0.80088
FSPPCF	0.81133	0.82101	0.81931	0.82731	0.83169	0.83511
FSPPCFs	0.83124	0.83210	0.83468	0.83304	0.83603	0.83947

sequence and compare recommendation performance among several schemes with different numbers of neighbors.

It can be seen from Fig. 3 that when the neighbors' number increases, the MAE and RMSE values will decrease, which means that the recommendation performance will become better. The recommendation quality of several classic collaborative filtering schemes (UCF, HCF, EUCF, and HUSMCF) is relatively poor, but the use of FCM relatively improves the prediction accuracy. FPPCFs, in particular, achieve good results in terms of prediction accuracy due to the addition of the similarity balancing factor. Additionally, it is evident that FSPPCFs outperform other schemes in terms of performance. The primary reason is that FSPPCFs introduces a similarity balance factor to the traditional similarity algorithm and incorporates Shapley values to account for the user's intrinsic attributes, thereby enhancing recommendation performance. Compared to HUSMCF and FPPCFs, FSPPCFs improves accuracy by at least 19% and 7%, respectively.

As illustrated in Fig. 4, however, we see that the precision and recall show opposite trends when neighbors' numbers increase. We can clearly find that the precision increases with the number of neighbors and the recall decreases with the number of neighbors. This is normal for us as we always want the precision of recommendations to be higher. At the same time, we can also see that FSPPCFs has an overwhelming advantage over other schemes in recommendation performance. Compared to HUSMCF and FPPCFs, the performance of FSPPCFs is enhanced by at least 11% and 4%, respectively.

Table 7 further presents the comparison results of the F1-value of the four algorithms under different numbers of neighbors. The data results demonstrate that FSPPCFs gains better performance than other schemes under different numbers of neighbors. When the number of neighbors is 60, the F1-value of FSPPCF will be close to 0.84, while the F1-value of other schemes (UCF, HCF, EUCF, and HUSMCF) are basically less than 0.75. Several other schemes using FCM also improve the prediction accuracy, but also not as good as FSPPCFs.

To further verify the generalizability of FSPPCFs, we conducted comparative experiments using the FilmTrust dataset, as illustrated in Fig. 5. In the experiment, we used FSPPCF as the control group to test the effect of the similarity balance factor on the prediction accuracy. At the same time, we listed the MAE and RMSE values of FSPPCFs and FSPPCF on ML-100K, as shown in Fig. 6. The experimental results show that the MAE and RMSE values of FSPPCFs are lower than those of FSPPCF, indicating that the prediction performance of FSPPCFs is better than that of FSPPCF. Specifically, on the ML-100K dataset, the MAE and RMSE for FSPPCFs improved by 4.73% and 4.46%, respectively, while on the FilmTrust dataset, the MAE and RMSE improved by 4.18% and 2.33%, respectively.

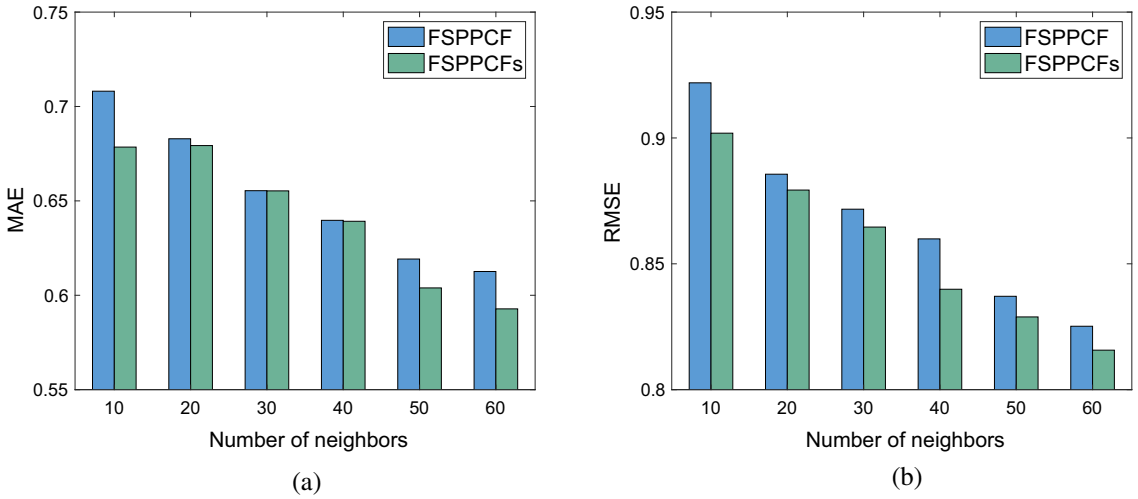

Fig. 5 The MAE and RMSE comparison with different numbers of neighbors on FilmTrust

