International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 6708-6721

https://doi.org/10.70517/ijhsa463576

Construction of Personalized Information Recommendation System for Intelligent Libraries Based on Time Series Data Analysis

Mingjie Zhang[1,*]
[1] Chome-3-2 Kagamiyama, Higashihiroshima, Hiroshima, 739-0046, Japan
Corresponding authors: (e-mail: zhming9971@163.com).

Abstract The process of intelligent library construction is advancing, and readers' demand for personalized information services is growing. Traditional recommender systems do not fully consider the temporal characteristics of users' borrowing behavior, resulting in limited recommendation accuracy. Meanwhile, challenges such as user interest evolution and cold-start problem constrain the improvement of library service quality. OBJECTIVE: To construct an intelligent library personalized information recommendation system based on temporal data analysis in response to the problem of underutilization of temporal features in library personalized recommendation systems. Methods: Adopt convolutional neural network for local characterization of time-series data, and extract local features of user-item scoring matrix through normalization and similarity calculation; design personalized recommendation model based on BiLSTM, integrate Embedding layer and multilayer perceptron, and extract features of readers' borrowing preference using bidirectional long and short-term memory network; construct recommendation system with B/S architecture, adopt MVC three-layer architecture and J2EE technology to realize the system functions. RESULTS: The experiments show that in the sparsity interval of 0.7-0.9, the accuracy of CNN local similarity prediction is higher than the Euclidean distance and Pearson correlation coefficient methods; the BiLSTM model achieves the optimal performance at the learning rate of 0.001, 2-layer network, and batch size of 256, with the MAE value of 0.787; compared with the UserCF, ItemCF and ConvMF algorithms, the proposed algorithm performs optimally in 10 experiments. CONCLUSION: The recommender system based on temporal data analysis effectively improves book recommendation accuracy and provides technical support for personalized library services.

Index Terms Time-series data analysis, Intelligent library, Personalized recommendation, BiLSTM, Deep learning, Recommender system

I. Introduction

In contemporary society, information technology has penetrated into many aspects of people's lives, and our retrieval and utilization of information has become more convenient. But on the other hand, the information explosion also makes it easier for people to become information disoriented and be at a loss in the face of a large amount of information. Libraries, as information intermediaries, are making use of new technologies and methods to transform into intelligent libraries, gradually moving closer to the construction of more intelligent, automated and networked venues, and continuously enriching their collections with physical documentary resources and digital resource databases [1]-[3]. This has more choices for readers in terms of information, and at the same time, if readers are not provided with certain help, readers themselves may have problems in the retrieval, selection, and utilization of literature.

In the face of massive information resources, readers must be very clear about their own information needs in order to retrieve the required information more efficiently. However, in practice, a part of the readers tend to go out of interest in reading, the library is not very sure of the direction of the search, then in a large number of literature resources to match the most suitable for its reading literature will be more difficult [4]-[7]. With the development of information technology, in order to adapt to the development of the times, libraries not only need to build intelligent libraries, but also change the passive service mode of library readers who only serve them when they put forward information needs, and change from "books" as the work center to "readers" as the work center [8]-[11]. The popularization of various mobile devices and reading devices provides readers with more choices of information channels, and the information intermediary role of libraries may be weakened if they still only passively accept service requests from readers [12], [13]. Therefore, the realization of accurate resource recommendation for libraries has become an important research direction.

In recent years, driven by a new generation of information technology such as big data, cloud computing and artificial intelligence, personalized recommendation has become an important direction of library service innovation. Libraries in developed countries and regions have been carrying out personalized services for many years and have accumulated rich theoretical and practical experience [14]. Personalized recommendation is an important feature of intelligent libraries, reflecting the innovative integration of digital resources, big data analysis, personalized services and other elements [15]. Taking personalized recommendation as a breakthrough, libraries can systematically plan the construction of digital resources and optimize the resource structure. Introduce intelligent recommendation system to enrich the function of information system. Strengthen the construction of data analysis team and improve the level of data utilization. Innovate personalized service mode and improve service quality [16]-[18]. The implementation of personalized recommendation will become a powerful driving force for the systematic change of the library, promote the systematic reconstruction of management concepts, business processes, service methods and other aspects, and then realize the refinement of management based on big data, intelligent services and personalized marketing, and comprehensively enhance the core competitiveness of libraries [19]-[22]. In recent years, intelligent libraries have developed rapidly, and personalized information recommendation has become a research hotspot in the industry.

As an important carrier of knowledge dissemination and information service, the library's service mode is undergoing profound changes. The readers' demand for book resources presents a diversified and personalized trend, and the traditional passive service mode has been difficult to meet the users' needs. Personalized recommendation system through the analysis of the user's historical behavior data, the initiative to the user to push the book resources in line with their interests and preferences, and become an effective way to improve the quality of library services. The current library recommender system mainly adopts collaborative filtering, content recommendation and other methods, which realize personalized service to a certain extent, but there are still many limitations. One of the most prominent problems is that it ignores the temporal characteristics of users' borrowing behavior, and the user's historical borrowing records are simply regarded as a collection of static data, which fails to effectively capture the dynamic evolution of users' interests. In fact, user's reading interest has obvious time dependence, and the borrowing behavior in different periods reflects the trajectory of user's interest. In addition, traditional methods are ineffective in dealing with problems such as data sparsity and cold start, which affects the accuracy and coverage of recommendations. The development of deep learning technology provides new ideas to solve the above problems, especially the advantage of recurrent neural networks in time-series data modeling, which creates conditions for constructing a more accurate recommendation system.

