Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **4**, November **2025**

Intelligent Information Management of University Books Based on Clustering and Extreme Learning Machine

Tao Xue*

Library of Qinghai University
Xining 810016, P. R. China
heidaxuetao@163.com

Sang-Yu Tsai

Central Philippine University
Iloilo 50003, Philippines
fz8035@163.com

*Corresponding author: Tao Xue

Received August 8, 2024, revised December 12, 2024, accepted June 27, 2025.

Abstract. *This article proposes an intelligent information management system for university books based on clustering and Extreme Learning Machine (ELM) to address the issues of low efficiency and resource waste faced in the management of university libraries. Firstly, through clustering analysis of book data, the system can automatically identify and classify books of different categories, simplifying the book management process and improving management efficiency. Firstly, in response to the issues of long-time consumption and low classification accuracy in current book automatic classification methods such as Support Vector Machine and Feedforward Neural Network, research was conducted to introduce Extreme Learning Machine (ELM) into the field of book classification, and a Chinese book automatic classification model based on ELM was proposed. This model takes the mixed features of books as the training object, and utilizes the ELM algorithm to compute the hidden layer output matrix and neural network weights, thereby obtaining the classification model. Secondly, a university book online recommendation algorithm based on ISODATA clustering is introduced, capable of adjusting clustering accuracy and enhancing user satisfaction. Experimental results demonstrate that this method surpasses traditional approaches in classification accuracy, prediction performance, and recommendation quality, significantly improving the management level and user experience of university libraries.*

Keywords: book classification; extreme learning machine; book recommendations; clustering

1. **Introduction.** With the emergence of the knowledge economy era, the circulation of books in the book publishing industry is constantly increasing, and the demand for university book procurement is also growing rapidly [1], resulting in the collection of tens of thousands of books in university libraries, with a huge quantity. The manual classification of book cataloging in university libraries is time-consuming and laborious. If book classification is done manually, it will not only be complex, costly, and inefficient, but also the accuracy cannot be absolutely guaranteed [2].

With the continuous development of artificial intelligence, using computer-based machine learning technology to achieve various automation tasks has become the mainstream and trend in various fields [3], significantly enhancing the efficiency of conventional manual processes. The use of advanced artificial intelligence technology to solve the informatization and classification automation of library resources has become a research focus and primary task of library expert system technology.

The collection of books in university libraries is growing continuously, and the traditional method of searching for literature through catalog systems can no longer meet the urgent need of users to accurately retrieve books of their interest [4]. Therefore, building a book recommendation system that meets user needs is of great significance for users to quickly search and make accurate choices. Generally speaking, in addition to classification and labeling, books are related to other books in terms of content and references, and recommendation systems mainly search and recommend based on this information.

The book recommendation system based on book title, author, publisher, and book theme still adopts a basic grammar-based retrieval method, which leads to information accumulation due to the need to include a large amount of index information. The problem of information accumulation can have an impact on the timeliness and quality of recommended books, leading to user errors in book selection. A good recommendation system should not only simplify user operations, but also enable users to work more effectively. The book recommendation system should minimize data output, allowing users to find the most suitable books in a very brief time based on their needs, enhancing user experience and resource utilization.

1.1. **Related Work.** At present, many classification research results based on various machine learning algorithms have been proposed in the field of Chinese text classification. Common text classification methods include Naive Bayes, Class Center Vector, K-Nearest Neighbor, and Support Vector Machine. He *et al.* [5] discussed an enhanced TF-IDF algorithm-based text classification method, which overcomes the disadvantage of the VSM that cannot effectively adjust weights, and uses Bayesian classification to enhance text classification accuracy. Liu *et al.* [6] introduced a patent text classification method using probabilistic hypergraph semi-supervised learning, which adopts a K-nearest neighbor (KNN) strategy to enhance classification accuracy and decrease the number of needed training samples.

In addition to the above machine learning algorithms, as a type of Extreme Learning Machine (ELM) utilizing a single hidden layer feedforward neural network structure [7, 8], it offers rapid solving speed and strong generalization, demonstrating excellent performance in tasks like classification, regression, clustering, and feature learning [9, 10].

Regarding book recommendation algorithms, scholars both domestically and internationally have conducted relevant research. The commonly used algorithms for recommendation systems include content-based recommendation, association rules, multi-model integration, and collaborative filtering algorithms. Content-based recommendation methods utilize prior algorithms to search for associations and dependencies between rules. The typical feature of recommendation systems based on multi-model ensemble algorithms is the multi-classifier structure, which requires the execution of two different layers. In the first layer, several basic classifiers are trained; in the second layer, ensemble methods such as XGBoost or AdaBoost are used to combine the basic classifiers. The collaborative filtering algorithm searches a large number of users and detects a small portion of users who have similar interests in a certain attribute. Among them, similarity measurement is an important component of collaborative filtering, which can identify the user set that selects the attribute.