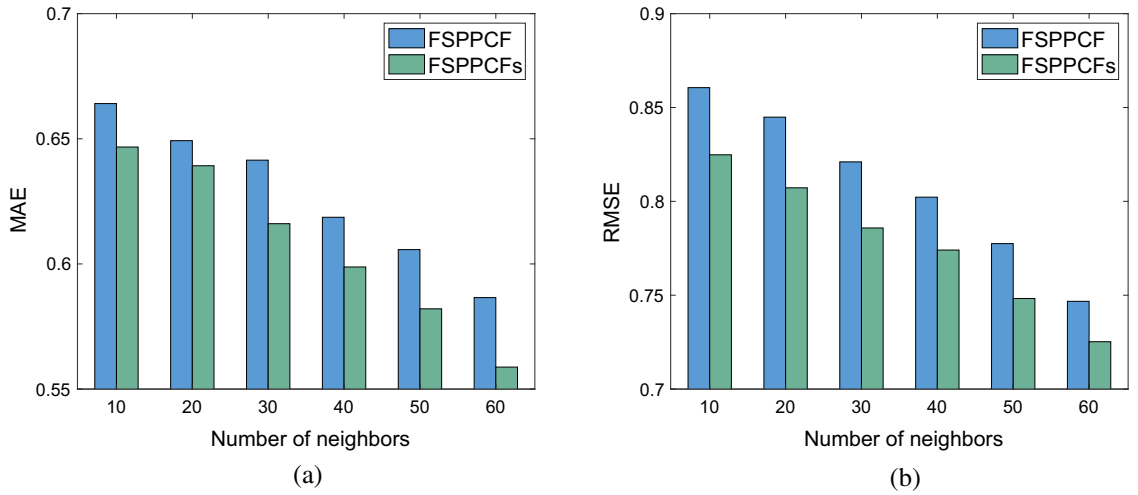

Fig. 6 The MAE and RMSE comparison with different numbers of neighbors on MovieLens 100K

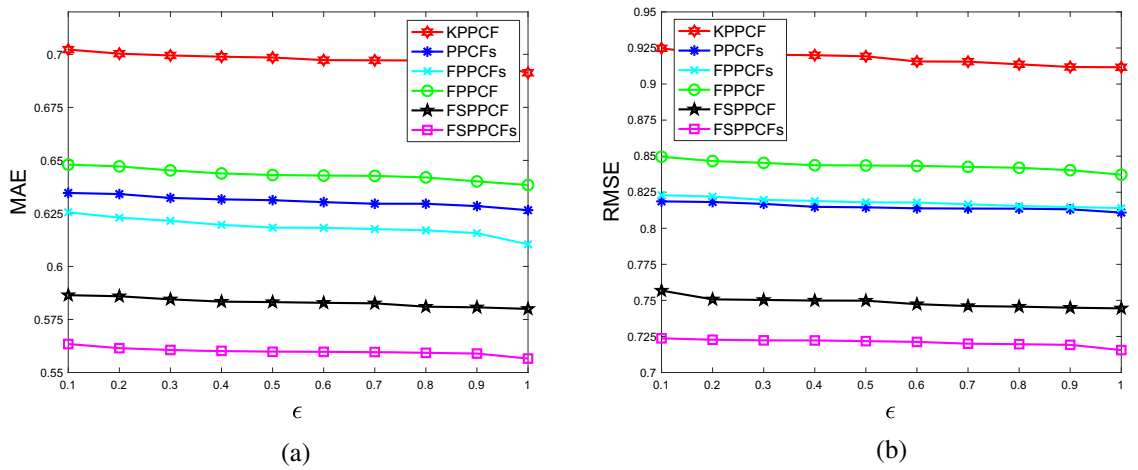

Fig. 7 The MAE and RMSE comparison with different privacy budgets on MovieLens 100K

