IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 May 2024, accepted 27 May 2024, date of publication 5 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3409752

RESEARCH ARTICLE

Personalized Book Intelligent Recommendation System Design for University Libraries Based on IBCF Algorithm

NA LIN
Jilin Agricultural University, Changchun 130118, China
e-mail: linna@jlau.edu.cn

ABSTRACT With the digital transformation and improvement of university library information technology, readers' demands for library services are increasingly diversified and personalized. They are no longer satisfied with the traditional borrowing services, but hope that the library can provide more accurate and personalized recommendation services. To solve these problems, this study first proposes an improved item-based collaborative filtering recommendation algorithm based on the mean model representation. Then, combining this algorithm with user-based collaborative filtering recommendation algorithm, an improved item-based collaborative filtering algorithm is designed. The results showed that the CPU usage of the whole system was not high during the operation of the improved item-based collaborative filtering recommendation algorithm, with an average usage rate of about 9.8%. The minimum root mean square error of the algorithm was 0.013 and the runtime was 12000 μs. Compared with existing the similar systems, when the number of users exceeded 200, the response speed was significantly reduced by more than 50%, and the coverage rate reached more than 90%. In summary, the personalized intelligent book recommendation system for university library proposed in the study has the advantages of high coverage, low resource consumption, high accuracy and so on, which can provide readers with more accurate and personalized recommendation services.

INDEX TERMS SIBCF algorithm, UBCF, collaborative filtering, book recommendations, mean model vector representation.

I. INTRODUCTION

Against the backdrop of the rapid development of information technology, university libraries are facing enormous challenges and opportunities [1]. With the continuous increase of digital book resources, readers often feel confused and find it difficult to find the books they are truly interested in when facing a huge library collection. To better meet the needs of readers and improve the utilization and satisfaction of book resources, the research on personalized book intelligent recommendation systems has become an urgent problem that needs to be solved. At present, some studies have adopted Collaborative Filtering (CF) algorithms to implement book recommendation systems [2], [3]. However, traditional CF algorithms have some problems, such as low accuracy caused by long items and sparse data. When new users join the system, they cannot accurately recommend books that are suitable for users' interests due to the lack of historical behavioral data. In addition, as the number of users and books increases, traditional CF algorithms will face huge computational complexity when calculating the similarity matrix, leading to a decrease in the scalability of the algorithm. This limits the application of algorithms in large-scale library systems. To address these issues, a study is conducted to represent item vectors using a mean square model, fill sparse data using clustering-based filling methods, and then combine the User-Based Collaborative Filtering (UBCF) algorithm to design a new personalized book intelligent recommendation system for university libraries based on Improved Item-Based Collaborative Filtering

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

(IIBCF). It is expected that the personalized intelligent book recommendation system proposed in the research of university libraries can provide readers with more accurate and personalized book recommendation services.

The personalized book intelligent recommendation system proposed in this article is based on the IBCF algorithm and combined with the UBCF algorithm. Through a series of innovative measures such as clustering based filling methods, the problem of data sparsity has been effectively solved, and recommendation efficiency and user satisfaction have been improved. The system adopts the mean model representation method, which shortens the length of the project vector, reduces computation time, and improves computational efficiency. Meanwhile, by integrating the advantages of IBCF and UBCF algorithms, the system can comprehensively understand user interests and behavior patterns, provide more personalized and accurate book recommendation services, and meet the growing personalized reading needs of users. In addition, the system also implements various recommendation functions such as new book recommendation, popular book recommendation, and personalized recommendation, significantly improving the operational efficiency of the library and providing users with a more intelligent and personalized service experience. The application of these innovative methods and technologies gives the system a certain degree of leadership and practicality in the field of personalized book recommendation.

The personalized book recommendation system based on the IIBCF algorithm proposed in this study has made significant contributions in terms of recommendation accuracy and efficiency. By introducing the mean model representation method, the length of the item vector is effectively shortened, thereby reducing computation time and improving recommendation efficiency. In addition, combined with the UBCF algorithm, the clustering filling method was used to solve the problem of data sparsity, further improving recommendation accuracy and user satisfaction. Previous recommendation algorithms often encountered difficulties in dealing with sparse data and cold start problems, but the method proposed in this study effectively solves these problems through mean model representation and cluster filling techniques, enabling the system to provide accurate recommendation results even in sparse data. Meanwhile, this method can also address the issues of new users and projects, improving the coverage and accuracy of recommendation systems.

This manuscript consists of six main sections. Section I is the introduction of this paper, and Section II discusses the related work in the field of network security. Section III establishes a malicious domain name server detection technology based on the transformer convolutional neural network algorithm. Section IV compares the algorithm performance of the model and analyzes the applicability of the model. Section V summarizes the research results and future prospects of the full text. Section VI is a discussion of the study of the full text.

II. RELATED WORKS

In recent years, with the rapid popularization of artificial intelligence technology, people have begun to focus on the research of accurate recommendation systems using artificial intelligence technology. Kataria et al. proposed a method using Nearest Common Collaborative Filtering (NCCF) for data sparsity in book recommendation systems and used a novel similarity index to match users and items. Results showed that this method significantly improved recommendation accuracy, recall rate, and F1-score, and couldreveal the duality between users and items [4]. Kumar proposed an effective consulting tool using Fuzzy Rough Set Theory-based Collaborative Filtering (FRSTBCF)to address the problem of data sparsity and large prediction errors in recommendation systems. The results showed that the FRSTBCF algorithm outperformed traditional algorithms in various evaluation indicators, such as Mean Average Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE) [5]. Kumar et al. proposed an aggregation technique using the Stacked Integration Model (STEM) to address the issue of providing appropriate suggestions for aggregating group member preferences in group recommendation systems. In the first stage, k-Nearest Neighbor (k-NN), Singular Value Decomposition (SVD), UBCF, and IBCF were used as basic learners. The experimental results showed that STEM provided better group recommendation strategies than existing technologies [6]. Sivabalaselvamani and other researchers proposed a clustering recommendation method based on user social information to address the sparsity problem in online Social Rating Networks (SRNs). This method shaped specific informal communities through daily collaboration between users, such as co-evaluating books and movies. Results showed that the proposed method had an accuracy improvement of 23.1% in predicting ratings and recommending items [7]. Li et al. explored how recommendation systems induce consumers to purchase and used recently developed causal mediation methods to explore these causal pathways. The experimental results indicated that personalized recommendations presence increased consumer purchase intention by 12.4% and shopping basket value by 1.7% [8]. Ananda M R and other researchers proposed a CF recommendation method by Android and IBCF to increase sales of a certain product. Results showed that this method increased product sales by 23.21% [9].

Montesinos-López O A proposed a genome prediction model using IBCF to solve the problem of low accuracy of Genome Best Linear Unbiased Prediction (GBLUP) in genome prediction. The experimental results showed that IBCF had high efficiency in implementation time, reaching over 400 times. The limitation was that this experiment was conducted under univariate conditions, so IBCF may perform better on multivariate data [10]. A high preference page recommendation system based on user access history was proposed by others to address the issue of low accuracy of content recommendation targets. The system combined the Jaccard index for IBCF, and results showed that the system accuracy was as high as 98%, making it a content recommendation system that was in line with marketing strategies [11]. Xiao and other researchers used CF as a new project selection method to address the difficulty of selecting important design features in computerized adaptive testing. The experimental results showed that CF outperformed traditional project selection methods, while IBCF performed well in project library utilization [12]. Zhang proposed a recommendation method that integrates bipartite graph networks into UBCF to address the cold start problem of UBCF for traditional users. Using normalized scoring information, a new weighted bipartite modal index was proposed for community partitioning, achieving clustering of users and projects. The experimental results showed that it was superior in terms of recommendation accuracy and diversity, and effectively alleviated the cold start problem in UBCF [13]. Poudel et al. proposed a density-based random stratified sampling algorithm to address the high computational cost of CF algorithms. This algorithm maintained the overall density while down sampling according to the specified down sampling ratio of users and items. By extensively simulating various classic CF algorithms, models for training time improvement (TTI) and accuracy loss (AL) were established. The results showed that both TTI and AL had a linear relationship with FUD and FID, and TTI had the same regression results for various datasets [14].

Although the previous studies achieved some achievements in the field of recommendation systems, there are still some shortcomings. First, the problem of data sparsity remains one of the major challenges facing recommendation systems, especially when user behavior data is scarce, and the accuracy of recommendation algorithms can be seriously affected. Secondly, many studies have only focused on improving recommendation accuracy while ignoring recommendation diversity and novelty, which may cause users to fall into the information cocoon. Moreover, some algorithms are computationally expensive when processing large-scale datasets, limiting their generalization in practical applications. Finally, many studies lack consideration for user privacy protection, which may raise user concerns about data security and privacy. In view of this study, a university library book recommendation system based on IIBCF algorithm was designed, and combined with UBCF algorithm to provide more personalized services for university libraries.

III. IIBCF ALGORITHM AND DESIGN OF BOOK RECOMMENDATION SYSTEM

A. THE OVERALL ARCHITECTURE OF THIS SECTION

This chapter consists of three sections. The section proposes a personalized book intelligent recommendation system based on IBCF, which includes three parts: data input, IBCF processing, and data output. An IBCF algorithm based on mean model representation was proposed to reduce the length of projects, to achieve the goal of reducing computational time.

The second section is mainly based on the IIBCF algorithm proposed in the first section, which is combined with the UBCF algorithm to fully leverage the advantages of each algorithm and further improve the IBCF algorithm. Through a series of improvement measures such as clustering-based filling methods, the IIBCF recommendation model is designed.

The third section mainly proposes a university library book recommendation system based on the IIBCF model, and designs various aspects of the system. The technologies used in the system mainly include the IIBCF recommendation algorithm, SparkMLlib (machine learning library based on Spark),HDFS (distributed file system), Spark (parallel computing framework), EasyExcel (Excel data processing plugin), Java SSM framework, and MySQL (web development technology).

B. IBCF ALGORITHM USING MEAN MODEL REPRESENTATION

IBCF is a CF recommendation algorithm that calculates the item's similarity based on user historical behavior data and then recommends similar items that have not yet been selected by the user to the user [15], [16]. Its basic idea is that if a user likes a certain project, they may like other projects similar to that project. The IBCF algorithm, along with UBCF and MBCF algorithms, belongs to CF-based recommendation algorithms. Compared to UBCF and MBCF, IBCF relies more on user historical behavior data and has higher recommendation accuracy and stability [17], [18]. Therefore, the study is based on the IBCF algorithm to construct a personalized book intelligent recommendation system. The IBCF-based personalized book intelligent recommendation system includes three parts: data input, IBCF processing, and data output. The first part is the data input section, inputting user behavior data, book information data, user book rating data, and domain knowledge data. User behavior data, including user purchase records, browsing records, collection records, etc., can reflect user preferences and needs for books. Book information data includes information such as the title, author, publisher, publication time, and classification of the book, which can describe the basic characteristics and attributes of the book. User book rating data includes user ratings, reviews, and more on books. Domain knowledge data includes professional knowledge and background knowledge in the field of books, which can provide a deep understanding and understanding of the book field. Then there is the IBCF algorithm section, which is divided into three stages: data pre-processing stage, book similarity calculation stage, and recommended book calculation stage. In the data pre-processing stage, the main task is to clean and process the original data, including data de-duplication, missing value handling, outlier handling, etc. In IBCF, it is necessary to parse the interaction data between users and items and convert it into a sparse matrix. The book similarity calculation stage's main goal is to calculate the similarity between books. The recommended book calculation stage mainly calculates the recommended book score based on user historical behavior and book similarity. Usually, weighted summation or simple accumulation methods are used to determine the score of recommended books. When calculating the score of recommended books, the historical behavior of users and the weight of book similarity will be considered. Finally, the obtained results are output, and the structural diagram of the personalized book intelligent recommendation system using the IBCF algorithm is shown in Figure 1.