This study is based on the demand for personalized library services, and designs and implements an intelligent recommendation system based on time-series data analysis. The study adopts convolutional neural network to locally characterize the user-item scoring matrix, and extracts the implied user preference features through data normalization and similarity calculation. On this basis, a deep learning recommendation model integrating BiLSTM, Embedding layer and multilayer perceptron is constructed, which makes full use of the advantages of bidirectional long and short-term memory networks in sequence modeling to effectively capture the backward and forward dependencies of readers' borrowing behaviors. The system implementation adopts the mature B/S architecture and J2EE technology framework to ensure the scalability and stability of the system. The effectiveness of the proposed method is verified through comparative experiments, which provides technical support and practical reference for personalized information service in libraries.

II. Library Recommender System Combined with Time Series Data Analysis

II. A. Local Characterization of Time Series Data

Time-series data analysis is basically handled with convolutional neural networks [23]. Not only that, but also can deal with the data into a one-dimensional vector, and then combined into a two-dimensional matrix to input, we have local features of the user's item ratings are made in the original user to the ratings ruling on the basis of this is different from the traditional model, and this characterization can make the CNN input data two-dimensional features are effectively maintained, and he can also capture the local features can be better access to the invisible preference features,. And because of the greater similarity of its elements, in the local meeting, it can optimize the performance of the intelligent library personalized recommendation system.

II. A. 1) Data normalization

Comparing the sparse fuzzy user item scores, the matrix scores are basically 1-5 or 1-10 which has a great impact on the CNN feature extraction. In order to reduce the impact, the floating point values of the scores among them are normalized to 1-0 and then used as inputs to the model in this chapter, the formula is shown in equation (1):

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Then, x_{norm} is viewed as normalized data, x is the initial raw data, $x_{\max}$, $x_{\min}$ represent the maximum and minimum values in the data respectively.

Local Characterization Local Characterization or definition means that two users who are close to each other, their similarity characterization is computed to form a multi-attribute element, $M_{ij} = R_i \oplus C_j$, where the local eigenvalue matrices are applied to the M_{ij} to represent the i rows and j columns, respectively. , and, the row vectors represented by R_i for the i user pairs in which the similarity is the highest, and, the column vectors represented by C_j for the j user pairs in which the similarity is the highest, in the user-item evaluation matrix.

The convolution kernel $w_{l,m}$ is the calculation of a kind of mathematical convolution on the user ratings of the items, and the obtained features can be expressed as shown in equation (2):

$$f = \sigma\left(b + \sum_{l=1}^{L}\sum_{m=1}^{M} w_{l,m} a_{j+l,k+m}\right) \tag{2}$$

where f denotes the new feature representation of the item rating matrix obtained by the user after convolution with the convolution layer, and the values of the item ratings we use L to represent the row vector of the matrix and M to represent the column vector of the matrix, respectively. The number of rows and columns of the matrix, $W_{l,m}$ denotes the computed value in the m column of the l row of the matrix and l is less than or equal to L and greater than or equal to 1, and m is greater than or equal to 1 and less than or equal to M. $a_{j+l,k+m}$ denotes that $k+m$ such a large number of items are rated by the $j+l$ user. A user to rate, we all know that the activation function in the neural network is very heavy a function, usually the most commonly used are ReLu function, tanh function and so on. We used the familiar sigmoid function, as shown in equation (3):

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

After a series of calculations, at this point, a matrix (containing column vectors and row vectors) is obtained, in order to calculate the best great similarity of two neighboring objects can be very stable, each time in the feature extraction, a different convolution kernel should be introduced to the calculation so as to obtain a number of feature matrices to ensure that the results of the calculation is stable.

According to formula (4), we first unify the rows of the regulation layer and the feature matrix so that they are the same, and then connect each row of neurons of these two to each other, and use the sigmoid function to get the results of the regulation layer. The neuron values of the conditioning layer are shown in equation (4):

$$T_i = \sigma\left(\sum_{f=1}^{F} t_{i,f}\right) \tag{4}$$

where T_i denotes the value of the i th neuron in the conditioning layer, $t_{i,f}$ denotes the value of the i th row of the feature matrix in the i th column of the feature matrix in the i th column of the feature matrix in the i th column of the feature matrix, and $l \le i \le l, 1 \le f \le F$, I, F denote the number of rows and the number of columns of the feature matrix respectively. The formula for exchanging rows and columns of the input scoring matrix at the conditioning level is shown in equation (5):

$$\text{exchange}(k_i, k_j) = C_i^k \Theta C_j^k \tag{5}$$

where $\text{exchange}(k_i, k_j)$ denotes the matrix transformation function and k_i represents the different rows and columns. Its main function is the item scoring matrix transformation operation. If there are the same row or column vectors in C_i^k and C_i^k, it is only necessary to exchange the row or column vectors that are different from each other. The matrix features between users and goods are measured using information entropy, the smaller and stable change of the entropy value indicates the end of the characterization work. The information entropy calculation method is shown in (6):

$$S(p_1, p_2, ..., p_n) = -K\sum_{i=1}^{n} p_i \log_2 p_i \tag{6}$$

where K is the constant, usually taken as 1. However, p_i denotes the probability of the occurrence of this sample, the specific formula is shown in equation (7):

$$p_i = \frac{c}{\sum_{l=1}^{n^{'}-k} l} \tag{7}$$

c denotes the number of times the rows or columns are exchanged this time for the first i sample, $\sum_{l=1}^{n^{'}-k} l$ denotes the number of times all the rows or columns in $R_{n\times m}$ may be exchanged, and $n^{'}$ denotes the number of rows or columns in $R_{n\times m}$. The number of rows n or columns m in the program.