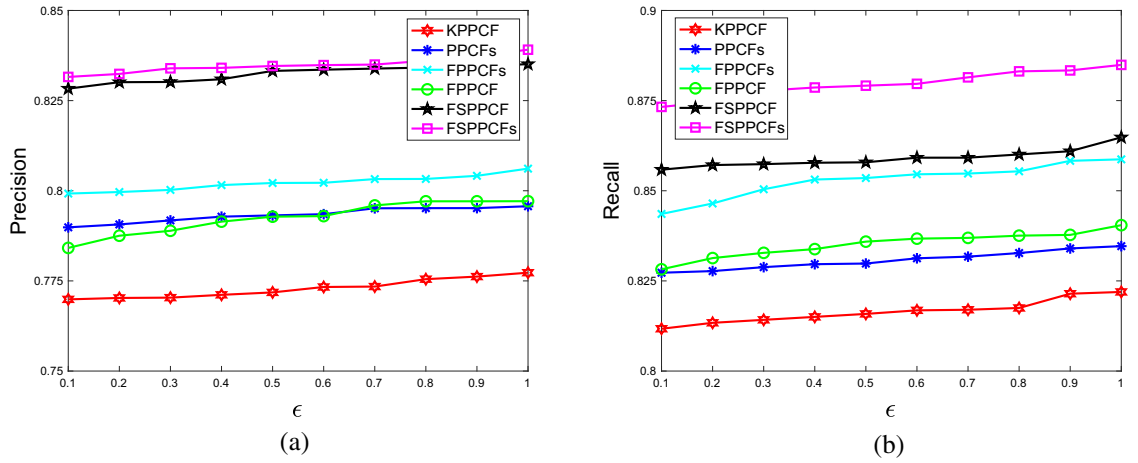

Fig. 8 The Precision and Recall comparison with different privacy budgets on MovieLens 100K

In Fig. 7, we investigate the influence of different privacy budgets on recommendation performance. We set the privacy budget ϵ at intervals of 0.1 on the range [0, 1] and compare the MAE, RMSE, precision, and recall among the six schemes. As illustrated in Fig. 7, we observe that as the privacy budget increases, the MAE and RMSE of the six schemes steadily decrease and exhibit a relatively flat trend. This can be attributed to the fact that these schemes do not directly introduce Laplace noise but utilize the exponential mechanism to introduce randomness in the neighbor set selection for the target user. The use of the exponential mechanism contrasts with the introduction of Laplace noise, which significantly impairs data availability. However, the exponential mechanism is not highly sensitive to the privacy budget value, allowing for accurate prediction results even when the privacy budget is relatively small. Compared with KPPCF and FPPCFs, the prediction accuracy of FSPPCFs is improved by approximately 20% and 9% respectively when $\epsilon = 0.1$.

In Fig. 8, we can observe that as the privacy budget ϵ is increased, so do precision and recall. This is due to the fact that increasing the privacy budget reduces noise and weakens privacy protection, which improves recommendation performance. Furthermore, our proposed scheme outperforms several other schemes in recommendation performance. It is even possible to see that the performance of PPCFs is superior to that of KPPCF. The main reason for this is that the similarity balance factor is added to PPCFs. Due to the assistance of the similarity balance factor, the performance of FSPPCFs and FPPCFs is better than that of FSPPCF and FPPCF, respectively. From a data perspective, FSPPCFs improve recommendation accuracy by 0.3%, 3.9%, and 5.7%, respectively, compared to FSPPCF, FPPCFs, and FPPCF. Compared with KPPCF, FSPPCFs improves the recommendation accuracy by about 7.5%. FSPPCFs utilize FCM and Shapley values for data preprocessing, thereby enhancing data utilization and recommendation performance. It is evident that the proposed scheme outperforms others in recommendation performance.

To further illustrate the performance of the proposed scheme, Table 8 lists the comparison between the F1-value of each algorithm under different privacy budgets. It can also be seen that FSPPCFs achieve better performance than other schemes. In fact, according to Definition 1, the larger the value of the privacy budget ϵ, the lower the level of privacy protection and the better the recommendation performance, and vice versa. Through the above analysis, our scheme can fully realize the compromise between the user's personal privacy and recommendation performance.