The IBCF algorithm can calculate project similarity offline. Similarity calculation is the core of algorithms, and the commonly used calculation method is cosine similarity. Its mathematical expression is shown in equation (1).

$$sim(i,j) = \frac{i * j}{||i||_2 * ||j||_2} \tag{1}$$

In equation (1), i represents a vector in the user space. If the user does not rate i, it will be set to 0. The similarity of projects will be represented by the angle between the cosines of two project vectors. Same as i, j represents another vector in user space. When calculating the similarity between users, cosine similarity only considers the directionality between user ratings, without considering the possible deviations between different user ratings. The adjusted cosine similarity takes into account the bias between user ratings. It introduces the mean of the item to correct the original cosine similarity and therefore studies the modified cosine similarity. In addition, in recommendation systems, user ratings of items are often sparse, that is, users have only rated a few items. The modified cosine similarity can be calculated by considering the user's score of the co-rated item, so the sparse datasets can be processed. For these reasons, the study chooses the modified cosine similarity, whose mathematical expression is shown in equation(2).

$$sim(i,j) = \frac{\sum\limits_{u \in U_{ij}} (r_{u,i} - \overline{r}_1) * (r_{u,j} - \overline{r}_j)}{\sqrt{\sum\limits_{u_i} (r_{u,i} - \overline{r}_i)^2} * \sqrt{\sum\limits_{u_j} (r_{u,j} - \overline{r}_j)^2}} \tag{2}$$

In equation (2), $r_{u,j}$ represents the user's rating of the project i. $\overline{r}_i$ represents the mean of the project i rating vector. $\overline{r}_j$ represents the mean of the project j rating vector. IBCF has high recommendation accuracy and a simple implementation method, making it easy to apply to practical systems. However, due to the long item vector, calculating similarity is very time-consuming. To address this issue, the study uses a mean square model to represent the item vector to shorten the time for calculating similarity. The essence of the mean model is a complete binary tree, and the equation for converting the item rating vector into the mean model item vector representation is shown in equation (3).

$$I_i' = T_k(I_i) \tag{3}$$

In equation (3), I_i represents the scoring vector of the i item and T_k represents the transformation equation of the tree's k layer, as shown in equation (4).

$$T_k(a_{ij}|a_{ij} \in A) = \frac{\sum\limits_{a_{ij}}}{card(A)} \tag{4}$$

In equation (4), $card(A)$ represents the cardinality of any set A. The expression of the scoring vector for the i item is shown in equation (5).

$$I_i = \{r_{1,i}, r_{2,i}, \ldots, r_{m,i}\} \tag{5}$$

In equation (5), m represents the number of users, and the steps for averaging the model vector are shown in Figure 2.

Firstly, the root node element t_0 is calculated according to T_0, as denoted in equation (6).

$$t_0 = T_0(r_{j,i}|r_{j,i} \in I_i) = \frac{\sum\limits_{j=1}^{m} r_{i,j}}{m} \tag{6}$$

Then the scoring vector is divided into two vectors, as shown in equation (7).

$$\begin{cases} I_i^{10} = \{r_{i,j}|r_{i,j} < t_0, 1 \le j \le m\} \\ I_i^{11} = \{r_{i,j}|r_{i,j} \ge t_0, 1 \le j \le m\} \end{cases} \tag{7}$$

Then I_i^{10}, I_i^{11} are composed into two vectors, where the decomposed I_i^{10} vectors are shown in equation (8).

$$\begin{cases} I_i^{20} = \{r_{i,j}|r_{i,j} < t_{10}, 1 \le j \le m\} \\ I_i^{21} = \{r_{i,j}|r_{i,j} \ge t_{10}, 1 \le j \le m\} \end{cases} \tag{8}$$

In equation (8), t_{10} represents the 0-th element of the first layer of the binary tree, and its calculation is shown in equation (9).

$$t_{10} = T_1(r_{i,j}|r_{i,j} \in I_i^{10}) \tag{9}$$

The two decomposed I_i^{11} sub-vectors are shown in equation (10).

$$\begin{cases} I_i^{22} = \{r_{i,j}|r_{i,j} < t_{11}, 1 \le j \le m\} \\ I_i^{22} = \{r_{i,j}|r_{i,j} \ge t_{11}, 1 \le j \le m\} \end{cases} \tag{10}$$

The calculation of t_{11} in equation (10) is shown in equation (11).

$$t_{11} = T_1(r_{i,j}|r_{i,j} \in I_i^{11}) \tag{11}$$

The uniform model is represented by a fully binary tree based on sorting, which integrates the characteristics of the tree structure and the statistical properties of vectors, providing an efficient and flexible method for processing complex data. The uniform model adopts the form of a complete binary tree, which means that except for the last layer, each layer is filled, and all nodes remain aligned to the left. This structure helps to achieve efficient storage and retrieval. In the uniform model, the transformation expressions of each layer are closely related to the elements of the previous layer. This hierarchical dependency ensures that data can maintain its inherent structure and correlation when transmitted layer by layer. The 0-th layer has only one point t0 as the root node of the binary tree. The root

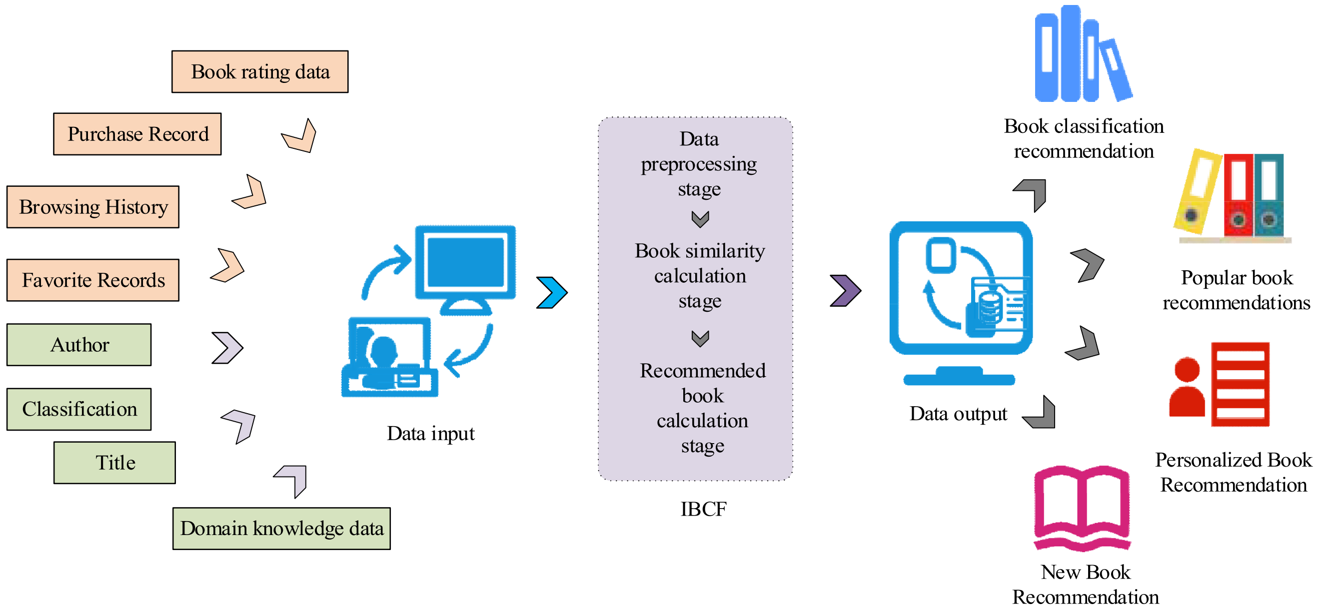


FIGURE 1. Structure diagram of personalized book intelligent recommendation system using IBCF algorithm.

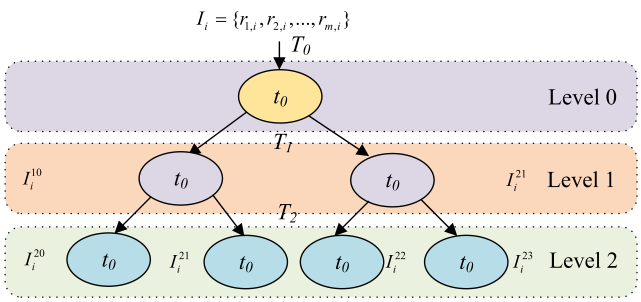


FIGURE 2. Schematic diagram of the generation process of the mean model vector.

node plays a core role in the entire tree structure, connecting all other nodes and carrying global information. In addition, the mean model vector representation method can perform dimensionality reduction on high-dimensional data without losing important information. By reducing the dimensionality of data, computational efficiency can be improved and storage costs can be reduced. During the dimensionality reduction, the mean model can find a balance point, ensuring both computational efficiency and data accuracy. This is particularly important for processing large-scale datasets. The mean model vector representation method can also smooth out the noise in the data. By reducing the interference of noise, the robustness of the model can be enhanced and its tolerance for outliers can be improved. Based on the mean model vector representation, the definition of project similarity is given as shown in equation (12).

$$\mathrm{sim}(i,j) = \frac{\mathbf{I}_i' * \mathbf{I}_j'}{\left\|\mathbf{I}_i'\right\| * \left\|\mathbf{I}_j'\right\|^{\circ}} \tag{12}$$

In equation (12), the average model vectors of two items i and j are represented as $\mathbf{I}_i'$ and $\mathbf{I}_j'$.

C. IIBCF ALGORITHM DESIGN COMBINED WITH UBCF

The UBCF algorithm is a recommendation algorithm using user similarity, which mainly analyzes the evaluation of the same book by different users, calculates the similarity between users, and is used as the main basis for recommendation [19]. This algorithm is widely used in recommendation systems because it can help the system better understand users' interests and behavior patterns, thereby providing more personalized recommendations [20], [21], [22]. Specifically, the UBCF algorithm can calculate the similarity between users by analyzing their historical behavior, preferences, and interests, and use these similarities as the main basis for recommendations. The UBCF algorithm process is shown in Figure 3.