II. A. 2) Predictive scoring metrics

The results of the recommendation need a means of evaluation, this experiment chose the MAE. It is a more commonly used quality evaluation of the performance of the recommendation results, the MAE calculation method is simply to select the user's prediction and the actual ratings of the difference between the cumulative summation and then averaged. The MAE value and the accuracy of the recommendation is inversely proportional to the recommendation, in order to make the recommendation effect is better, we should choose the smaller MAE value. For example, $\{P_1, P_2 ..., p_N\}$ denotes the set of ratings of N users predicted by the recommendation, $\{q_1, q_2 ..., q_N\}$ denotes the corresponding set of actual ratings of N users, then the mean absolute error MAE is calculated as shown in equation (8):

$$MAE = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N} \tag{8}$$

II. A. 3) Local Characterization Experiments

In order to validate the local characterization experiment, we randomly select the similarity between two users or items to be calculated, in order to validate the similarity criterion between them, we use the Euclidean distance and the Pearson correlation coefficient as the tools, and we use the A dataset and the B dataset respectively to do a comparison experiment. Take a look at the results after they are locally characterized. The calculation formula is as follows:

$$d = \frac{\sum (R_r \cap T_r) - 1}{2^{*} | R |} \tag{9}$$

where the target set of r elements is a set with two other elements, collectively called R_r. And T_r is a three-element set containing the r element in the test set and two other elements that are close together. $R_r \cap T_r$ This denotes the intersection of the elements in the test set of these two target sets. The larger the value of d, the higher the similarity of their local features. The $| R |$ denotes the size of the target set involved in the computation.

Figure 1,, and Figure 2 show the results of the experiments on dataset A and B respectively, from the experiments it can be seen that the sparsity is in the interval of 0.2 and 0.5, and the degree of local characterization Pearson correlation coefficient in dataset A is a little higher than the Euclidean distance and CNN local similarity prediction. The sparsity is similar for all three in the interval of 0.7 and 0.9. In the A dataset experiments, the CNN local similarity prediction is higher than the other two methods when the sparsity is in the 0.7 and 0.9 interval.

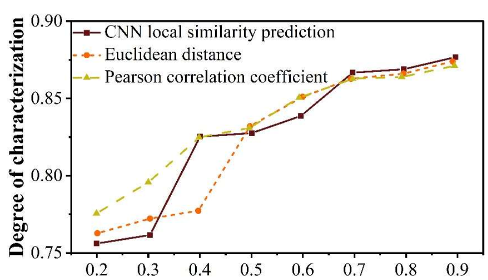

Figure 1: Experimental results of the A dataset

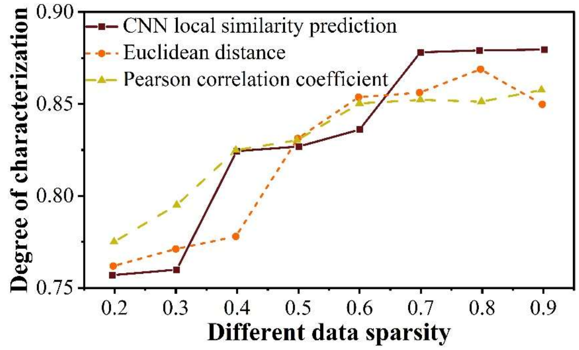

Figure 2: Experimental results of the B dataset

II. B.Personalized Recommendation Model

Considering that the data in this study is time-series data, for this reason, this section proposes a BiLSTM-based personalized recommendation model for libraries based on standard deep learning recommendation models.

II. B. 1) Basic Deep Learning Recommendation Models

As shown in Figure 3, the basic deep learning recommendation model is built based on the Embedding layer and the MLP layer of the multilayer perceptron, which is mainly composed of the input layer, embedding layer, splicing layer, and MLP layer [24], [25]. Embedding, also known as "vector mapping", refers to the use of low-dimensional dense vectors to represent an "object", and the spatial distance between the vectors can reflect the correlation between the "objects". At present, Embedding technology has become the basic core operation in deep learning and is widely used in deep learning frameworks. Book features and reader features are converted into dense vectors that can be handled by the neural network through the Embedding layer, which are spliced and input into the MLP for training. In the basic deep learning recommendation model, readers' borrowing order is not taken into account, but all the information is input into the neural network for training. In practical application scenarios, the reader's borrowing history is a time series data, and the time series data have a certain dependency relationship, which can reflect the evolution of the user's potential interest, which is important for the “next recommendation”. Therefore, on the basis of the standard neural network recommendation model, this thesis adopts Bi-LSTM model for feature extraction of readers' borrowing preferences, taking into account the sequence of readers' borrowing history. At the same time, in order to alleviate the cold start problem of readers and increase the number of positive samples, textbook information is introduced into the borrowing history.