To better demonstrate the effectiveness of the similarity balance factor in the scheme FSPPCFs in improving the recommendation performance. We choose four values for the weight function: 0.91, 0.51, 0.25, and 0.16, and we also consider how the recommendation performance of FSPPCFs with different weight function values varies under different numbers of neighbors. We know that the weight function $\tau(H)$ will become smaller as H increases, so we consider observing the change of recommendation performance of FSPPCFs when H increases. In Figs. 9 and 10, we compare the variation of Precision, recall, and accuracy values with different weight functions under different numbers of neighbors when the privacy budget $\epsilon = 0.1$ and $\epsilon = 0.5$ respectively.

As illustrated in Figs. 9a, b and 10a, b, for different numbers of neighbors, the precision and recall show opposite trends. In Figs. 9(a) and 10(a), we see that the precision increases with the constant number of neighbors, and we can also see that the smaller the τ value, the greater the precision. However, the recall will decrease with the constant number of neighbors. At the same time, it can also be seen that the smaller the value of τ, the greater the recall, as illustrated in Figs. 9(b) and 10(b).

Table 8 F1-value comparison results of different algorithms on the dataset MovieLens 100K

Algorithm	Privacy budget ϵ									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
KPPCF	0.7902	0.7912	0.7917	0.7925	0.7932	0.7945	0.7946	0.7959	0.7982	0.7990
PPCFs	0.8081	0.8088	0.8099	0.8108	0.8111	0.8119	0.8130	0.8135	0.8141	0.8147
FPPCFs	0.8208	0.8224	0.8246	0.8265	0.8270	0.8276	0.8282	0.8285	0.8303	0.8337
FPPCF	0.8056	0.8088	0.8102	0.8121	0.8138	0.8143	0.8159	0.8168	0.8169	0.8182
FSPPCF	0.8419	0.8434	0.8436	0.8441	0.8454	0.8462	0.8463	0.8469	0.8475	0.8497
FSPPCFs	0.8519	0.8530	0.8553	0.8558	0.8563	0.8567	0.8576	0.8589	0.8598	0.8614

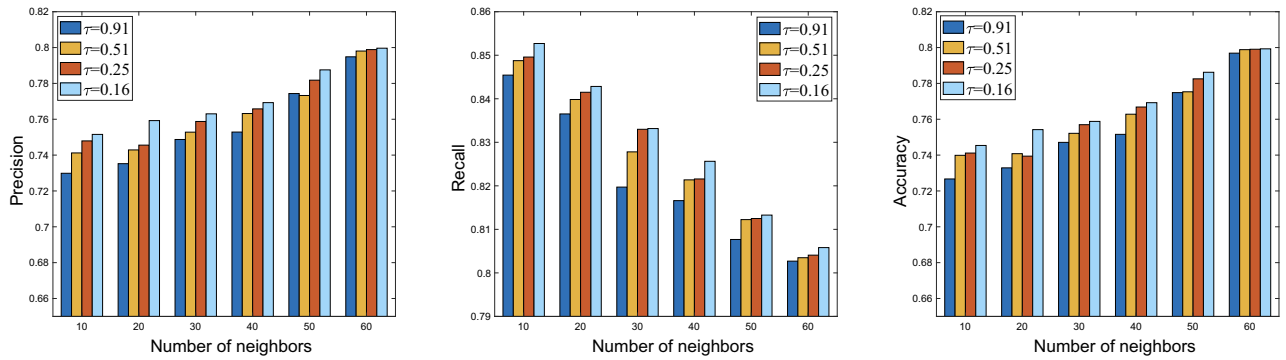

Fig. 9 Performance comparison of Precision, Recall, and Accuracy with different numbers of neighbors ($\epsilon = 0.1$)