Cui *et al.* [11] discussed the importance of ontology-based personalized book recommendation and collaborative filtering algorithm recommendation, and introduced the main methods and technologies of collaborative filtering recommendation systems. Phorasim and Yu [12] developed a movie recommendation system, and proposed a new method for recommending to current users based on user clustering. With the increase of library collections and the development of the Internet industry, the recommendation accuracy or retrieval efficiency of these methods can no longer satisfy the demands of accurate recommendation of libraries. Collaborative filtering algorithm is a common book recommendation technique, with an accuracy rate of 88% in research [13], which is relatively low. However, when adding the Jaccard similarity coefficient in collaborative filtering, its recall rate can reach a relatively high level. Content filtering methods typically recommend based on the similarity between articles. Most recommendation systems are driven by big data and can provide recommendations based on search popularity, relevance, and book content. Data sparsity is the main problem of traditional book recommendation systems, and the use of neural network-based personal ranking algorithms can solve this problem. In addition, some context-aware rule-based techniques, as well as their latest pattern analysis [14, 15], classification-based, or rule-based belief prediction techniques, can be used to construct recommendation systems.

1.2. **Contribution.** This article addresses the issues of lengthy training times and poor classification results in existing book classification models. It introduces a new machine learning algorithm, Extreme Learning Machine (ELM), into book classification and proposes an ELM-based model. ELM, a learning algorithm for single hidden layer feedforward neural networks, eliminates the need to adjust hidden element bias and network input weights during training, requiring only the setting of the number of hidden nodes. Compared to traditional classification algorithms, it simplifies parameter selection, accelerates training speed, and improves generalization.

ELM, as a single hidden layer feedforward neural network, offers fast speed and good generalization by randomly setting the input weights of hidden elements and networks without requiring adjustments during training. It can effectively overcome the shortcomings of commonly used algorithms in book classification, such as SVM and BP networks. Thus, this article introduces an ELM-based classification model that extracts mixed book features using LDA and TF-IDF algorithms, and then trains the book feature information using the ELM algorithm to obtain a classifier. In the training process, ELM allows random setting of network input parameters, while the activation function and number of hidden nodes must be manually chosen. This article conducts extensive experiments to determine the optimal number of nodes and activation function. Additionally, to evaluate the advantages and disadvantages of the ELM-based classification model against traditional models, classifiers were trained using SVM and BP neural networks, and the classification performance of all three models was compared using the same dataset.

Secondly, this study introduces a precise recommendation method based on the ISODATA clustering algorithm, which incorporates splitting and merging operations into the k-means algorithm, and is extensively used in text mining and behavior analysis. This algorithm enhances the flexibility of recommendation system classification by adaptively determining the number of categories, particularly when dealing with large datasets where manual categorization is impractical. Moreover, unlike traditional recommendation algorithms, it leverages both user information and preferences, thereby improving the accuracy and efficiency of recommendations.

2. Theoretical Analysis.

2.1. **Overview of Automatic Classification of Chinese Books.** The book classification number is crucial in book retrieval and inquiry, but as mentioned earlier, the traditional manual classification method is no longer suitable for the increasing number of book publications. It is an urgent problem to replace manual work with machines to complete this task.

Nowadays, the classification numbers of Chinese books are generated based on the Chinese Library Classification System (referred to as Zhongtu Number). Zhongtu Number is a notable large-scale classification system created and published after the establishment of the People's Republic of China, which uses a mixed number of letters and numbers and adopts a layered numbering system. It divides all knowledge systems into 22 categories, from A (Marx, Lenin, Mao Zedong, Deng Xiaoping Theory) to Z (Comprehensive Books), and each category is further subdivided into several subcategories, such as A1 (Marx, Engels' works) and A2 (Lenin's works) under category A.

There are many categories and complex levels in the Chinese Library Classification, and generally speaking, the deeper the level, the fewer books there are. The premise of machine learning is to have a certain amount of data support; for deep-level categories such as TP311 (program design, software programming), because of the limited number of book samples, the performance of classifiers trained using machine learning is unsatisfactory.

2.2. **Feature Extraction.** Feature extraction is a prerequisite for book classification, and its main task is to quantify the feature words in book titles, abstracts, catalogs, and other information to represent book information. Due to the fact that book titles or abstracts are unstructured textual information that computers cannot recognize or process, it is necessary to scientifically abstract these texts and establish mathematical models to describe and replace them. This article mainly focuses on the TF-IDF model and LDA Topic Model to extract feature information of books.

LDA is a document generation model that assumes that an article contains several topics, each of which contains several words. It is widely used in fields such as text feature extraction, article topic extraction, and sentiment analysis. The steps for constructing an article using the LDA model are as follows: first, choose a topic with a certain probability, then pick a word from this topic with a certain probability, and repeat this process until the entire article is created [16, 17]. The joint probability of LDA is given by Equation (3).

$$p(\theta, z, w, \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{V} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{1}$$

where α and β represent parameters at the corpus level, which are the same for each document and only need to be sampled once during the generation process. θ serves as a variable at the document level, which is different for each document. Throughout the entire generation process, each document is sampled only once; z is generated by θ, and w is generated jointly by z and β, all of which are variables at the word level. A word w corresponds to a topic z.