The book recommendation process includes collecting user evaluation data such as historical purchase records, browsing records, and collection records. The mean and variance of each book's rating are then calculated using the user's book evaluation data. Next, similarity between users is calculated using modified cosine similarity to identify similar users, who serve as neighbors to the target user. Unrated books are predicted for the target user based on their neighbors' ratings through weighted averaging or voting. The highest-rated books are recommended through sorting algorithms, and personalized book recommendation services are provided to the target user. Continuous feedback is used to update recommendation results and improve accuracy. In the personalized book intelligent recommendation system, the UBCF algorithm often faces the problem of data sparsity. This is because the number of users in university libraries is

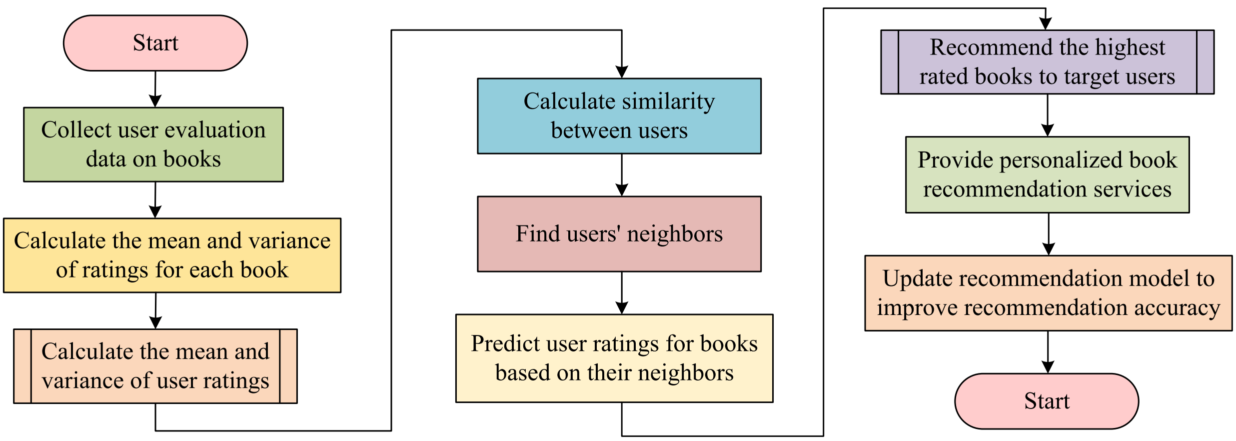


FIGURE 3. **Basic process of UBCF book intelligent recommendation algorithm.**

large, and each user has relatively little data on borrowing or evaluating books. This leads to very sparse data in the system, where most users do not have borrowing or evaluation records for most books. The issue of data sparsity can have a negative impact on the recommendation performance of the UBCF algorithm, as it can lead to inaccurate item similarity calculation and cold start issues in recommendations. Specifically, due to data sparsity, it is difficult for algorithms to accurately calculate the similarity between books, thereby affecting the accuracy of recommendations. In addition, for new users or books, it is difficult for algorithms to provide accurate recommendations due to the lack of sufficient borrowing or evaluation data. To address this issue, a clustering-based filling method is used in the study. This method aggregates similar data points into one class and then fills in missing values for each class using the center value of that class. The specific steps are shown in Figure 4.

The book clustering process involves several steps. In the first step, data pre-processing is performed to clean and process the original data, including de-duplication, missing value handling, and outlier handling using advanced algorithms. This ensures that the dataset is free from duplicate records and outliers. In the second step, feature selection is carried out to identify relevant features that can reflect the similarities or differences between books. These features can include text features (such as keywords and descriptions), content features (such as chapters and paragraphs), labels and classification features (such as book genres and themes), and user behavior characteristics (such as borrowing records and clicking behavior).In the third step, after feature selection, the K-means algorithm is applied to cluster similar books based on the selected features. To evaluate the quality of the clustering, the contour coefficient is used as an objective evaluation index. The contour coefficient measures the distances between sample points within a cluster and the other clusters. A higher contour coefficient value indicates better clustering. In the fourth step, category centers are calculated, representing the overall characteristics of each cluster. This is done by calculating the sum of the squares of the distances from each sample point in the cluster to its center. The Euclidean distance is commonly used for this calculation. Overall, these steps help in organizing books into meaningful clusters based on their shared characteristics.

$$\min j \in (1, 2, \ldots, k)||x_i - c_j||2 \tag{13}$$

In equation (13), x_i represents the sample i and c_j represents the center of the cluster j. The calculation for its criterion function is shown in equation (14).

$$J = \sum i = 1n \min j \in (1, 2, \ldots, k)||x_i - c_j||2 \tag{14}$$

In equation (14), n represents the number of samples and k represents clustering. By continuously iterating and calculating, each sample point is made as close as possible to its corresponding category center, thus forming a cluster. The fifth step is to identify missing values by checking for empty, abnormal, or inconsistent values in the data, and to identify samples with missing values in the original data. The following is to determine the filling value for missing data. This involves assigning the missing samples to the appropriate cluster based on their characteristics and inferring the most suitable values to fill in the missing data by analyzing the patterns within each cluster. Once the cluster to which the missing value belongs is determined, the next step is to use the center point value of the cluster as the filling value for the missing value. The cluster center point represents the average or central trend of all samples within the cluster, so using the center point value as the filling value can ensure consistency and similarity between the filled data and other samples within the cluster. Finally, the filling effect is evaluated and the MAE is used to evaluate the filled data to understand the filling effect. The calculation is shown in equation (15).

$$MAE = \frac{\sum_N (|r_i - p_i|)}{N} \tag{15}$$

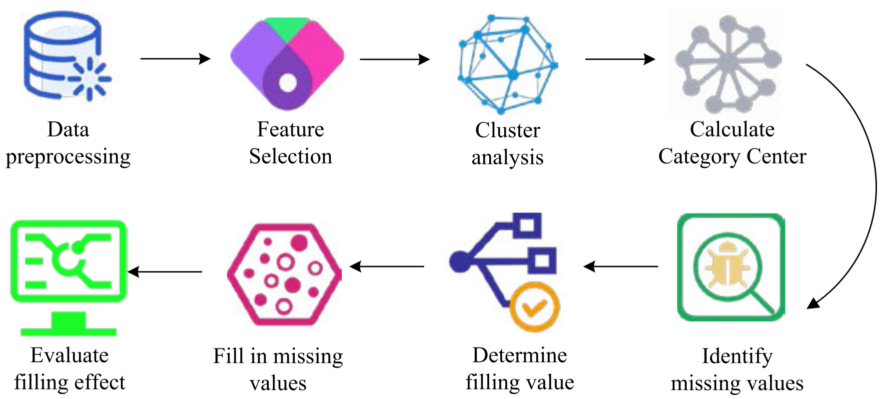


FIGURE 4. The basic process of data filling based on clustering filling method.

In equation (15), N represents the size of the test set, p_i represents the predicted score, and r_i represents the true score. By using clustering-based filling methods, data sparsity in personalized book intelligent recommendation systems in university libraries can be effectively solved, thereby improving recommendation effectiveness and user satisfaction. Both the UBCF algorithm and the IBCF algorithm can play important roles in recommendation systems, but their functions in book intelligent recommendation systems are slightly different, as shown in Figure 5.

The UBCF algorithm and IBCF algorithm are widely used in recommendation systems, and they play different roles in personalized book intelligent recommendation systems in university libraries. The UBCF algorithm mainly analyzes the "preference" behavior of users towards a shared set of products. It calculates user similarity and identifies users with similar interests to the target user, recommending books based on their preferences. This algorithm can recommend books based on the similarity between users, thereby improving the accuracy of recommendations and user satisfaction. The core idea of IBCF algorithm is to recommend similar projects to users from the perspective of project similarity, and pay more attention to the similarity between projects. If the user likes an item, the system will find other items that are highly similar to this item and the user has not yet chosen, and then recommend them to the user [21], [22]. It will recommend items by the target user's historical interests based on the similarity between items. This algorithm can recommend books based on the similarity between items, thereby meeting the personalized needs of users and improving the accuracy of recommendations. By combining the IBCF algorithm and UBCF algorithm, it is possible to simultaneously utilize user historical behavior data and other user behavior patterns for prediction and recommendation. This helps to improve the coverage of recommendations, meaning that the recommendation system can cover more books and meet more user needs.

D. PERSONALIZED BOOK INTELLIGENT RECOMMENDATION SYSTEM DESIGN USING IIBCF

In the overall scheme of the personalized book intelligent recommendation system for university libraries based on the IBCF algorithm proposed in the research, the IBCF algorithm is mainly responsible for recommending based on user historical interests. It predicts user ratings for un-reviewed books by analyzing user historical behavior data and uses this as the main basis for recommendations. In addition, the IBCF algorithm can also discover books that users may be interested in by calculating the similarity between items. The UBCF algorithm is mainly responsible for recommending based on user similarity. It analyzes the evaluations of the same book by different users, calculates the similarity between users, and uses it as the main basis for recommendations. The UBCF algorithm can help the system better understand user interests and behavior patterns, thereby providing more personalized recommendations. Based on the IBCF algorithm and combined with the use of the UBCF algorithm, it can effectively utilize user's historical behavior data and similarity information to improve the accuracy of personalized book recommendations. In the system initialization phase, it will initialize the various components of the recommendation system, including the database, algorithm module, and user interface, and load necessary configuration parameters. Subsequently, in the information collection stage, it will collect historical behavioral data such as user borrowing records, browsing history, and evaluation information, and clean and process them to ensure data quality. Next, the process is divided into two branches. The first branch is based on the IBCF algorithm, which filters complete information, calculates similarity and finds similar users, fills in the sparse matrix, and recalculates similarity to find the target user's neighbor group. The similarity between books is used to predict the target user's rating of unread books. The second branch is based on the UBCF algorithm, which constructs a user rating matrix, calculates the similarity between users and finds similar books, uses an average model to represent item vectors, and searches for books that are

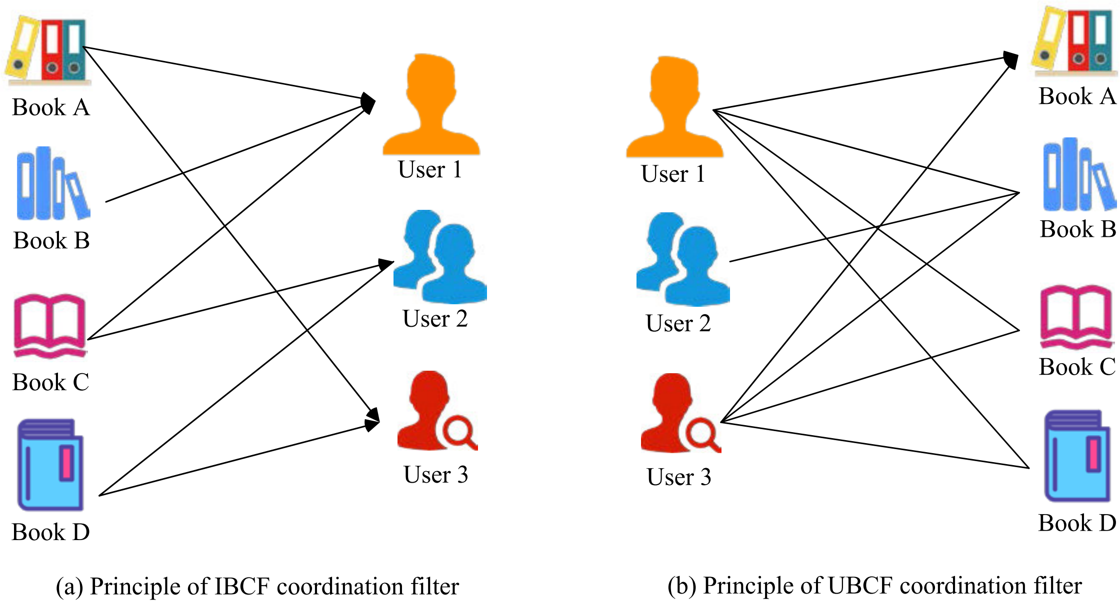


FIGURE 5. **The functions of UBCF algorithm and IBCF algorithm in the intelligent book recommendation system.**

similar to the target book. It uses the similarity between users to recommend books that are evaluated by similar users. Finally, the two branches converge and combine the information of the two branches to predict the rating of books that the target user has not evaluated and generate a preliminary list of recommendation results. By sorting and filtering the preliminary recommendation result list, the diversity and accuracy of the recommendation results are ensured, and the final recommendation results are presented to the target users in various forms such as book covers, introductions, and evaluation information. By integrating the advantages of IBCF and UBCF algorithms, this solution can comprehensively understand user interests and behavior patterns, provide more personalized and accurate book recommendation services, and meet the growing demand for personalized reading by users. The algorithm implementation process is shown in Figure 6.