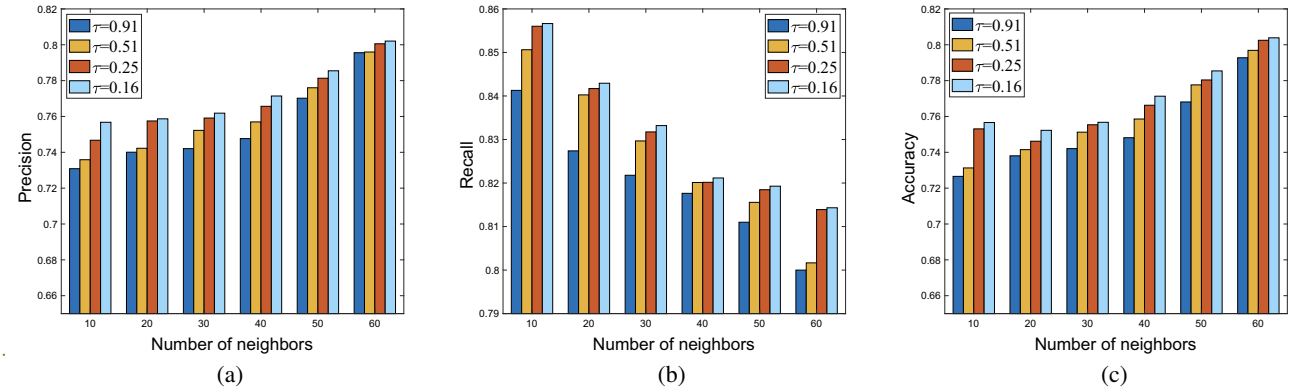

Fig. 10 Performance comparison of Precision, Recall, and Accuracy with different numbers of neighbors ($\epsilon = 0.5$)

In Figs. 9c and 10c, we can find that accuracy will become larger with the constant number of neighbors. Meanwhile, when the τ value is smaller, the accuracy will be greater. The main reason for this is that the smaller the τ value, the larger the H value, which will ensure the balance between the similarity measure and the number of items, thereby avoiding the problem of inaccurate similarity measure due to data sparseness. Ultimately, the recommendation performance will be further improved.

In addition, in Table 9, we present the F1-value of various weight functions in the proposed schemes for different numbers of neighbors, considering privacy budgets $\epsilon = 0.1$ and $\epsilon = 0.5$. The results demonstrate a gradual increase in the F1-value as the weight function decreases. This indicates that a lower weight function contributes to improved F1-value, implying better performance in terms of precision and recall.

Challenges

Based on the above experimental results analysis, FSPPCFs has played an important role in promoting the development of recommendation systems. However, there are still three challenges that need to be emphasized in actual application scenarios: (1) In real-world scenarios, user behavior

Table 9 Comparison of F1-value of different weight functions with different numbers of neighbors when the privacy budget is 0.1 and 0.5 respectively

Privacy budget	Weight function $\tau(H)$	No. of Neighbors					
		10	20	30	40	50	60
$\epsilon = 0.1$	$\tau = 0.91$	0.78341	0.78260	0.78261	0.78343	0.79066	0.79857
	$\tau = 0.51$	0.79135	0.78839	0.78852	0.79122	0.79227	0.80076
	$\tau = 0.25$	0.79418	0.79195	0.79415	0.79270	0.79685	0.80146
	$\tau = 0.16$	0.79893	0.79886	0.79655	0.79646	0.80021	0.80272
$\epsilon = 0.5$	$\tau = 0.91$	0.78219	0.78129	0.77988	0.78107	0.79006	0.79778
	$\tau = 0.51$	0.78909	0.78822	0.78906	0.78728	0.79529	0.79880
	$\tau = 0.25$	0.79765	0.79737	0.79377	0.79198	0.79947	0.80717
	$\tau = 0.16$	0.80361	0.79859	0.79592	0.79551	0.80202	0.80814

may change significantly. Therefore, ensuring that the recommendation model can still provide accurate and stable recommendations in the face of these dynamic changes is an important and challenging issue; (2) In practical applications, how to integrate this privacy-preserving recommendation mechanism into existing systems and maintain good scalability to cope with large-scale data and user needs is an issue that needs to be focused on during design; (3) In a large-scale data environment, the system needs to maintain efficient recommendation performance and processing capabilities. Designing a scalable system with real-time responsiveness is an important challenge.