The LDA topic model can be used to obtain the "document–topic" distribution and "topic–word" distribution of an article, which have wide applications in many fields. In the field of machine learning, topics in the "document–topic" distribution are used as document feature words for machine learning algorithms to complete classification or prediction tasks. In the recommendation field, the distribution of items on topics is calculated, and then the similarity between the distributions of two items on topics is calculated to provide recommendations to users. Overall, using LDA topic models to extract text topics as features is more effective in uncovering the core ideas of the text compared to TF-IDF or VSM models, and has lower dimensionality. However, LDA calculation is more complicated, time-consuming, and requires that the length of the text should not be too short.

2.3. **Classification Model.** Classification has long been a key research focus in machine learning, with many researchers contributing classic algorithms such as decision trees, support vector machines, neural networks, and naive Bayes. Below, we will briefly explain several commonly used classification algorithms and explore their advantages and disadvantages in the field of book classification.

2.3.1. *Support Vector Machine.* The simplest idea for the classification task, given a training sample set of $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, is to search for a hyperplane partition in the sample space. The partitioning of hyperplanes in the sample space is described as follows:

$$\omega^T x + b = 0 \tag{2}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_d)$ represents the normal vector of the plane, b represents the displacement, and obviously ω and b determine the position of the plane. The distance from any point in space to a plane can be expressed as:

$$r = \frac{|\omega^T x + b = 0|}{\|\omega\|} \tag{3}$$

If the hyperplane can perfectly partition the sample space, then let:

$$\omega^T x_i + b \geq +1, \quad y_i = +1 \tag{4}$$

$$\omega^T x_i + b \leq -1, \quad y_i = -1 \tag{5}$$

Obviously, finding a hyperplane that perfectly partitions the sample space and has good generalization is equivalent to finding a partition hyperplane with the "maximum interval". That is, finding reasonable parameters ω and b that maximize γ while satisfying the constraint conditions, that is:

$$\max_{\omega,b} \frac{2}{\|\omega\|} \tag{6}$$

The above formula can also be written as:

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2 \tag{7}$$

The primary goal of SVM is to identify the optimal hyperplane for partitioning feature space, focusing on maximizing the classification margin. Support vectors play a key role in SVM classification decisions, meaning that the computational complexity is not related to the dimensionality of the sample space, but depends on the number of support vectors, which can to some extent avoid the "curse of dimensionality" [18, 19]. However, SVM uses quadratic programming to solve for support vectors, requiring the calculation of m-order matrices during the operation, which consumes a lot of memory and time when facing large-scale training samples. In theory, SVM can only solve binary classification problems, and to apply it to practical multi-classification problems, it requires strategies to construct multiple binary support vectors while combining the advantages of other algorithms to improve multi-classification accuracy [20].

2.3.2. *Extreme Learning Machine.* The extreme learning machine (ELM), a single hidden layer neural network algorithm, can be described by a linear equation as: $H\beta = T$ where

$$H = \begin{bmatrix} g(a_1 \cdot x_1 + b_1) & \cdots & g(a_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(a_1 \cdot x_N + b_1) & \cdots & g(a_L \cdot x_N + b_L) \end{bmatrix}, \tag{8}$$

$$\beta = [\beta_1, \beta_2, \ldots, \beta_L]^T \in \mathbb{R}^{L \times d}, \tag{9}$$

$$T = [t_1, t_2, \ldots, t_n]^T \in \mathbb{R}^N \tag{10}$$

If the activation function $g(x)$ is infinitely differentiable, the input weight a of the neural network and the bias b of the hidden layer nodes can be set randomly without adjustment during training. For randomly initialized a and b, training a single hidden layer neural network involves finding the least squares solution of the linear system $H\boldsymbol{\beta} = T$. When the number of hidden nodes L equals the number of input samples N, with randomly set a and b, H becomes a reversible square matrix, allowing the network to fit the training samples with zero error.

However, usually, the number of training samples exceeds the number of hidden nodes, making H an $N \times L$ matrix, thus requiring the calculation of the Moore–Penrose pseudoinverse:

$$\hat{\boldsymbol{\beta}} = H^{+}T = (H^T H)^{-1} H^T T \tag{11}$$

Unlike traditional single hidden layer neural networks, the ELM algorithm does not require adjusting the input weight a and hidden layer bias b during training. It only needs to adjust the value of P according to the corresponding algorithm to achieve a global optimal solution, greatly enhancing the training speed.

2.4. **Introduction to clustering algorithms.** A clustering algorithm is an unsupervised learning method that partitions objects in a dataset into groups or clusters based on similarity or distance. The aim is to maximize similarity within clusters and minimize similarity between clusters for effective data grouping. Common clustering can be classified based on partitioning, density, and hierarchy.

(1) Partition based clustering algorithm is a clustering algorithm that divides a dataset into disjoint subsets. The basic idea is to divide the dataset into several clusters, and then iteratively partition these clusters to obtain the clustering results. A typical algorithm such as K-means [21] includes the main steps of initializing the centroid, calculating the distance between sample points and the centroid, assigning sample points, and recalculating the centroid. Repeat the above steps until a certain stopping condition is met, such as achieving the iteration count or the variance of the samples within the cluster no longer changing.