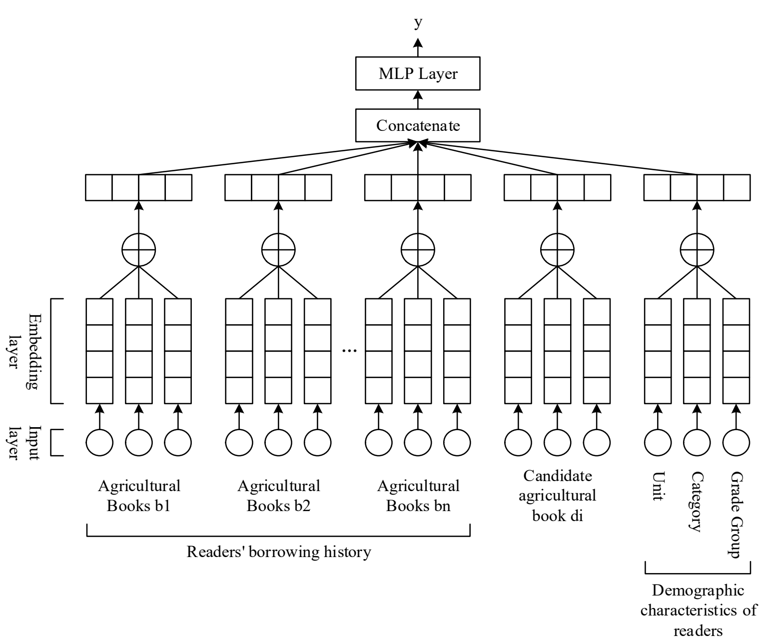

Figure 3: Basic deep learning recommendation model

II. B. 2) Bi-LSTM modeling

Bi-LSTM, also known as Bidirectional Long Short-Term Memory Network, is a network model that combines forward LSTM and reverse LSTM. Since unidirectional LSTM has the problem of long-term dependency, which leads to information loss, nowadays the industry commonly adopts Bi-LSTM model to replace LSTM.Bi-LSTM model combines forward and reverse bidirectional LSTM network coding, which can avoid the information loss caused by unidirectional LSTM to a certain extent.The training process of Bi-LSTM model is shown in Fig. 4. Forward LSTM encoding will get forward hidden state value $\vec{h}_t$, reverse LSTM encoding will get reverse hidden state value $\overleftarrow{h}_t$, averaging the two hidden state values at the same moment will get hidden state h_t at moment t, which integrates forward and reverse features and overcomes the disadvantage of unidirectional LSTM that can only be encoded from one direction.

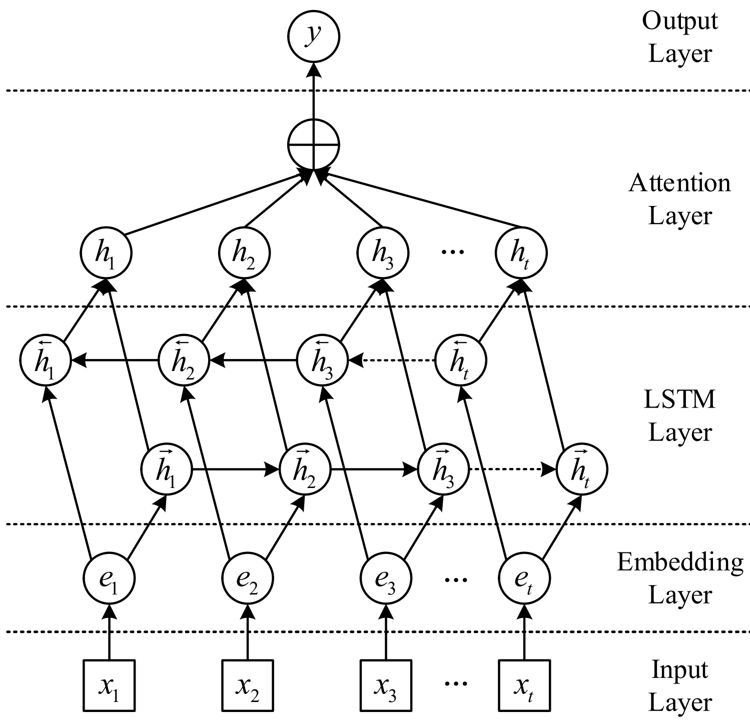

Figure 4: Bi-LSTM Training Process

II. B. 3) Reader preference feature extraction

We utilize Bi-LSTM model to extract reader preference features. If a reader u borrows n books, the borrowing record is denoted as $B^u = \{b_1^u, b_2^u, ..., b_j^u, ..., b_n^u\}$, where b_j^u denotes the j th book borrowed by reader u, then $\{e(b_1^u), e(b_2^u), ..., e(b_j^u), ..., e(b_n^u)\}$ denote the feature vector representation of agricultural books obtained after feature extraction, and then input into the Bi-LSTM model. each moment of the LSTM will have a vector output, and the output of the cell state of each moment of the forward LSTM is denoted as $\overline{H} = \{\overline{h}_1, \overline{h}_2, ..., \overline{h}_n\}$, and the output of the cell state at each moment of the inverse LSTM is given by $\overline{H} = \{\overline{h}_1, \overline{h}_2, ..., \overline{h}_n\}$ to denote it, and the final output h_t at moment t is the average of the forward and inverse LSTM. The final interest vector of the reader u can be expressed either in terms of the cell state at the last moment or in terms of the average of the cell state outputs at each moment. When the cell output state at the last moment is used, the final interest vector representation of user u is shown in Equation (10):

$$e(u) = h_n^u \tag{10}$$

When the average value is used, the final interest vector representation of user u is shown in Equation (11):

$$e(u) = \frac{1}{n} * \sum_{j=1}^{n} h(b_j^u) \tag{11}$$

II. B. 4) Model construction

The personalized recommendation model based on BiLSTM is shown in Fig. 5, using the bidirectional long and short-term memory network Bi-LSTM model to extract the reader's interest preferences, and the obtained reader vectors are spliced with the candidate book vectors and the reader's demographic feature vectors to obtain the final prediction results by inputting them into the MLP layer.