Conclusion and future works

In this paper, we primarily investigate the compromise problem between users' personal privacy concerns and system performance in recommendation systems. We first designed an algorithm that integrates FCM and Shapley value, which is used to preprocess historical data, aiming to effectively solve the problem of recommendation performance degradation caused by privacy protection. Then, we introduced the concepts of weight difference and weight function between users to derive a method for calculating the similarity balance factor, and used it to improve the traditional Pearson similarity algorithm to improve the recommendation performance. Finally, we use Shapley value to fully explore the intrinsic relationship between users and present a novel privacy-preserving user-based collaborative filtering recommendation scheme utilizing FCM and Shapley value, FSPPCFs, to enable a better compromise between user privacy and recommendation performance. Experimental simulation results demonstrate that the proposed scheme is superior to other schemes, especially the significant difference between whether there is the similarity balance factor or not.

In the future, our work intends to study the offensive and defensive problems in recommendation systems. The recommendation system may encounter model inversion attacks or data tampering attacks, which may affect the system's performance when facing malicious users. Therefore, enhancing the system's defense capabilities against malicious attacks is also an important direction for future research. Furthermore, we intend to improve the security and reliability of the system by designing a more robust security mechanism by combining anomaly detection technology and evolutionary game theory. At the same time, we will optimize and innovate the recommendation algorithm by combining cutting-edge technologies such as deep learning and federated learning to improve the performance of the overall system.

Author Contributions Weiwei Wang: Conceptualization, Methodology, Software, Writing-original draft, Writing-review & editing. Wenping Ma: Funding acquisition, Supervision. Kun Yan: Software, Validation.

Funding This work was supported by the Key Industry Innovation Chain Project of Shaanxi Provincial Science and Technology Department, China, under Grant 2022ZDLGY03-08.