(2) Density based clustering algorithm is an algorithm that clusters data sets based on their spatial density. This algorithm defines a density threshold, considers regions in space with a density greater than the threshold as clusters, and merges sample points within the clusters into the same cluster. This algorithm can recognize clusters of any shape and has good performance in processing noisy data. However, this algorithm poses certain difficulties for datasets with large density variations, and requires adjusting density thresholds and other parameters to achieve optimal clustering results. The common DBSCAN [22] clustering algorithm is density based clustering algorithm.

(3) Hierarchical clustering algorithm is a bottom-up or top-down clustering method, whose main idea is to merge or separate similar or dissimilar samples by calculating the distance between them, gradually forming a hierarchical clustering tree. Clustering methods can be classified into cohesive hierarchical clustering and split hierarchical clustering. The former uses a bottom-up approach to gradually merge individual samples into larger clusters, while the latter uses a top-down approach to gradually subdivide the dataset into smaller clusters. The clustering strategy can be analyzed based on the specific scenarios in which the algorithm is used.

3. Intelligent information management method for university books based on Clustering and extreme learning machine.

3.1. **Automatic book classification based on extreme learning machine.** The purpose of this section is to construct a hierarchical classifier for the middle graph method, using a mixed feature model and an extreme learning machine algorithm. Unlike typical single-layer classifiers, hierarchical classifiers place greater emphasis on the hierarchical relationships between categories. They decompose a classification task into several smaller layers and construct classifiers in each layer. This mode can reduce computational complexity and is suitable for situations with multiple hierarchical categories. The basic steps are as follows:

(1) Data preprocessing: A word segmentation system is used to segment the bibliographic information from the training and testing sets, and remove stop words and special characters to obtain a word document distribution matrix.

(2) The LDA model is used to train the word distribution matrix for abstract information, obtain the corresponding topic document distribution matrix, and determine the optimal number of topics through multiple training comparisons. The TF-IDF model is used to weight the word distribution matrix for book title information, and the two are combined into a mixed matrix.

(3) Train the mixture matrix of the training set obtained in the previous step using the ELM algorithm, construct a text classifier, and obtain a classification model.

(4) According to the classification model, predict the category of the test set.

(5) Repeat the above steps to obtain classifiers for the second and third level categories respectively, and test the classification performance of the corresponding test set on the classifiers.

Extreme learning machines are extensively used in fields like power monitoring and image recognition, weather prediction, etc. due to their fast-training speed, good classification performance, and strong generalization ability. The core of extreme learning machine is to find the least squares solution of linear system $H\beta = T$. Therefore, the book classification algorithm based on extreme learning machine is described as Algorithm 1:

Algorithm 1 The book classification algorithm based on ELM

Input: $TrainSetTr = \{T_{r1}, T_{r2}, \ldots, T_{rm}\}$; $TestSetTr = \{T_{r1}, T_{r2}, \ldots, T_{rn}\}$; L; $g(x)$.
++ **Output:** Classification mode TM; Training and testing time;

1: Loading training and testing sets;
2: Randomly generate hidden layer node parameters;
3: Calculate the hidden layer output matrix H;
4: Calculate output weights β;
5: Obtain the classification mode TM of Chinese books through $H^{+}\beta$;
6: By comparing the training and testing sets, we obtained the accuracy and time metrics for both training and testing.

3.2. **Recommendation algorithm based on ISODATA clustering.** This section investigates a book recommendation algorithm based on ISODATA clustering. It includes three parts: data acquisition, preprocessing, and recommendation algorithms. The dataset is sourced from Kaggle's Goodreads Books knowledge base. Kaggle's Goodreads books repository contains seven datasets, of which four datasets (`Books.csv`, `Book_tags.csv`, `Ratings.csv`, and `Max_Rating.csv`) were used according to algorithm requirements. Firstly, preprocessing techniques were applied to merge all datasets. Then, the dataset was denoised and vectorized to form a new dataset for analysis. Finally, clustering algorithms were applied to accurately recommend books of the specified category to users.

In recent years, recommendation algorithms based on k-means have developed rapidly, which can improve data sparsity and algorithm scalability issues and have high recommendation accuracy. This study proposes an improved k-means algorithm, namely the ISODATA algorithm, for clustering-based recommendation. Compared with k-means, the significant advantage of this algorithm is that it can dynamically adjust the number of book categories k according to the actual situation. When the dataset is large, it is difficult to manually determine the number of categories. In this case, the ISODATA algorithm automatically adjusts the number of categories through splitting and merging operations, enhancing the flexibility of the recommendation system.