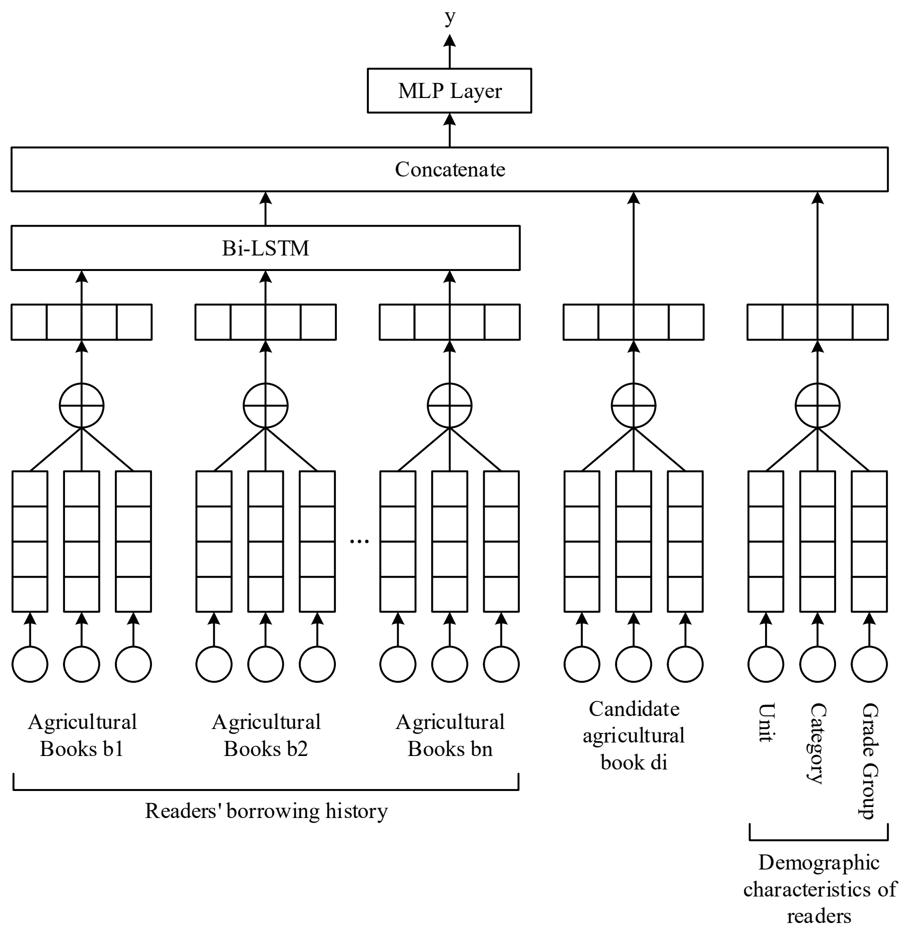

Figure 5: Personalized recommendation model based on BiLSTM

II. C.System design

On the basis of the previous recommendation model, the library personalized information recommendation system in this paper is designed to meet the personalized needs of readers, and the application system is designed according to the actual situation of the current state of library management, which has a relatively high practical application significance. The system adopts MVC three-layer architecture design, the mature enterprise application platform J2EE architecture brings good security and reliability to the system, and the excellent characteristics of one-time compilation and cross-platform operation make it have excellent portability. The specific design process is shown below:

II. C. 1) System objectives and principles

(1) System design objectives

The development, I hope to be able to design and implement an information security, reliable, simple, clear, easy to use, easy to install, fast, easy to learn, easy to use and easy to expand the library book personalized recommendation system, can be based on the reader's reading list recommended books, and even draws on the current particularly popular SNS (Social Network System), analyze the readers of the circle of the common interests of the focus to recommend books. The main functions that the system can accomplish are: system login that can distinguish between administrators and ordinary users, book database management, user circle system, behavioral record analysis, and book recommendation.

(2) System design principles

The development of this system, the first thing to comply with is the unified process of software development, and use the Unified Modeling Language (UML) to strictly control the implementation. Secondly, a series of principles to be followed for the design of this system are:

(a) Principle of system security.
(b) The principle of system data reliability.
(c) The principle of easy and quick installation of the system.
(d) The system's easy to learn, easy to use, the principle of simplicity of use.
(e) The principle of scalability of the system

II. C. 2) Overall system architecture

Because this system is B/S architecture, the overall architecture of the system can be represented by the B/S structure diagram. The B/S structure of the application system, the client does not need to install special client software, greatly simplifies the client computer load, reduces the cost and workload of the system maintenance and upgrading, and reduces the overall cost of the user [26], [27]. Its advantage is a one-time in place development, can realize different people, from different locations, with different access methods (such as LAN, WAN, Internet/Intranet, etc.) to access and operate the common database. It can effectively protect the data platform and manage access rights, the server database is also very safe, especially after the emergence of cross-platform languages such as JAVA, B/S architecture management software is more convenient, fast and effective.