Data availability The authors confirm that the data supporting the findings of this article are available in the associated links within the paper.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Patra BK, Launonen R, Ollikainen V, Nandi S (2015) A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowl-Based Syst 82:163–177. https://doi.org/10.1016/j.knosys.2015.03.001
2. Feng S, Meng J, Zhang J (2021) News recommendation systems in the era of information overload. J Web Eng. https://doi.org/10.13052/jwe1540-9589.20210
3. Lü L, Medo M, Yeung CH, Zhang Y-C, Zhang Z-K, Zhou T (2012) Recommender systems. Phys Rep 519(1):1–49. https://doi.org/10.1016/j.physrep.2012.02.006
4. Cui Z, Xu X, Fei X, Cai X, Cao Y, Zhang W, Chen J (2020) Personalized recommendation system based on collaborative filtering for iot scenarios. IEEE Trans Serv Comput 13(4):685–695. https://doi.org/10.1109/TSC.2020.2964552
5. Zhang Y, Yin C, Wu Q, He Q, Zhu H (2019) Location-aware deep collaborative filtering for service recommendation. IEEE Trans Syst Man Cybern Syst 51(6):3796–3807. https://doi.org/10.1109/TSMC.2019.2931723
6. Zhang Q, Lu J, Jin Y (2021) Artificial intelligence in recommender systems. Complex Intell Syst 7:439–457. https://doi.org/10.1007/s40747-020-00212-w
7. Huang Q, Zeng Y (2024) Improving academic performance predictions with dual graph neural networks. Complex Intelli Syst. https://doi.org/10.1007/s40747-024-01344-z
8. Bhatia V (2024) Dlsf: deep learning and semantic fusion based recommendation system. Expert Syst Appl 250:123900. https://doi.org/10.1016/j.eswa.2024.123900
9. Li N, Xia Y (2024) Movie recommendation based on als collaborative filtering recommendation algorithm with deep learning model. Entertain Comput 51:100715. https://doi.org/10.1016/j.entcom.2024.100715
10. Fu M, Qu H, Yi Z, Lu L, Liu Y (2018) A novel deep learning-based collaborative filtering model for recommendation system. IEEE Trans Cybern 49(3):1084–1096. https://doi.org/10.1109/TCYB.2018.2795041
11. Alenizi J, Alrashdi I (2023) Sfmr-sh: secure framework for mitigating ransomware attacks in smart healthcare using blockchain technology. Sustain Mach Intell J 2:1–4. https://doi.org/10.61185/SMIJ.2023.22104
12. Ismail M, Abd El-Gawad AF (2023) Revisiting zero-trust security for internet of things. Sustain Mach Intell J 3:1–6. https://doi.org/10.61185/SMIJ.2023.33106
13. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theoret Comput Sci 9(3–4):211–407. https://doi.org/10.1561/0400000042
14. Guo T, Luo J, Dong K, Yang M (2019) Locally differentially private item-based collaborative filtering. Inf Sci 502:229–246. https://doi.org/10.1016/j.ins.2019.06.021
15. Guo T, Peng S, Li Y, Zhou M, Truong T-K (2023) Community-based social recommendation under local differential privacy protection. Inform Sci. https://doi.org/10.1016/j.ins.2023.119002
16. Chen Z, Wang Y, Zhang S, Zhong H, Chen L (2021) Differentially private user-based collaborative filtering recommendation based on k-means clustering. Expert Syst Appl 168:114366. https://doi.org/10.1016/j.eswa.2020.114366
17. Zhu X, Sun Y (2013) Differential Privacy for Collaborative Filtering Recommender Algorithm. In: Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics. IWSPA '16, pp. 9–16. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/2875475.2875483
18. Koohi H, Kiani K (2016) User based collaborative filtering using fuzzy c-means. Measurement 91:134–139. https://doi.org/10.1016/j.measurement.2016.05.058
19. Zhong Y, Huang C, Li Q (2022) A collaborative filtering recommendation algorithm based on fuzzy c-means clustering. J Intell Fuzzy Syst 43(1):309–323. https://doi.org/10.3233/JIFS-212216
20. Duan L, Wang W, Han B (2021) A hybrid recommendation system based on fuzzy c-means clustering and supervised learning. Korean Soc Internet Inform (KSII) 15:2399–2413. https://doi.org/10.3837/tiis.2021.07.006
21. Liu J, Kang X, Nishide S, Ren F (2020) Collaborative Filtering Recommendation Algorithm Based on Bisecting K-means Clustering. In: International Symposium on Artificial Intelligence and Robotics 2020, vol. 11574, pp. 311–318 . https://doi.org/10.1117/12.2580026. SPIE
22. Chen L, Luo Y, Liu X, Wang W, Ni M (2021) Improved collaborative filtering recommendation algorithm based on user attributes and k-means clustering algorithm. J Phys Conf Ser 1903:012036. https://doi.org/10.1088/1742-6596/1903/1/012036
23. Zarzour H, Maazouzi F, Al-Zinati M, Nusayr A, Alsmirat M, Al-Ayyoub M, Jararweh Y (2022) Using k-means clustering ensemble to improve the performance in recommender systems. In: 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), pp. 176–180 . https://doi.org/10.1109/IDSTA55301.2022.9923070. IEEE
24. Deng J, Guo J, Wang Y (2019) A novel k-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering. Knowl-Based Syst 175:96–106. https://doi.org/10.1016/j.knosys.2019.03.009
25. Chen J, Zhao C, Chen L (2020) Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. Complex Intell Syst 6(1):147–156. https://doi.org/10.1007/s40747-019-00123-5
26. 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. https://doi.org/10.1007/s00521-018-3959-2
27. Li M, Wen L, Chen F (2021) A novel collaborative filtering recommendation approach based on soft co-clustering. Phys A 561:125140. https://doi.org/10.1016/j.physa.2020.125140
28. Jumonji S, Sakai K, Sun M-T, Ku W-S (2023) Privacy-preserving collaborative filtering using fully homomorphic encryption. IEEE Trans Knowl Data Eng 35(3):2961–2974. https://doi.org/10.1109/TKDE.2021.3115776
29. Kim J, Koo D, Kim Y, Yoon H, Shin J, Kim S (2018) Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption. ACM Trans Privacy Secur (TOPS) 21(4):1–30. https://doi.org/10.1145/3212509
30. Zhou J, Gao G, Cao Z, Choo K-KR, Dong X (2023) Lightweight privacy-preserving distributed recommender system using tag-based multikey fully homomorphic data encapsulation. IEEE Trans Dependable Secure Comput. https://doi.org/10.1109/TDSC.2023.3243598
31. Ren H, Xu G, Zhang T, Ning J, Huang X, Li H, Lu R (2022) Efficiency boosting of secure cross-platform recommender systems over sparse data. arXiv preprint arXiv:2212.01537
32. Zhu T, Ren Y, Zhou W, Rong J, Xiong P (2014) An effective privacy preserving algorithm for neighborhood-based collaborative filtering. Futur Gener Comput Syst 36:142–155. https://doi.org/10.1016/j.future.2013.07.019
33. Xiong P, Zhang L, Zhu T, Li G, Zhou W (2020) Private collaborative filtering under untrusted recommender server. Futur Gener Comput Syst 109:511–520. https://doi.org/10.1016/j.future.2018.05.077