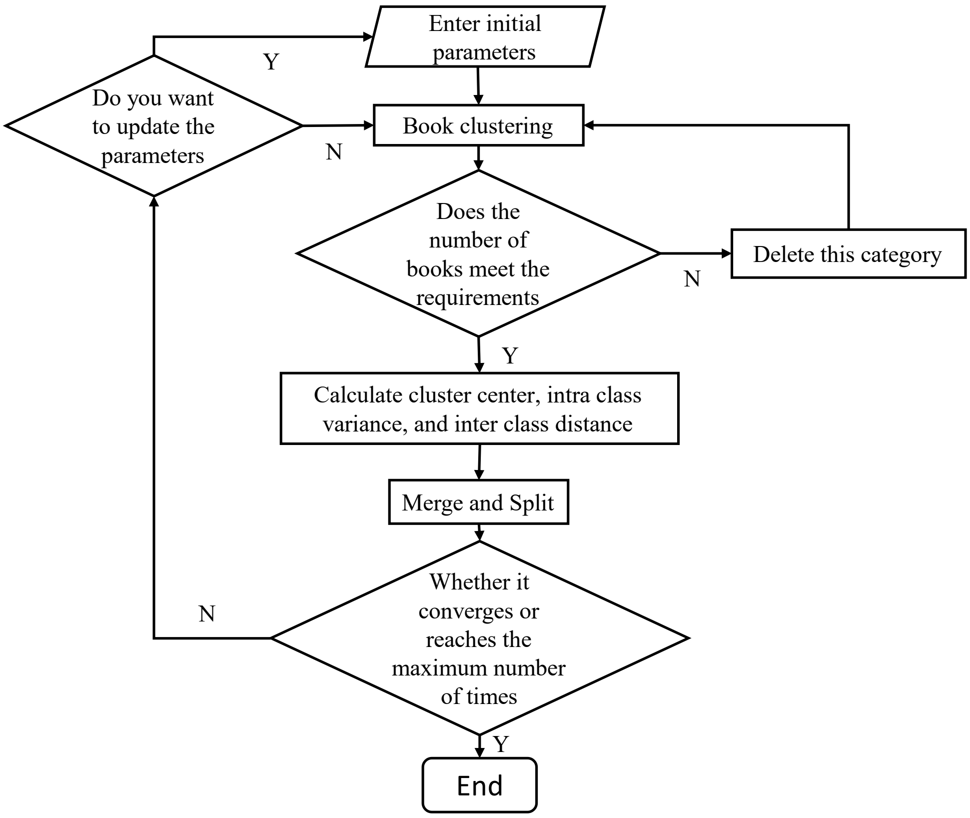

Figure 1. ISODATA algorithm flowchart.

Firstly, select Ko's book as the initial cluster center in the dataset, denoted as $C = \{C_1, C_2, C_3, \ldots, C_k\}$, calculate the distance from each book a_i to each cluster center in the dataset, and change its label to the category corresponding to the nearest cluster center, such as "*Sense and Sensibility*" being labeled as a "romantic" book and "*Treasure Island*" being labeled as an "adventure" book. If the number of books in a certain category is less than the set threshold $N_{\min}$, the category will be deleted, that is, $K = K - 1$, and the category of each book in the dataset will be recalculated. After temporarily determining the number of categories, recalculate the cluster centers for each category, which are the centroids of all books in that category:

$$c_i = \frac{1}{|c_i|} \sum_{a \in c_i} a \tag{12}$$

After obtaining the new cluster center, it is necessary to determine the size of the K value of the current number of book categories. If $K < K_0/2$, it is necessary to determine whether to perform a category splitting operation: first, calculate the intra-class variance of all book categories, traverse each category, and if the following conditions are met, perform a splitting operation on it:

$$\begin{cases} \sigma_i > \Sigma \\ n_i > 2N_{\min} \end{cases} \tag{13}$$

where Σ is the intra-class variance threshold, determined based on the overall dispersion of the book dataset; n_i is the number of samples in class i.

The splitting operation divides the categories that meet the conditions into two subcategories, such as splitting the "horror category" into "adventure category" and "suspense category". At this time, $K = K + 1$, and the clustering centers m_{i1} and m_{i2} are:

$$\begin{cases} m_{i1} = m_i + \sigma_i, \\ m_{i2} = m_i - \sigma_i, \end{cases} \tag{14}$$

where m_i is the cluster center of the split category.

If $K > 2K_0$, it is necessary to determine whether to perform a merge operation: first, calculate the inter-class distance $d(i, j)$ of all book categories, where $d(i, j) = 0$, traverse each category pair, and if the following conditions are met, perform a merge operation on them:

$$d(i, j) < D, \quad (i \neq j) \tag{15}$$

where D is the inter-class distance threshold.

The merge operation is to merge the categories of two books that meet the conditions. Its effect is opposite to the split operation. At this time, $K = K - 1$, and the new cluster center m_{ij} can be represented as:

$$m_{ij} = \frac{1}{n_i + n_j}(n_i m_i + n_j m_j), \tag{16}$$

where n_i and m_i respectively represent the number of books and the centroid of the i-th and j-th merged categories.

4. **Experiment.**

4.1. **Data set.**

The dataset used in this experiment is the Goodreads book dataset repository. This dataset stores rating data for 10,000 best-selling books. This dataset consists of seven tables, namely `Books.csv`, `Generators.csv`, `Book_tags.csv`, `Max_rating.csv`, `Ratings.csv`, `Read.csv`, and `Tags.csv`.