The personalized book intelligent recommendation system proposed in the study for university libraries has implemented three recommendation functions: new book recommendation, popular book recommendation, and personalized recommendation. The implementation of these functions not only significantly improves the operational efficiency of the library, but more importantly, provides users with a more intelligent and personalized service experience. When the library introduces new book resources, the system can automatically perceive this change and immediately notify users through a notification mechanism. This recommendation is not only based on the category and content of the book but also fully considers the user's professional field and reading interests, ensuring that users can timely access the latest academic materials or new books that meet their personal preferences. The popular book recommendation function is to analyze the borrowing data of the library in real time, organize books with high borrowing frequency into a comprehensive list, and further filter out popular books related to the user's professional field and personalized needs based on the IIBCF personalized recommendation model for the recommendation. This approach not only allows users to easily grasp the current academic trends and reading trends but also better meets their reading needs in specific fields, improving the reading experience and efficiency of academic research. To further improve the accuracy and efficiency of personalized recommendations, the system integrates the advantages of the IBCF and UBCF algorithms. This fusion algorithm considers both the similarity between items and the similarity between users, allowing for more accurate discovery of books of interest to similar users. Based on user interests, borrowing history, and other information, it displays highly matched recommendation results to them. This intelligent recommendation method that integrates multiple algorithms not only improves the accuracy of recommendations but also greatly enhances the user experience and satisfaction. The reason for choosing these technologies is their respective advantages and complementarity. IIBCF can effectively handle the issues of new users and projects, improving the accuracy of recommendations. The recommendation of popular books combines borrowing data and user professional fields, which can better meet the professional needs of users. The algorithm that integrates IBCF and UBCF can fully utilize the similarity between items and users, improving the accuracy and efficiency of recommendations. The comprehensive application of these technologies enables the system to provide users with more

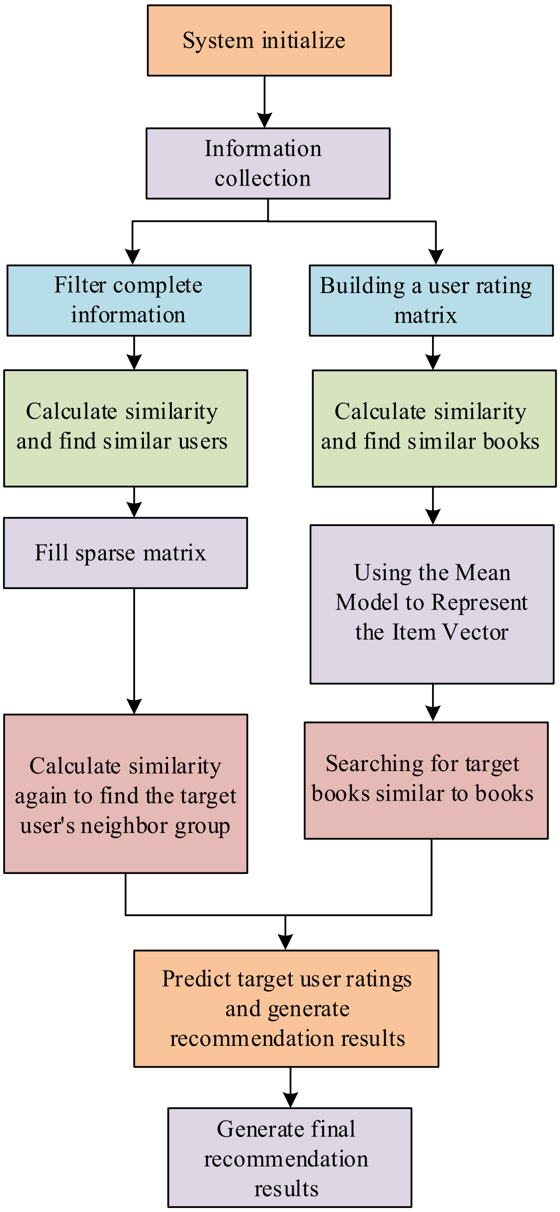


FIGURE 6. **Overall design of personalized book intelligent recommendation system for university libraries using IBCF.**

intelligent and personalized book recommendation services. The timing diagram of the system is shown in Figure 7.

The main technologies used in the system include a CF recommendation algorithm, SparkMLlib (a machine learning library based on Spark), HDFS (distributed file system), Spark (parallel computing framework), EasyExcel (Excel data processing plugin), Java SSM framework, and MySQL (web development technology). CF recommendation algorithm: This is the core technology of this article, used to implement a book collection recommendation system for university libraries. Specifically, this algorithm combines UBCF and IBCF. The UBCF algorithm compares user behavior with the behavior of other users and makes recommendations based on this. The IBCF algorithm recommends items based on their characteristics (i.e. books) and user evaluations of the projects. SparkMLlib: MLlib is Spark's machine learning library, which includes common machine learning algorithms and utilities such as classification, regression, clustering, CF, etc. In this study, SparkMLlib is used as the implementation core of CF recommendation algorithms. Compared to Mahout, SparkMLlib is more efficient and easier to integrate into Spark's ecosystem. By utilizing Spark's parallel computing capabilities, it is possible to effectively process copiousamountsof data and improve the efficiency and accuracy of recommendation algorithms. Although HDFS is a component of Hadoop, it can also be used in conjunction with Spark as a distributed file system for storing data. Itis used to store and process a large amount of user behavior data and book information, providing necessary data support for recommendation algorithms. Spark is a fast and versatile large-scale data processing engine. Compared to Hadoop's MapReduce, Spark provides richer APIs and more advanced data processing capabilities, such as DataFrame and Dataset operations. It is used to process user behavior data and book information, calculate recommendation results, and perform various data transformation and aggregation operations. EasyExcel is a Java plugin used to process Excel data, making it easy to read and write Excel files. It is used to process and analyze user behavior data and book information, which may be provided in Excel format and used to save recommendation results for further analysis or display. The Java SSM framework (Spring, Spring MVC, MyBatis) and MySQL database are commonly used technology combinations in Java Web development and data storage. They are used to achieve front-end and back-end interaction, data storage, and management functions of the system. By using the SSM framework, a clear hierarchical structure and modular development approach can be achieved, improving the maintainability and scalability of the system. The overall E-R model of the system is shown in Figure 8.

The overall E-R model of a personalized book recommendation system is shown above. In this model, rectangles represent entities, which objectively exist, while ellipses represent the properties of entities. The diamond represents the connection between entities and labels the type of connection on the un-directed edge. In terms of connection types, one-to-one (1:1) indicates that the relationship between two entities is a one-to-one correspondence, meaning that one entity can only have a unique connection with another entity, and vice versa. One-to-many (1: n) indicates that an entity can be associated with multiple entities of another type, but the opposite is not true, meaning that an entity can have multiple related entities, and related entities can only be associated with one entity. Many-to-many (m: n) indicates that there can be a many-to-many relationship between two entity types, meaning that one entity can be associated with multiple entities of another type, and vice versa. In the E-R model of a personalized book recommendation system, six entities - administrators, users, popular recommendation

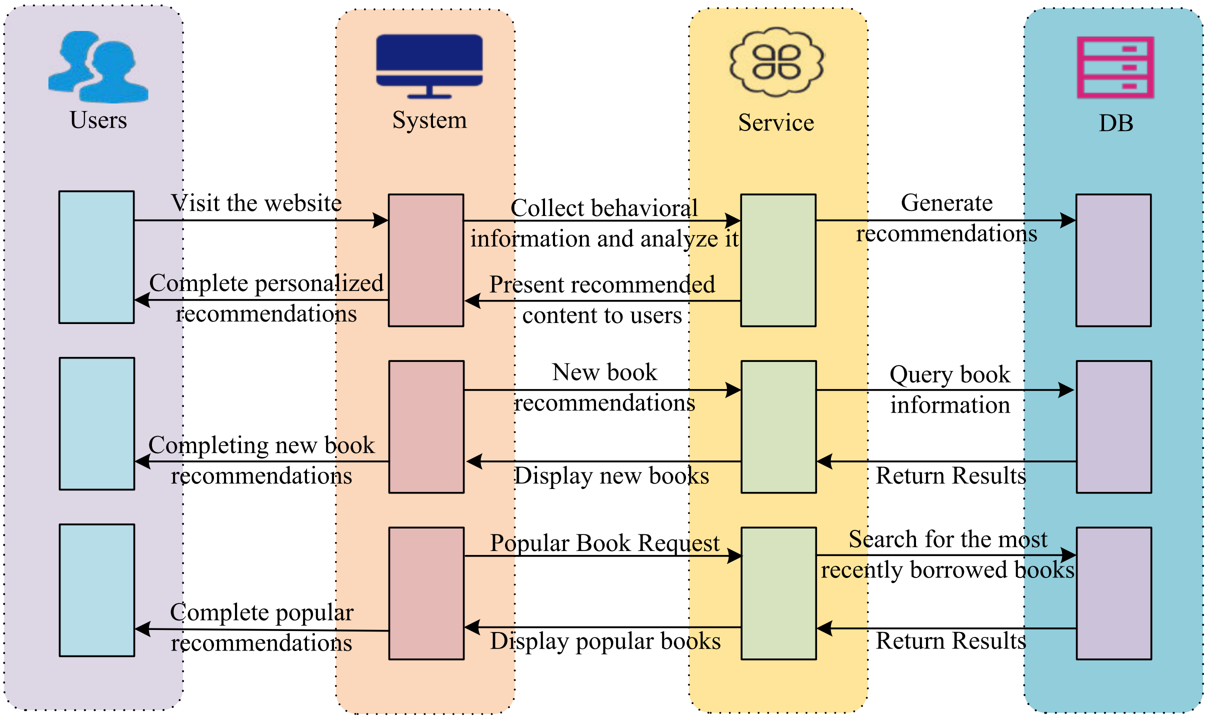


FIGURE 7. Timing chart of personalized book recommendation system.