34. Chronis C, Varlamis I, Himeur Y, Sayed AN, Al-Hasan TM, Nhlabatsi A, Bensaali F, Dimitrakopoulos G (2024) A survey on the use of federated learning in privacy-preserving recommender systems. IEEE Open J Comput Soc. https://doi.org/10.1109/OJCS.2024.3396344
35. Feng C, Feng D, Huang G, Liu Z, Wang Z, Xia X-G (2024) Robust privacy-preserving recommendation systems driven by multimodal federated learning. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2024.3411402
36. Harasic M, Keese F-S, Mattern D, Paschke A (2024) Recent advances and future challenges in federated recommender systems. Int J Data Sci Anal 17(4):337–357. https://doi.org/10.1007/s41060-023-00442-4
37. Metwaly AA, Elhenawy I (2023) Protecting iot devices from botnet threats: a federated machine learning solution. Sustain Mach Intell J 2:1–5. https://doi.org/10.61185/SMIJ.2023.22105
38. Metwaly AA, Elhenawy I (2023) Sustainable intrusion detection in vehicular controller area networks using machine intelligence paradigm. Sustain Mach Intell J 4:1–4. https://doi.org/10.61185/SMIJ.2023.44104
39. Walli SA, Sallam K (2024) Machine learning for intrusion detection: a reproducible baseline is all you need. Sustain Mach Intell J 7:1–3. https://doi.org/10.61356/SMIJ.2024.77103
40. Yan K, Ma W, Sun S (2024) Communications and networks resources sharing in 6g: challenges, architecture, and opportunities. IEEE Wirel Commun. https://doi.org/10.1109/MWC.003.2400038
41. Koohi H, Kiani K (2017) A new method to find neighbor users that improves the performance of collaborative filtering. Expert Syst Appl 83:30–39. https://doi.org/10.1016/j.eswa.2017.04.027
42. Fkih F (2022) Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison. J King Saud Univ-Comput Inform Sci 34(9):7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014
43. Bezdek JC, Ehrlich R, Full W (1984) Fcm: the fuzzy c-means clustering algorithm. Comput Geosci 10(2):191–203. https://doi.org/10.1016/0098-3004(84)90020-7
44. Askari S (2021) Fuzzy c-means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. Expert Syst Appl 165:113856. https://doi.org/10.1016/j.eswa.2020.113856
45. Li N, Lyu M, Su D, Yang W (2016) Differential privacy: from theory to practice. Synth Lect Inform Secur Privacy Trust 8(4):1–138. https://doi.org/10.1007/978-3-031-02350-7
46. Roger BM, et al. (1991) Game theory: analysis of conflict. The President and Fellows of Harvard College, USA **66**
47. Garg VK, Narahari Y, Murty MN (2012) Novel biobjective clustering (bigc) based on cooperative game theory. IEEE Trans Knowl Data Eng 25(5):1070–1082. https://doi.org/10.1109/TKDE.2012.73
48. Genther H, Runkler TA, Glesner M (1994) Defuzzification based on fuzzy clustering. Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference 3, 1645–1648 . https://doi.org/10.1109/FUZZY.1994.343943
49. Ji H, Li J, Ren C, He M (2013) Hybrid collaborative filtering model for improved recommendation. In: Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 142–145 . https://doi.org/10.1109/SOLI.2013.6611398. IEEE
50. Wang Y, Deng J, Gao J, Zhang P (2017) A hybrid user similarity model for collaborative filtering. Inf Sci 418:102–118. https://doi.org/10.1016/j.ins.2017.08.008