II. C. 3) Data architecture

The system database is the foundation of the whole system, in which the data architecture can be expressed by the diagram, and there exists a certain connection between various data, and the specific system data architecture diagram is shown in Figure 6. Among them, the data in the readers' lending record table inherits the basic book information table, the system administrator and the readers' circle administrator table inherits from the user information table, the data in the book category table is associated with the basic book information table, and the data in the readers' circle table is associated with the user information table.

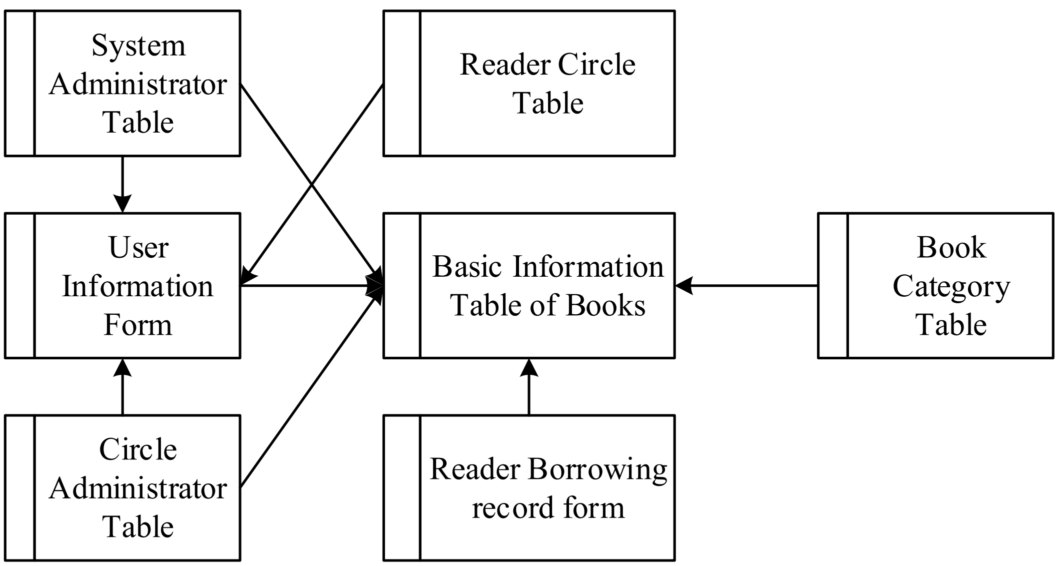

Figure 6: And the data architecture of the system!

II. C. 4) Deployment architecture

The deployment architecture describes the physical configuration used to deploy and run the software. This section indicates the main deployment of the system and the deployment architecture. The focus is on the software deployment of the system, so the system operating system and third-party software such as firewalls are ignored. The entire application is deployed on a single application server or a cluster of multiple application servers. The database can be deployed on a single database server or a cluster of multiple database servers as needed. As needed, a separate web server or a cluster of web servers can be deployed to improve the responsiveness and reliability of the system. It should be noted that since the system is a personalized recommendation system, it is necessary to push the information to the readers in time, so the client that the readers browse is not necessarily PC, but may also be a cell phone WAP browsing, or the readers view the SMS sent by the system.

II. C. 5) System Functional Architecture Design

The design of a system takes into account the functions realized by the whole system, so before going into coding, there should be a functional architecture diagram of the system. Based on this architecture diagram, the various modules of the system and the functions that each module can fulfill can be clearly seen. The design of the system will be based on the various modules in the diagram, respectively, to carry out the realization of various functions. The whole system consists of reader borrowing behavior record module, book personalized recommendation module, SMS platform, database management module, reader circle module, user management module and so on.

II. C. 6) System modeling

System modeling is done using the Unified Modeling Language (UML), which is a set of diagrams supported by a single model, commonly used in software design and development. These diagrams help in the representation and design of software systems, especially those constructed using an object-oriented approach. Moreover, one of the greatest advantages of the Unified Modeling Language as a tool for system analysis and design is the inheritance between artifacts during the design process, i.e., the original analysis and design results can be progressively refined in different iteration cycles, which is the model used for the detailed design of this system.

II. C. 7) Database design

The design of the database is crucial for the whole system. The good or bad database design is directly related to the quality of the whole system, which will be designed and described in more detail for the fields, data tables and the relationship between the tables in the database of the system. As the whole system is logically divided into many parts, although the system database does not have many information tables, the amount of information is relatively large, and there are many cross-applications between the various tables, so we set up a unique identification number for the information tables, for example: the user table will be USER_ID (the user's ID number) as the primary key, and each table is connected to the other information tables through the ID number of each table, and there is also an interdependence between them. There are also mutual dependencies, associations, etc. between them, which are represented here in a data relationship class diagram.

III. System depth probe analysis

III. A. Personalized Recommendation Model Validation Analysis

III. A. 1) Hyper parameterization

In the algorithmic model, the choice of parameters often has a great impact on the specific performance situation of the model, and the choice of reasonable parameters can affect the performance of the model to a greater extent, but also improve the stability of the model and reduce the model's impact on the noise in the data. This experiment mainly considers 4 aspects of the parameters:

(1) Division of time series: Since the time series is added in the processing of the data set, in order to verify the impact of the division of the time series on the prediction results, the time series has a division into the user ratings from top to bottom for the time positive order and time inverted order.