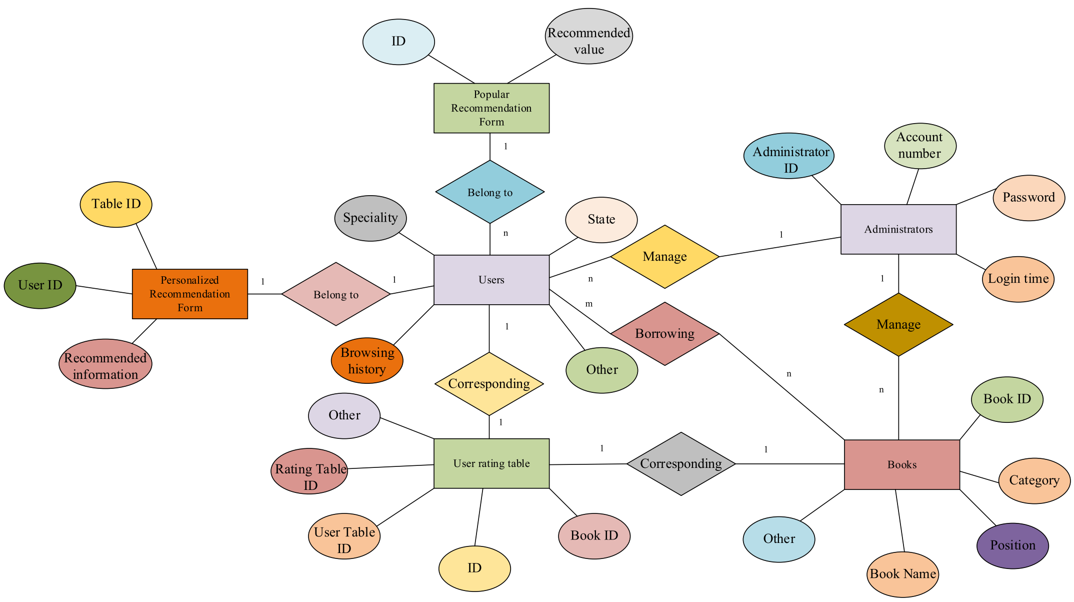


FIGURE 8. E-R diagram of personalized book recommendation system.

tables, personalized recommendation tables, user rating tables, and books - are intertwined through their respective connections, collectively forming a complete and complex recommendation ecosystem. Firstly, as one of the core entities of the system, administrators have the authority to comprehensively manage the system. They can create, edit, and delete user accounts, which means there is a one-to-many (1: n) management relationship between administrators and users, where one administrator can manage multiple users, but each user can only be directly managed by one administrator (in a specific operational context). At the same time, administrators can also add, modify, and delete books, which also reflects a one-to-many (1: n) connection, that is, an administrator can manage multiple books, but each book can only be operated by one administrator at the same time. User entities play the role of interacting and receiving recommendations in the system. They can browse and search for books, which means there is a many-to-many (m: n) connection between users and books, as users can browse multiple books, and each book can also be browsed by multiple users. In addition, users can also evaluate and rate books, and this evaluation information is recorded in the user rating table, which reflects a one-to-one (1:1) relationship, where each user's evaluation corresponds to a unique entry in the rating table. Popular recommendation tables and personalized recommendation tables are key entities in the system that implement recommendation functions. The popular recommendation table is generated based on factors such as the number of recommended books and the recommended time, while the personalized recommendation table generates a personalized recommendation list based on the user's historical behavior and the behavior of other similar users. Both recommendation tables have a many-to-many (m: n) relationship with users and books, as the recommendation table can contain recommendation information for multiple books, and each book can also appear in multiple recommendation tables. Similarly, each user can receive recommendations from multiple recommendation tables, and each recommendation table can also serve multiple users. As a fundamental element in the system, book entities contain detailed information about books. These pieces of information are crucial for users to browse, search, and evaluate books. A one-to-many (1: n) relationship is established between books and user rating sheets through rating behavior, meaning that each book can be rated by multiple users, but each rating can only correspond to one book.

IV. IIBCF ALGORITHM PERFORMANCE TESTING AND RECOMMENDATION MODEL APPLICATION EXPERIMENT

This chapter is mainly divided into two sections. The first section assessed the IIBCF performance proposed in the study and compared the algorithms horizontally. The second section mainly focused on the book recommendation system based on the IIBCF algorithm to verify its application effect.

A. DATASETS

The experiment used datasets obtained from the database of a university library management system in Shanghai, containing 513,546 books and 94.561 million user data points. Of these, there were 185,121 borrowing records and over 200,000 other attributes. To avoid the influence of temporal order in the time series data on model performance, the data set was divided into 7:3 scale training set and test set. The dataset spans from January 2010 to December 2020, with split points in September 2016. In the experiment, the UBCF module had 50 neighbors (k), the IBCF module had 20 neighbors (k), and the damping factor was set to 0.01. An example in the table 1 is described for the book evaluation data.

TABLE 1. Evaluation data of some books from a certain university in Shanghai.

User ID	Score	Book ID
U231579	1	B8796543
U231544	3	B8796589
U231946	5	B8798654
U231657	5	B8793642
U231423	1	B8756489
U231364	5	B8765454

B. EXPERIMENTAL SETUP

The experiments were conducted on a Windows 10 operating platform using an Intel i5-2500K CPU. Both the IIBCF algorithm and the traditional IBCF algorithm were implemented in Python, running on this hardware platform and operating system. The CPU used was an Intel i5-2500K with four cores, a primary frequency of 3.3 GHz, and multilevel caches (such as L1, L2, and L3 caches). The experiments made use of multiple data resources to train the most appropriate model. Specifically, 35,686 detailed book reviews were carefully extracted from the evaluation system of a renowned university library in Shanghai. These reviews not only provided a large quantity of data but also maintained a high level of quality, thus offering robust support for training the book recommendation model. A personalized book recommendation system was designed based on IBCF and combined with UBCF. Due to the long item vector, calculating similarity was very time-consuming. To address this issue, the study used a mean square model to represent the item vector to shorten the time for calculating similarity. To verify the effectiveness of this method, the study conducted 100 parallel comparison experiments with the traditional IBCF method, using the computational time and performance simplification method as evaluation indicators.100,200,500,800 and 900 user data were taken for time calculation experiments. To verify whether this performance improvement is statistically significant, paired sample t-tests were conducted in the experiment. This test compared the running time of algorithms before and after compressing vector dimensions under the same user data size. Real time CPU usage data was obtained and recorded for comparative analysis. To determine the optimal number of nearest neighbors for the algorithm, 10 to 100 nearest neighbors were selected for accuracy evaluation, with MAE as the evaluation indicator. To fully validate the superiority of the personalized book intelligent recommendation model for university libraries based on the IBCF algorithm proposed in the study, the experiment introduced Content-based Collaborative Filtering (CBCF) and Hybrid Recommendation (HR) algorithms for horizontal comparison experiments.

The experiment repeatedly demonstrated the excellent superiority of the IIBCF algorithm in book recommendation systems. However, this move only stayed at the theoretical level and needed to be further applied to practical book recommendation work, thus more accurately testing whether the algorithm can effectively meet the diverse needs of students. Therefore, the personalized book recommendation system based on the IIBCF algorithm proposed in the experiment was applied to the library of the school and rigorously tested. During the testing process, the personalized recommendation function of the system was first tested. The experiment also tested the recommendation time of the two functions of new book recommendation and popular book recommendation. To fully study the superiority of the system, the Accu-NRecSys method proposed in reference [23] and the SPSRB method proposed in reference [24] were compared and applied to actual book recommendation tasks, with coverage and IO response time as evaluation indicators. Coverage is an important indicator to measure whether a recommendation system can widely cover user interest areas. High coverage means that the system can recommend more diverse and comprehensive book resources to users, thereby meeting a wider range of user needs. The IO response time reflects the speed at which the system processes recommendation requests. A shorter IO response time means that the system can provide recommendation results to users more quickly, which is crucial for improving user experience. In terms of demographic data, the study collected basic information about participants, including gender, age, major, and grade. These pieces of information are conducive to analyzing whether there are differences in the evaluation of system performance among different user groups and provide a reference for subsequent system improvements. Finally, the study summarized the rating data of all participants and plotted the score results.

C. RESULT AND DISCUSSION

Figure 9 shows the impact of compressed vector dimension and uncompressed vector dimension on the running time of each algorithm when the nearest neighbor k=90. From Figure 9 (a), without compressing the vector dimension, the algorithm running time increased as the vector dimension increased. Among them, the traditional IBCF algorithm had the largest increase, with a 40000 increase in running time from 100 users to 900 users. The increase exceeded 400%. By breaking the constraint of strict matching of object attributes, the running time of the algorithm was significantly reduced. Figure 9 (b) shows the percentage improvement in time performance of each algorithm after compressing the vector dimension, and its calculation is shown in equation (16).

$$p = \frac{t_0 - t_t}{t_0} * 100\% \tag{16}$$

In equation (16), t_0 represents the algorithm running time that does not compress the vector dimension, and t_t represents the algorithm running time that compresses the vector dimension. From the figure, when the vector dimension was 100, the improvement in Level 1 was the greatest, and when the vector dimension was greater than 800, the differences among the algorithms were not significant. To verify the generalization ability of the model, the study conducted K-fold cross-validation in the Titanic dataset. The results are shown in Figure 9 (c). It can be seen that the performance of the model was not much different in different data sets, which proved that the model had high robustness.