4.2. **Evaluating indicator.**

This article selects two evaluation metrics for recommendation algorithms to evaluate the performance of the ISODATA algorithm, namely the contour coefficient and Rand index. These two metrics primarily assess the clustering performance of the dataset, with higher values indicating better clustering results.

The prediction accuracy of book recommendation systems is also an important indicator for evaluation. The accuracy metric performs poorly on imbalanced datasets; for example, when a user has 10 books they are interested in and 90 books they are not interested in, even if the classifier labels them all as not interested, there is still a 90% accuracy rate. Therefore, the receiver operating characteristic (ROC) curve is used to evaluate the predictive accuracy of the book recommendation system. The ROC curve [19] is commonly used in machine learning and deep learning to visually summarize classifier accuracy. Unlike the P–R curve, the ROC curve's shape remains relatively stable even when the number of positive and negative samples varies. This stability allows the ROC curve to minimize interference from different test sets and objectively measure model performance.

The prediction problem in this study can be divided into two categories: the books that users are interested in are referred to as positive examples; if the user is not interested, call it a negative example. Therefore, there are four possible types of classification instances. A correctly classified positive case is a True Positive (TP), while an incorrectly classified one is a False Negative (FN). A correctly classified negative case is a True Negative (TN), and an incorrectly classified one is a False Positive (FP). Based on the above classification, evaluation indicators such as Precision (P), FP rate (FPR), Recall rate (R), TP rate (TPR), and $F1_{score}$ can be defined, which are expressed as follows:

$$P = \frac{TP}{TP + FP} \tag{17}$$

$$FPR = \frac{FP}{TN + FP} \tag{18}$$

$$TPR = \frac{TP}{TP + FN} \tag{19}$$

$$F1_{score} = \frac{2 \times P \times R}{P + R} \tag{20}$$

where R measures the proportion of recommended books that users are interested in, while the FPR calculates the proportion of recommended books that users are not interested in. The $F1_{score}$ calculates the harmonic mean of recall and precision. The maximum value for $F1_{score}$ can be 1.

4.3. **Experimental results and analysis.** The confusion matrix, a common tool for evaluating classifier performance, is detailed in Table 1 for the test results. The test contains 1000 samples, with 400 positive cases and 600 negative cases, respectively. Using the proposed extreme learning machine and clustering algorithm, 793 samples were correctly classified and 207 samples were misclassified, generating a confusion matrix as shown in Table 1. From the classification results, this algorithm has good classification accuracy.

Table 1. Test result.

Predict category	**Positive sample**	**Negative sample**	**Total**
True	302	109	411
False	98	491	589
Total	400	600	1000

At the same time, experimental verification was conducted on the book recommendation situation for real users. The dataset was divided into ten small datasets containing 1000 samples and tested on ten different users. The indicators TPR, FPR, and $F1_{score}$ were calculated for each user. Table **??** shows the experimental results. It can be seen that User 1 has a higher TPR than other users, which means that it has the highest probability of recommending a book list that interests the user; User 3's FPR is 33%, indicating it can better help users discover and eliminate books they are not interested in. The $F1$ score is more useful than accuracy as it balances precision and recall.

Due to the algorithm used by the research institute focusing on the classification results during classification, it is difficult to calculate the classification confidence and directly plot the ROC curve. Thus, a TPR–FPR distribution chart was created based on the ROC curve definition to visually present the test results, as shown in Figure 2. The chart

Table 2. Experimental result.

Test Number	**TPR**	**FPR**	$F1_{score}$
1	0.66	0.34	0.608
2	0.51	0.42	0.485
3	0.65	0.33	0.605
4	0.57	0.41	0.513
5	0.58	0.35	0.538
6	0.52	0.40	0.543
7	0.56	0.40	0.575
8	0.63	0.36	0.607
9	0.60	0.36	0.587
10	0.58	0.38	0.565
Average Value	**0.587**	**0.377**	**0.5625**

shows that the TPR–FPR distribution is closer to the upper left corner, indicating good algorithm performance.