(2) Learning rate: the choice of learning rate affects the convergence speed of the model, i.e. the number of iterations required to reach the optimal solution. A larger learning rate can speed up the convergence of the model, but too large a learning rate may lead to unstable updates and oscillations. On the contrary, a smaller learning rate may require more iterations to converge.

(3) Number of BiLSTM layers: the number of BiLSTM stacked layers can improve the model's expressive and memorization abilities. Deeper models usually perform better than shallower models, especially for complex datasets. However, increasing the number of layers leads to an increase in the computational overhead of the model, which may lead to overfitting for some datasets.

(4) Number of iteration samples Batch_Size: Increasing the number of iteration samples allows the model to fit the training data better, which improves the model's prediction accuracy, but may lead to model overfitting. And decreasing the number of iteration samples may lead to the model not being able to fully learn the distribution and features of the data, thus affecting the generalization ability of the model.

III. A. 2) Data analysis

(1) Selection of hyperparameters and experimental results

To ensure the selection of appropriate hyperparameters, comparative experiments will be conducted on the four parameters mentioned in the previous subsection.

(a) Time ordering

By sorting the scoring times in the dataset from top to bottom into positive and negative order, the MSE values obtained from 10 times of comparison experiments with different time ordering are shown in Fig. 7. The results obtained from 10 experiments in the time-ordered data for the same user from top to bottom for the most recent time to the most recent time i.e. reverse order, the MAE value of the algorithm is smaller.

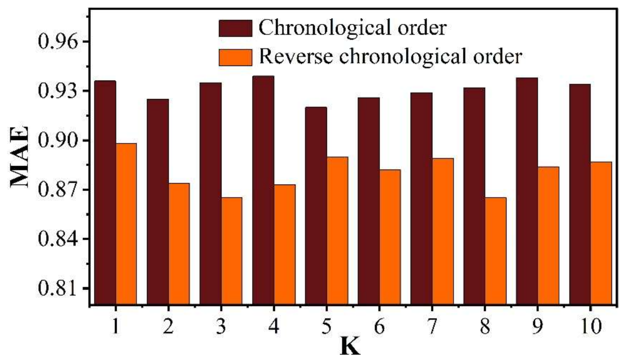

Figure 7: The MAE values under different time sorts

(b) Learning rate

The candidate range of learning rate is [0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1], and experiments are carried out at different learning rates, and the changes in the MAE value of the model are shown in Figure 8. After the experiments, it can be seen that the algorithm reaches the minimum MAE value of 0.787 at a learning rate of 0.001.

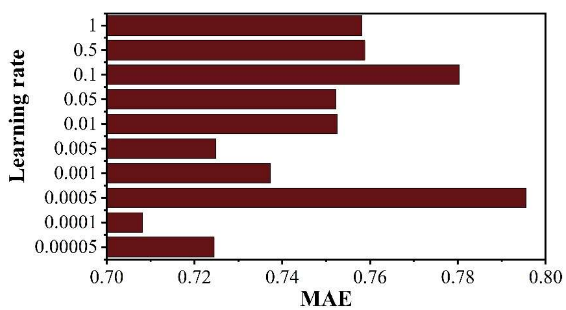

Figure 8: The variation of the MAE value of the model

(c) Number of LSTM layers

The candidate ranges for the number of BiLSTM layers are [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], and the values of MAE under different numbers of BiLSTM layers are shown in Fig. 9. The performance of the timing data in the figure shows that with the gradual increase in the number of BiLSTM layers, the value of MAE appears to decrease and then increase, which occurs to indicate that the more the number of BiLSTM layers is, the stronger the learning ability of the timing data, and it is more likely to overfitting, but if the number of layers decreases, the learning ability of the timing data will be weakened, and therefore, the number of layers is set to 2 layers.

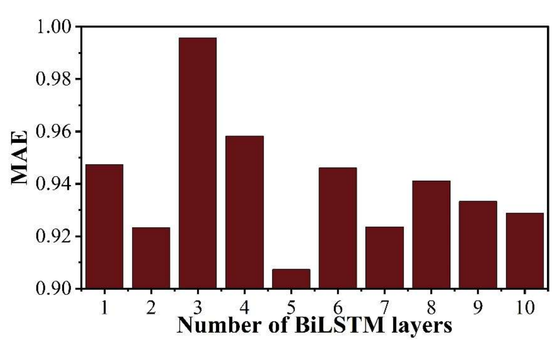

Figure 9: MAE values under different numbers of BiLSTM layers

(d) Number of iteration samples Batch_Size

The values of the number of iteration samples Batch_Size are in the range of [2, 4, 8, 16, 32, 64, 128, 256, 256, 512, 1024], and the MAE values under different Batch_Size are shown in Fig. 10. According to the experimental results, it can be seen that the algorithm has the best performance and the smallest MAE value when the value of Batch_Size for the number of iteration samples is 256.