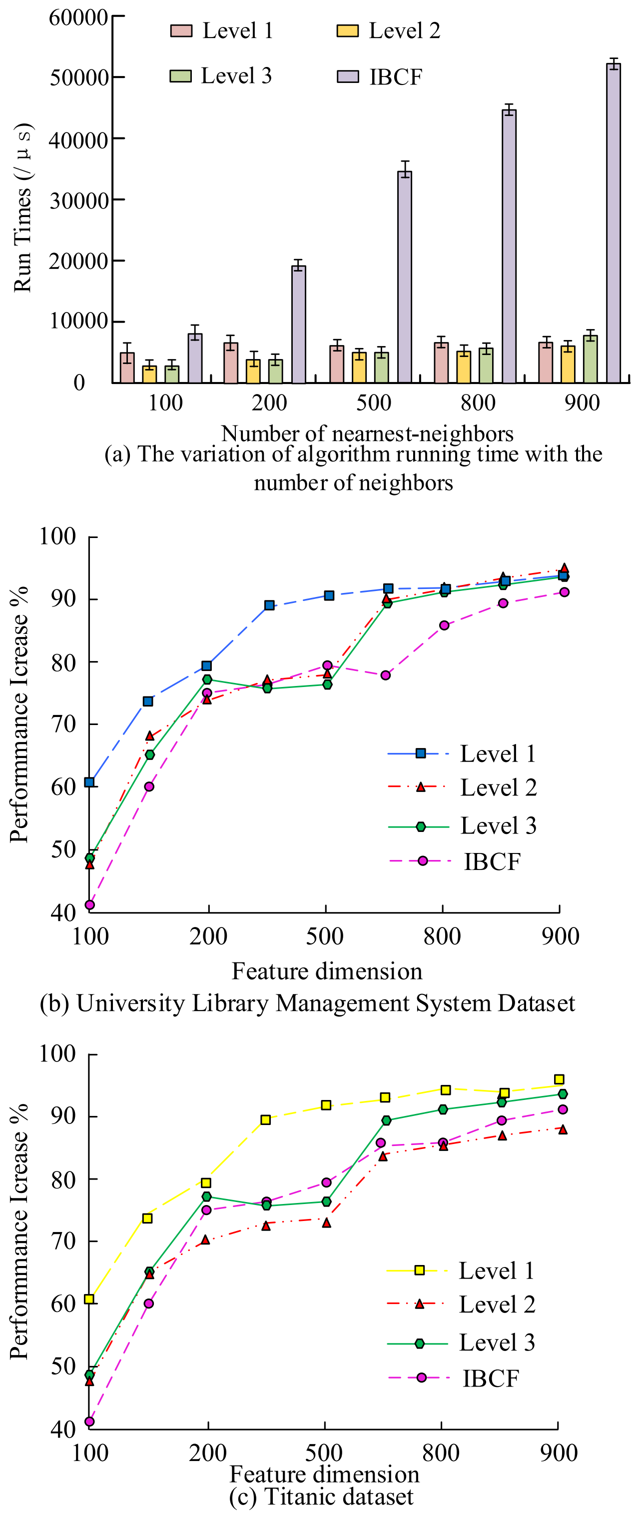
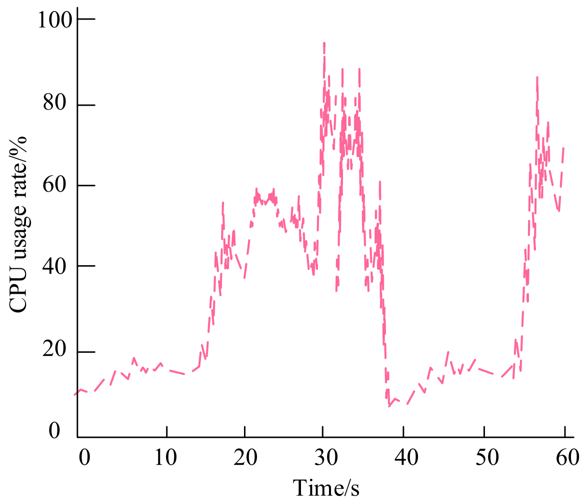
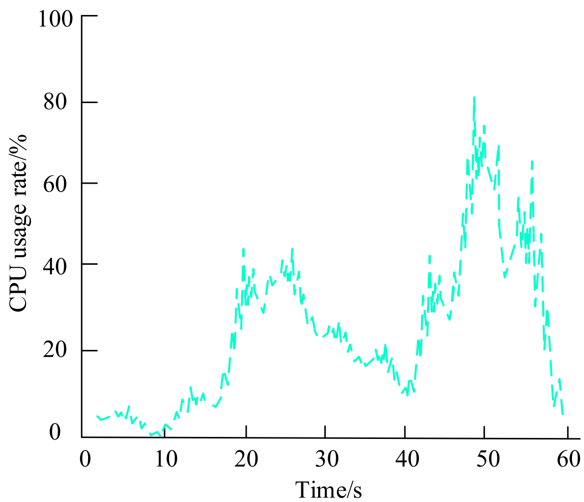


FIGURE 9. **Improving the performance analysis of different user sizes of the IBCF algorithm.**



(a) IBCF CPU usage rate



(2) IIBCF CPU usage rate

FIGURE 10. The utilization rate of computer CPU by the intelligent logistics system.

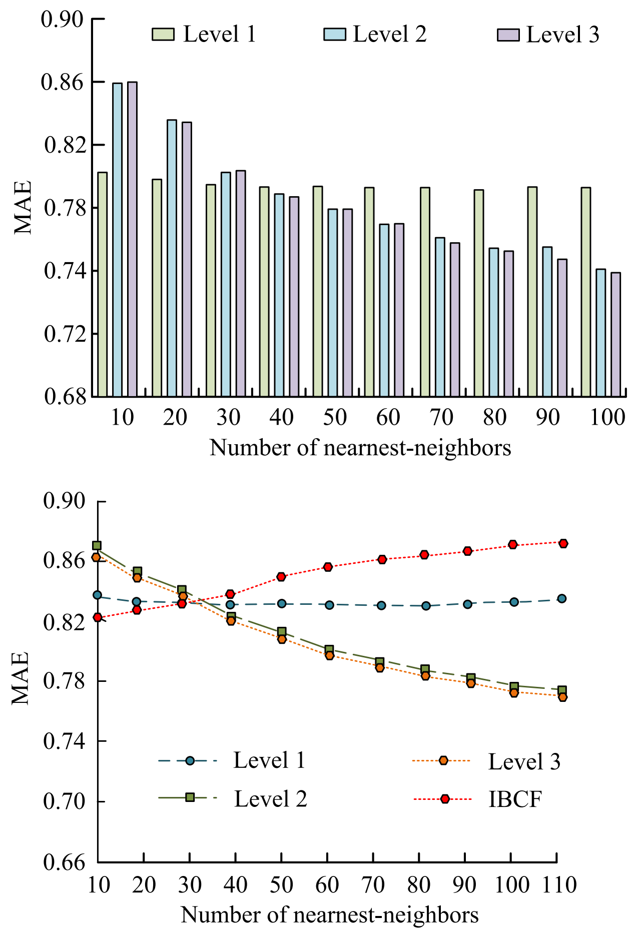
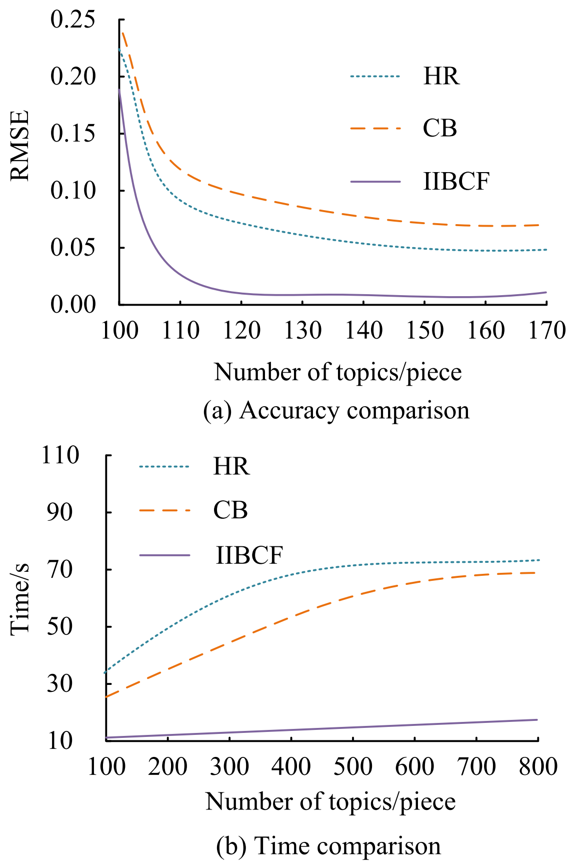


FIGURE 11. MAE value varies depending on the number of neighbors.

Figure 10 shows the CPU usage rate of the system using the standard IBCF algorithm during operation. The overall system CPU usage rate was high during the algorithm operation, with an average usage rate of 13.8% and a peak utilization rate of 93.6%. Figure 10 (b) shows the CPU usage rate of the system using the IIBCF algorithm during operation. The CPU usage rate of the entire system was not high during the algorithm operation, with an average usage rate of about 9.8% and a peak usage rate of about 73.5%. Although at some times the CPU usage may peak at 73.5%, compared to Figure 10 (a), the overall CPU usage of the system was much lower. This indicated that the algorithm performance was improved, and the system was more stable and reliable.

Figure 11 shows the performance test results of the item vector algorithm represented by the mean model. In Figure 11(a), the accuracy of level 2 and level 3 was very close, and as nearest-neighbors increased, the MAE value gradually decreased. The accuracy of level 1 was relatively stable, and when the nearest-neighbor k was within the range of [10, 30], the effect of level 1 was better than that of level 2 and level 3. In Figure 11 (b), the traditional IBCF algorithm was optimal when the nearest-neighbor k was within the range of [10, 30]. When it was within the range of [30, 100], the CF algorithm using the mean model had good performance.

Figure 12 shows the comparison of the RMSE and running time of the three algorithms. As samples increased, the three algorithms' RMSE continued to decrease. Among them, the IIBCF,CBCF, and HR algorithms had the lowest RMSE of 0.013, 0.068,and0.0868, respectively. Figure 12 (b) shows the running times comparison of the three algorithms. As samples increased, the three running times continued to increase. When the number of samples was 100, the HR,CBCF, and IIBCF running times were 35000 μs, 28000 μs, and 12000 μs. When the sample size was 800, the HR,CBCF, and IIBCF running times were 74000 μs, 67000 μs, and 12000 μs.



(a) Accuracy comparison

(b) Time comparison

FIGURE 12. Comparison results of RMSE and running time.

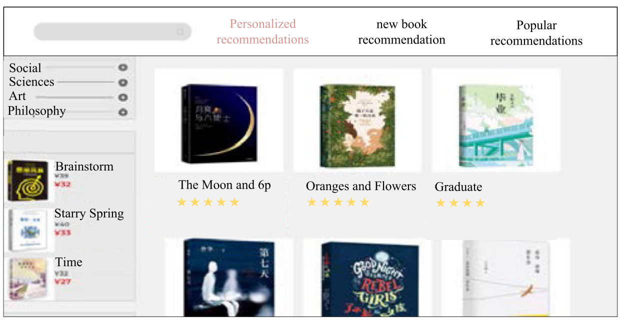
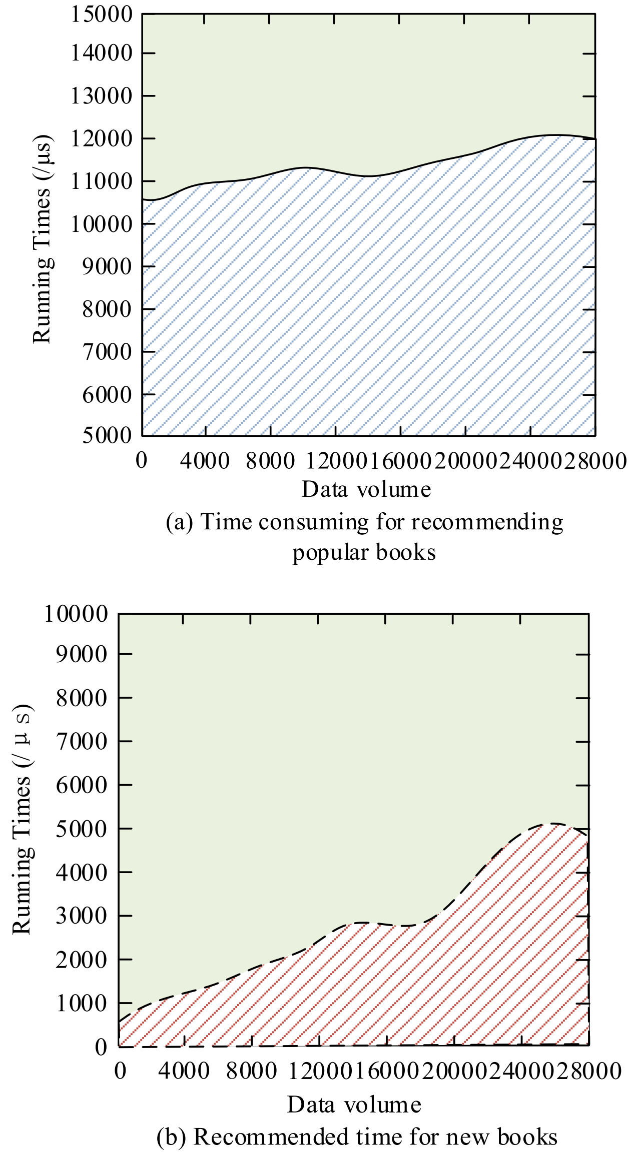


FIGURE 13. System screenshot of personalized recommendation function.