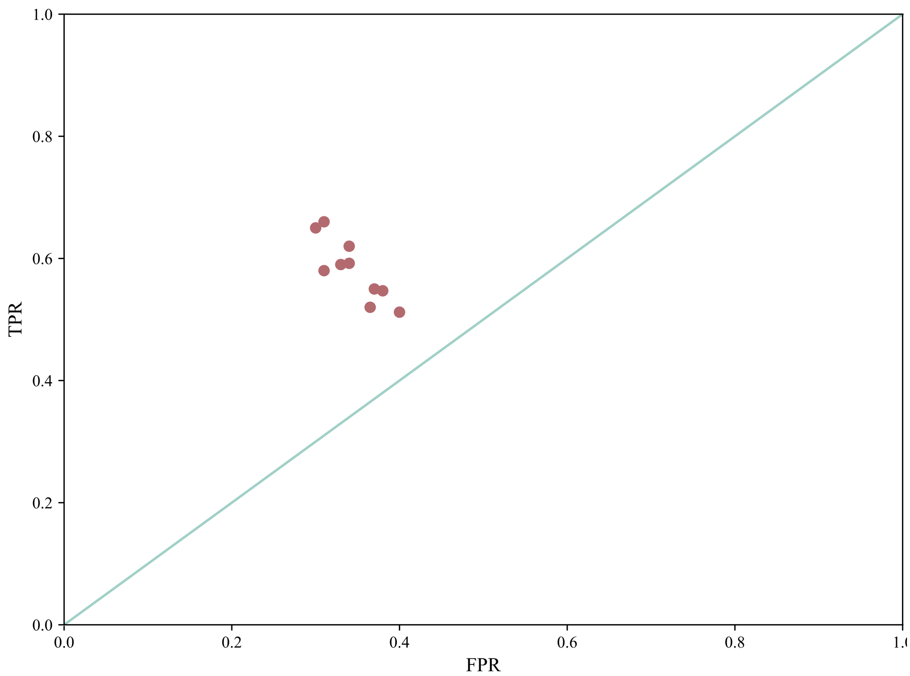

Figure 2. TPR–FPR distribution.

This study compared the clustering performance of the proposed algorithm with the k-means algorithm across the entire dataset. Figure 3 shows the comparison of contour coefficients and Rand indices for both algorithms with $k = 8$. The results indicate that the proposed algorithm outperforms the k-means algorithm in both metrics for this dataset.

Finally, comparative experiments were conducted between the proposed algorithm and the k-means clustering algorithm on ten datasets, using $F1_{score}$ the same evaluation metrics. Figure 4 displays the results, showing that the proposed algorithm outperforms the k-means algorithm in all but the second and fourth datasets. This indicates that the book recommendation system based on the Extreme Learning Machine and ISODATA algorithm offers better accuracy compared to traditional recommendation systems.

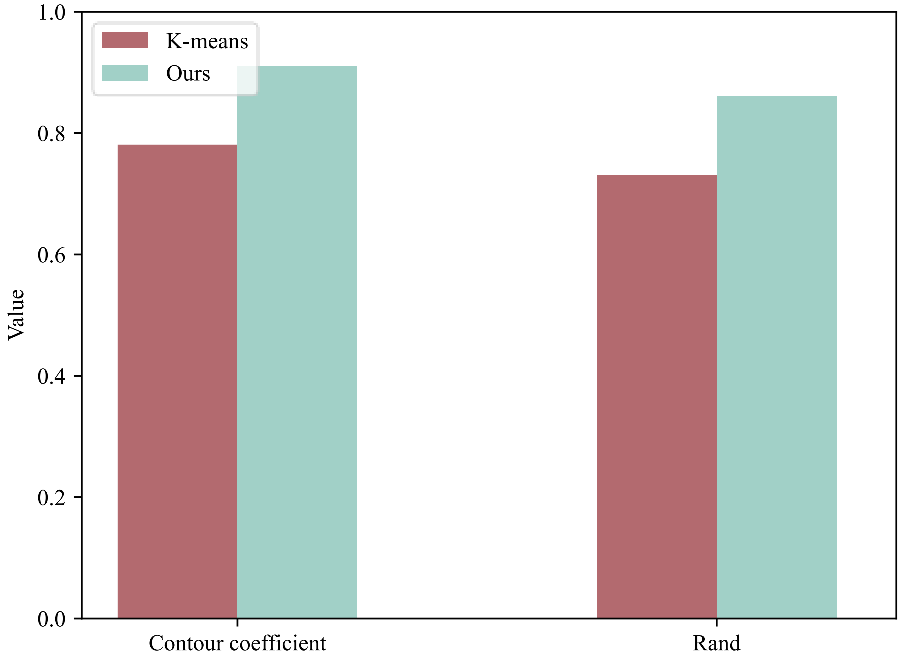

Figure 3. Comparison of clustering algorithms.

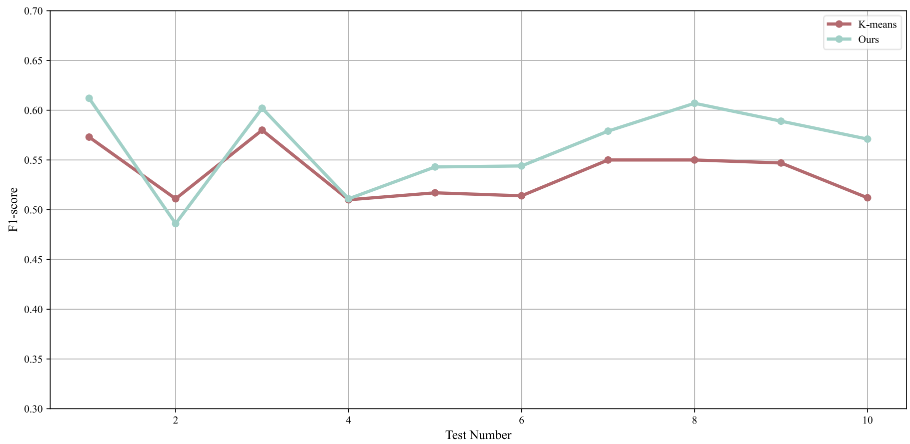

Figure 4. Comparison of recommendation results.