Figure 13 shows a detailed screenshot of a personalized recommendation function system. In this screenshot, a concise and clear user interface was shown. The system generated a personalized book recommendation list for users based on their browsing history and IIBCF calculation results. Each recommendation in this list included information about the book and a recommendation index, which was represented in the form of 1 to 5 stars. This recommendation method was very intuitive, and users could easily understand which books they may be interested in. In addition, on the interface left side was the book's classification information.



(a) Time consuming for recommending popular books

(b) Recommended time for new books

FIGURE 14. The recommended time for the two functions of adding book recommendations with popular book recommendations.

These classifications were meticulous and covered almost all possible book types. For example, there were arts, philosophy, society, science, etc. These classifications made it more convenient for users to search for books of interest, and readers of any type can find suitable reading materials here. This further reflected the refinement of the personalized recommendation system, which was designed from the perspective of the user.

Figure 14 shows in detail the recommendation time of the newly added book recommendation and popular book recommendation functions in the system. When observing Figure 14 (a), the time-consuming performance of popular book recommendations was quite stable, with almost no significant fluctuations. This was mainly due to the system's precise recording of popular books, which effectively reduced

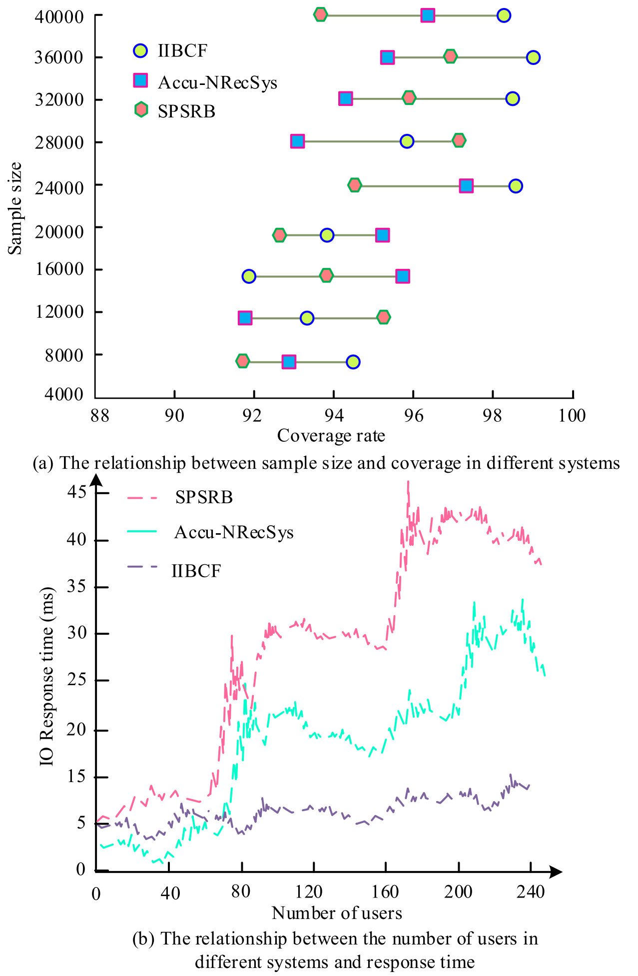


FIGURE 15. **Changes in coverage and IO response time during different system operations.**

the amount of effective data and thus reduced computational costs. For the new book recommendation function in Figure 14 (b), its time consumption showed an increasing trend with the gradual increase of data volume. This reflected a positive correlation between the computational complexity of this function and the amount of data. Overall, the time consumption of the newly added book recommendation function was still relatively short, and it maintained a relatively fast response speed even in situations with large amounts of data.

From Figure 15 (a), as the sample size increased, the coverage of the three recommendation systems showed an overall trend of improvement. This indicated that when there was more sample data available for analysis, these systems more accurately captured user interests and preferences, thereby providing a wider range of recommended content. In addition, the IIBCF scheme proposed in the study generally outperformed the Accu-NRecSys scheme and SPSRB scheme in terms of coverage, which may be due to the higher efficiency and accuracy of the IIBCF scheme in algorithm design or data processing. From Figure 15 (b), as the number of users increased, the IO response speed of the IIBCF scheme proposed in the study generally remained stable and unchanged. This indicated that the IIBCF scheme had good scalability and stability when handling a large number of user requests. By contrast, when the number of users reached 200, the IO response speed of theAccu-NRecSys scheme and SPSRB scheme showed a significant increase of over 50%. This may be due to the need for more complex calculations or data processing operations when processing user requests, which increases system load and slows IO response speed. In summary, the IIBCF scheme proposed in the study shows good performance in terms of coverage and IO response speed. This may be due to the optimization of algorithm design, data processing, and system architecture in this scheme, which can better adapt to large-scale user requests and data processing needs. Therefore, in practical applications, the IIBCF scheme may be a more advantageous and practical recommendation system scheme. In the experimental design, 40 university students from different majors and grades were recruited as participants to ensure the diversity and representativeness of the sample. Before the user study began, the study provided participants with a detailed introduction to the purpose and process of the experiment and obtained their informed consent. Subsequently, the study provided a simulated book recommendation system interface for each participant, which includes three main functions: new book recommendations, popular book recommendations, and personalized recommendations. Participants were required to interact with the system and experience each recommendation function. In the experience process, they needed to pay attention to the usability of the system, the accuracy of recommended content, the diversity of recommended results, and the IO response speed of the system. To quantify these indicators, a detailed survey questionnaire was designed, which included multiple questions related to system performance and user satisfaction.

Figure 16 shows the statistical results of 40 volunteers scoring the system's new book recommendations, popular book recommendations, personalized recommendations, and overall performance based on their reading preferences. From the graph, the IIBCF recommendation algorithm received high evaluations in all aspects. The average score of the personalized book recommendation function was as high as 95.8 points. Meanwhile, the average score of the newly added book function was 96.3 points, and finally, the score of the newly added book function was 97.6 points. The personalized book recommendation function performed particularly well in all evaluation items, with an average score of 95.8 points. This number was quite high, fully demonstrating the recognition and appreciation of the volunteers for this function. This feature can effectively provide users with book recommendations that match their personal reading preferences, reducing the effort and time they need to spend searching for books of interest. In addition, the newly added book function also received high praise,

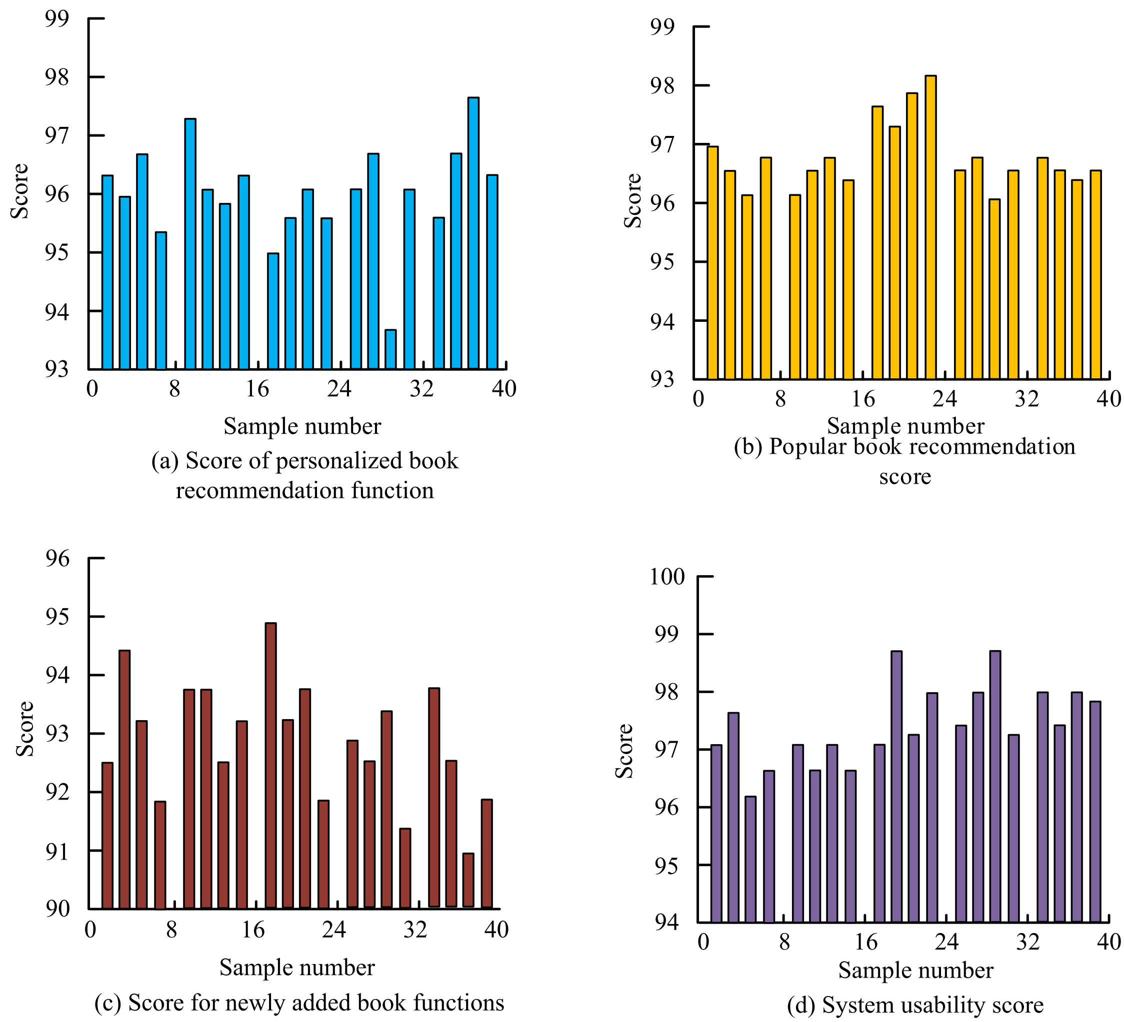


FIGURE 16. Score results of various aspects of the system.

with an average score of 96.5 points. This result indicated that this system can accurately grasp users' reading needs and recommend new books that meet their interests. At the same time, the score for the newly added book function was 97.6, which indicated that users had a high level of satisfaction with this feature of the system. This score fully proved that the system accurately and timely recommended new books to users, and the recommended quality was very high, which met users' reading needs. In summary, the IIBCF recommendation algorithm received high praise in all aspects. These evaluations fully demonstrated the superiority of the algorithm and the practical value of the system. From the perspective of personalized recommendation function and new book recommendation, users highly rated the system, proving its excellent performance in providing high-quality reading recommendations to users. Although user evaluation is one of the important indicators for evaluating system performance, it has a certain degree of subjectivity. Therefore, it is necessary to treat the evaluation results with caution and combine them with other objective evaluation methods and indicators to conduct a comprehensive and accurate evaluation.

V. RESULTS AND DISCUSSION

To provide readers with more accurate book recommendation results, this study designed a book recommendation model based on the IIBCF algorithm. The results showed that the system was not only popular among users but also particularly outstanding in the personalized book recommendation function, with an average score of 95.8 points. Behind this high score was the precise capture and satisfaction of user reading preferences by the system, which fully proved the effectiveness and superiority of the IIBCF algorithm in the field of personalized recommendation. Further comparison was made between the IIBCF scheme and the currently popular Accu-NRecSys scheme and SPSRB scheme. The IIBCF scheme showed significant advantages in both coverage and IO response speed, two key performance indicators. This advantage is mainly attributed to the innovative algorithm design of the IIBCF scheme, as well as the optimization of data processing and system architecture. These optimizations enable the IIBCF scheme to better respond to large-scale user requests, process massive book data, and provide more extensive and rapid book recommendation services. In addition, the system also demonstrates good scalability and stability. Whether in the context of a surge in user numbers or processing complex book data, the system can maintain efficient and stable operation, providing users with continuous and reliable book recommendation services. This achievement not only greatly enhances the reading experience of users, but also provides a new and efficient book recommendation solution for library and other book resource providers. In contrast, while the neural network recommendation models reviewed by L. Wu et al. show promise in general recommendation tasks, their specific performance in personalized book recommendations and large-scale data processing remains unclear. Moreover, the scalability and stability of these neural network models under different user loads and complex data scenarios may pose challenges [23]. Furthermore, this IIBCF-based system exhibits robust scalability and stability, maintaining efficient and reliable operation even during surges in user numbers or when processing complex book data. This ensures continuous and dependable book recommendation services for users, significantly enhancing their reading experience.

In summary, the study has successfully validated the excellent performance and practical value of a personalized book recommendation system based on the IIBCF algorithm. This system not only accurately meets the personalized reading needs of users, but also performs well in coverage, response speed, scalability, and stability, which is significantly better than similar research schemes. Therefore, the system can be applied and promoted in the future, bringing tangible benefits to more users and book resource providers. In addition, presenting content that users may be interested in through personalized algorithms may result in users only being exposed to information that aligns with their existing viewpoints and interests, thereby limiting the diversity and breadth of information they are exposed to. This is also one of the improvement directions for future research.

VI. CONCLUSION

The study successfully constructed an efficient book recommendation system by designing a new CF algorithm based on an IIBCF algorithm. The system performed particularly well in personalized recommendations, with an average score of 95.8 points. Compared with traditional Accu-NRecSys and SPSRB schemes, the proposed IIBCF scheme exhibited significant advantages in coverage and IO response speed, due to its innovative algorithm design and optimization of data processing and system architecture. However, research also realized that personalized recommendation algorithms may lead to user information silos, where users only come into contact with information that aligns with their existing viewpoints and interests, thereby limiting information diversity and breadth [25], [26]. To address this issue, future research should focus on increasing the diversity of recommended content while maintaining recommendation accuracy. In summary, this study validated the effectiveness and superiority of a personalized book recommendation system based on the IIBCF algorithm. This system not only met the personalized reading needs of users, but also performed excellently in coverage, response speed, scalability, and stability. Meanwhile, future research should continue to explore how to balance recommendation accuracy and content diversity to provide more comprehensive and valuable book recommendation services. This article mainly uses accuracy as an indicator to evaluate whether the results of the recommendation model meet the needs of users. However, there is a lack of evaluation of the recommendation results of the model from the perspectives of diversity and interpretability. In the future, it is possible to consider introducing diversity evaluation indicators, such as similarity between items in recommendation lists, popularity distribution, etc. In addition, it is possible to generate recommendation reasons that are easy for users to understand based on various factors such as user's historical behavior, item characteristics, and contextual information, in order to improve the interpretability of the model. Finally, methods such as weighted fusion and sorted fusion can be used to fuse the recommendation results of different algorithms and technologies to generate the final recommendation list.

REFERENCES

[1] M. U. Danjuma, B. Yusuf, and I. Yusuf, "Reliability, availability, maintainability, and dependability analysis of cold standby series-parallel system," *J. Comput. Cognit. Eng.*, vol. 1, no. 4, pp. 193–200, Apr. 2022.

[2] N. Shakeel, P. Teradata, and S. Shakeel, "Context-free word importance scores for attacking neural networks," *J. Comput. Cognit. Eng.*, vol. 1, no. 4, pp. 187–192, Sep. 2022.

[3] P. A. Ejegwa and J. M. Agbetayo, "Similarity-distance decision-making technique and its applications via intuitionistic fuzzy pairs," *J. Comput. Cognit. Eng.*, vol. 2, no. 1, pp. 68–74, Jan. 2022.

[4] S. Kataria and U. Batra, "Implementation of nearest co-cluster collaborative filtering using a novel similarity index," *Indian J. Sci. Technol.*, vol. 16, no. 29, pp. 2204–2216, Aug. 2023.

[5] C. Raja Kumar and V. Jayanthi, "A novel fuzzy rough sets theory based CF recommendation system," *Comput. Syst. Sci. Eng.*, vol. 34, no. 3, pp. 123–129, 2019.

[6] N. Kumar and P. A. R. Kumar, "STEM: Stacked ensemble model design for aggregation technique in group recommendation system," *Int. J. Bus. Intell. Data Mining*, vol. 21, no. 1, p. 66, 2022.

[7] D. Sivabalaselvamani, M. S. Vidhyasree, M. P. Pavithra, M. G. Soundarya, and M. M. Preethika, "Books and movies recommendation and rating prediction based on CF networks," *IJAST*, vol. 29, no. 5, pp. 705–714, Apr. 2020.

[8] X. Li, J. Grahl, and O. Hinz, "How do recommender systems lead to consumer purchases? A causal mediation analysis of a field experiment," *Inf. Syst. Res.*, vol. 33, no. 2, pp. 620–637, Jun. 2022.

[9] M. R. Ananda and C. B. Harahap, "Rancang bangun aplikasi pemesanan barbershop menggunakan metode IBCF berbasis Androidi," *Jurnal Info Digit*, vol. 1, no. 3, pp. 945–960, Sep. 2023.

[10] O. A. Montesinos-López, E. Franco-Pérez, F. J. Luna-Vázquez, J. Salinas-Ruiz, S. Sandoval-Carrillo, M. A. V. Jiménez, J. Cuervas, and P. C. Santana-Mancilla, "Benchmarking between item based collaborative filtering algorithm and genomic best linear unbiased prediction (GBLUP) model in terms of prediction accuracy for wheat and maize," *Biotecnia*, vol. 22, no. 2, pp. 136–146, Aug. 2020.

[11] J. Kang and H. Lim, "Proposal of content recommend system on insurance company web site using CF," *J. Digit. Converg.*, vol. 17, no. 11, pp. 201–206, Nov. 2019.

[12] J. Xiao and O. Bulut, "Item selection with collaborative filtering in on-the-fly multistage adaptive testing," *Appl. Psychol. Meas.*, vol. 46, no. 8, pp. 690–704, Aug. 2022.

[13] Z. Zhang, M. Dong, K. Ota, and Y. Kudo, "Alleviating new user cold-start in user-based collaborative filtering via bipartite network," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 3, pp. 672–685, Jun. 2020.

[14] S. Poudel and M. Bikdash, "Optimal dependence of performance and efficiency of collaborative filtering on random stratified subsampling," *Big Data Mining Anal.*, vol. 5, no. 3, pp. 192–205, Sep. 2022.

[15] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, and C. Valentino, "A content-based recommendation approach based on singular value decomposition," *Connection Sci.*, vol. 34, no. 1, pp. 2158–2176, Dec. 2022.

[16] P.-Y. Hsu, J.-Y. Chung, and Y.-C. Liu, "Using the beta distribution technique to detect attacked items from collaborative filtering," *Intell. Data Anal.*, vol. 25, no. 1, pp. 121–137, Jan. 2021.

[17] L. Badis, M. Amad, D. Aïssani, and S. Abbar, "P2PCF: A collaborative filtering based recommender system for peer to peer social networks," *J. High Speed Netw.*, vol. 27, no. 1, pp. 13–31, Mar. 2021.

[18] W. Liang, S. Xie, J. Cai, J. Xu, Y. Hu, Y. Xu, and M. Qiu, "Deep neural network security collaborative filtering scheme for service recommendation in intelligent cyber–physical systems," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22123–22132, Nov. 2022.

[19] F. Wang, H. Zhu, G. Srivastava, S. Li, and L. Qi, "Robust CF recommendation with user-item-trust records," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 4, pp. 986–996, Mar. 2021.

[20] D. Dang, C. Chen, W. Yu, and H. Hu, "A semantic-aware collaborative filtering recommendation method for emergency plans in response to meteorological hazards," *Intell. Data Anal.*, vol. 24, no. 3, pp. 705–721, May 2020.

[21] J. Redelinghuys and Ş. Er, "SoMLier: A South African wine recommender system," *Journal Wine Research*, vol. 34, no. 1, pp. 54–80, 2023.

[22] X. Fu, Y. Liao, S. Cheng, R. Deng, and Z. Yang, "Stable isotope-labeled precursor tracing reveals that L-alanine is converted to L-theanine via L-glutamate not ethylamine in tea plants in vivo," *J. Agricult. Food Chem.*, vol. 69, no. 50, pp. 15354–15361, 2021.

[23] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4425–4445, Jan. 2022.

[24] M. Bourreau and G. Gaudin, "Streaming platform and strategic recommendation bias," *J. Econ. Manag. Strategy*, vol. 31, no. 1, pp. 25–47, 2021.

[25] Z. Zhang, Y. Zhang, M. Dong, K. Ota, Y. Zhang, and Y. Ren, "Collaborative tag-aware graph neural network for long-tail service recommendation," *IEEE Trans. Services Comput.*, early access, pp. 1–15, 2024, doi: 10.1109/tsc.2024.3349853.

[26] Q. Huang, "Mathematical expression recommendation model based on genetic algorithm," in *Proc. IEEE 4th Annu. Flagship India Council Int. Subsections Conf. (INDISCON)*, Aug. 2023, pp. 1–5, doi: 10.1109/INDISCON58499.2023.10270013.

NA LIN was born in Jilin, China, in 1979. She received the bachelor's degree in information management and information systems from Changchun University of Technology, in 2003, and the master's degree in plant nutrition from Jilin Agricultural University, in 2016. She is currently an Associate Research Librarian with the Library, Jilin Agricultural University. She has led or participated in more than ten scientific research projects. She has also published more than 20 academic articles. Her research interests include library and information science, library management, big data analysis, and literature retrieval.