5. **Conclusion.** This article introduces an intelligent information management system for university books that utilizes clustering and ELM to address issues of inefficiency and resource waste in library management. First, for book classification, the proposed system employs the ELM algorithm to create a Chinese book classification model. This model uses mixed book features for training and generates an efficient classification model by computing the hidden layer output matrix and neural network weights. Experimental results demonstrate that this model significantly improves classification accuracy and processing speed compared to traditional methods, thereby enhancing library management efficiency. Second, for book recommendations, the article proposes an online recommendation algorithm based on ISODATA clustering. This algorithm adjusts clustering accuracy dynamically to improve recommendation precision and user satisfaction. By analyzing book data clusters, the system can automatically categorize and classify books, streamline management processes, and offer personalized recommendations. The results show that this approach surpasses traditional methods in classification accuracy, prediction performance, and recommendation quality, thereby enhancing library management and

user experience. Overall, the proposed system based on clustering and ELM significantly boosts management efficiency, optimizes resource allocation, and improves user satisfaction, supporting the modernization of university libraries.

REFERENCES

[1] S. Barsha, and S. A. Munshi, "Implementing artificial intelligence in library services: A review of current prospects and challenges of developing countries," *Library Hi Tech News*, vol. 41, no. 1, pp. 7–10, 2023.

[2] S. Shang, Y. Mi, L. Yang, K. Jiao, F. Sheng, Y. Shen, and H. Guo, "An overview of the development of university library in the information age," *Open Journal of Social Sciences*, vol. 5, no. 11, pp. 1–10, 2017.

[3] P. Warren, and D. Alsmeyer, "The digital library: a case study in intelligent content management," *Journal of Knowledge Management*, vol. 9, no. 5, pp. 28–39, 2005.

[4] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[5] K. He, Z. Zhu, and Y. Cheng, "A research on text classification method based on improved TF-IDF algorithm," *Journal of Guangdong University of Technology*, vol. 33, no. 5, pp. 49–53, 2016.

[6] G. Liu, M. Wang, and H. Liu, "Probabilistic Hypergraph Based Semi-supervised Learning Method for Patent Document Categorization," *Journal of Intelligence*, vol. 35, no. 9, pp. 187–191, 2016.

[7] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, "Statistical learning theory and ELM for big social data analysis," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 45–55, 2016.

[8] Y. Zhao, G. Wang, Y. Yin, Y. Li, and Z. Wang, "Improving ELM-based microarray data classification by diversified sequence features selection," *Neural Computing and Applications*, vol. 27, no. 1, pp. 155–166, 2016.

[9] A. Gazdar, and L. Hidri, "A new similarity measure for collaborative filtering based recommender systems," *Knowledge-Based Systems*, vol. 188, 105058, 2020.

[10] K. Tsuji, N. Takizawa, S. Sato, U. Ikeuchi, A. Ikeuchi, F. Yoshikane, and H. Itsumura, "Book recommendation based on library loan records and bibliographic information," *Procedia-Social and Behavioral Sciences*, vol. 147, pp. 478–486, 2014.

[11] L. Cui, C. Wang, and B. Yang, "Personalized book recommendation based on ontology and collaborative filtering algorithm," *The Open Cybernetics & Systemics Journal*, vol. 8, pp. 632–637, 2014.

[12] P. Phorasim, and L. Yu, "Movies recommendation system using collaborative filtering and k-means," *International Journal of Advanced Computer Research*, vol. 7, no. 29, p. 52, 2017.

[13] J. Das, S. Majumder, P. Gupta, and K. Mali, "Collaborative recommendations using hierarchical clustering based on Kd trees and quadtrees," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 27, no. 4, pp. 637–668, 2019.

[14] I. H. Sarker, A. Colman, and J. Han, "Recencyminer: mining recency-based personalized behavior from contextual smartphone data," *Journal of Big Data*, vol. 6, no. 1, pp. 1–21, 2019.

[15] Z. Yun-tao, G. Ling, and W. Yong-cheng, "An improved TF-IDF approach for text classification," *Journal of Zhejiang University-Science A*, vol. 6, no. 1, pp. 49–55, 2005.

[16] S.-W. Kim, and J.-M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, pp. 1–21, 2019.

[17] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, pp. 15169–15211, 2019.

[18] J. Cao, T. Xia, J. Li, Y. Zhang, and S. Tang, "A density-based method for adaptive LDA model selection," *Neurocomputing*, vol. 72, no. 7–9, pp. 1775–1781, 2009.

[19] M. A. Chandra, and S. Bedi, "Survey on SVM and their application in image classification," *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, 2021.

[20] M. E. Mavroforakis, and S. Theodoridis, "A geometric approach to support vector machine (SVM) classification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 671–682, 2006.

[21] K. P. Sinaga, and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[22] D. Birant, and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007.