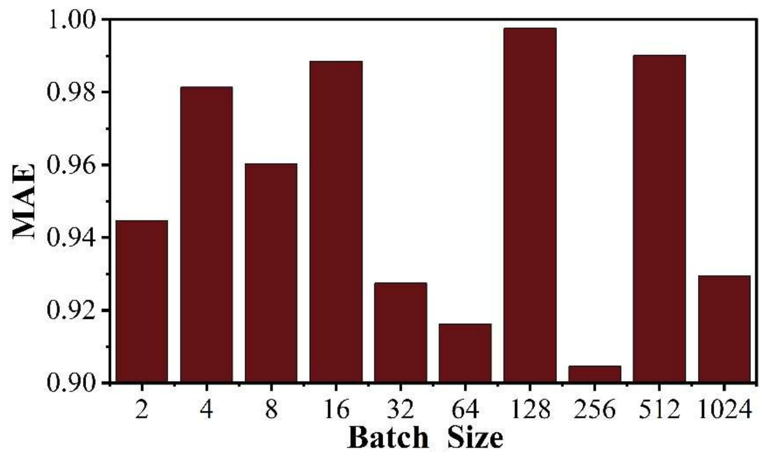

Figure 10: MAE values under different Batch sizes

(2) Comparison experiment results and analysis

In order to verify the effectiveness of the algorithmic model designed in this paper, this paper will choose to compare with UserCF recommendation algorithm, ItemCF recommendation algorithm and ConvMF (convolutional matrix factorization) algorithm. The UserCF recommendation algorithm and ItemCF recommendation algorithm have been described in detail in the previous sections, while the ConvMF recommendation algorithm mainly utilizes the idea of convolutional neural networks to represent the user-item rating matrix as a three-dimensional tensor, where each element of the matrix represents the user's rating of the item. The algorithm introduces a convolutional kernel to capture the local features of the user-item matrix. Based on the convolutional feature map, the user-item matrix is decomposed into two low-dimensional matrices by matrix decomposition, which represent the potential features of the user and the item, respectively. Update the parameters in the matrix decomposition. Through iterative optimization, the model is made to approximate the actual scores of the user-item matrix. Under the same conditions, the three algorithms mentioned above as well as the algorithm proposed in this paper are carried out 10 times respectively, and the results of the comparison experiments are shown in Fig. 11. Based on the time-series data analysis in the figure, it can be seen that the performance of the algorithm proposed in this paper is better than other algorithms on the preprocessed dataset using the MSE value as the comparison term, and the experimental results also prove that the algorithm proposed in this paper has better accuracy in personalized information recommendation in libraries.

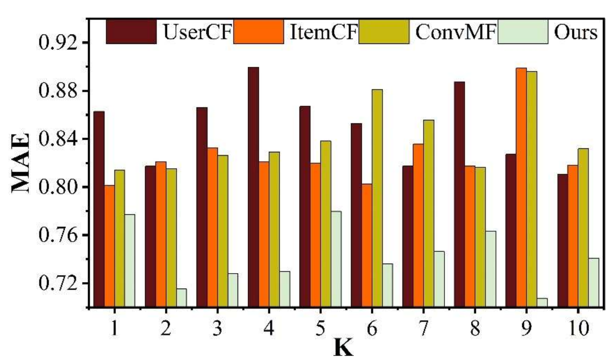

Figure 11: Compare the experimental results

(3) Analysis of ablation experiments

In order to study the influence of each component within the BiLSTM algorithm on the algorithm, the two-way propagation mechanism is discarded on the basis of the original algorithm, so that the important role of the two-way propagation mechanism in the algorithm model can be explored. The specific experimental results are shown in Figure 12. Combined with the results of the timing data analysis in the figure, it can be seen that the performance of the original algorithm model BiLSTM is superior to that of the algorithm model that discards the two-way propagation mechanism, indicating that the two-way propagation mechanism can extract user social information more effectively, making the characteristics of the user nodes richer, and can improve the model's performance of library personalized information recommendation.

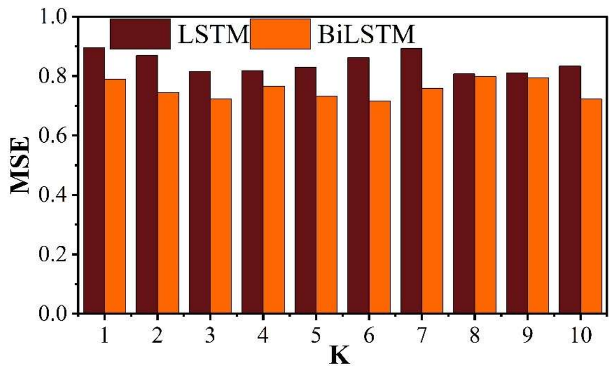

Figure 12: Ablation experiment results

III. B. System Performance Testing

The objectives, requirements and overall design scheme of the personalized information recommendation system for intelligent libraries have been described in the previous section. At the same time, the realization ideas of the main functional modules of the system have been described in detail. In this chapter, the aspects of test scheme development, environment construction and test result analysis will be elaborated. The performance test of the system is carried out by formulating a reasonable test plan. The details are as follows:

III. B. 1) Selection of test methods

Software testing is an important part of the software development process. Software testing can test the usability of the software as well as find the problems of the software through testing. Commonly used software testing methods mainly include the following two categories:

(1) White-box testing: mainly for the system's program composition or logical implementation of the test, which contains the development process of each logical mastery.

(2) Black-box testing: the system is regarded as a black box, ignoring the program composition and logic inside the system. In the process of applying black box testing, the main focus is on the requirements of the software. The purpose of testing is to find out the non-conformity with the software requirements in the process of testing. In summary, this paper uses a combination of black-box testing and white-box testing to test the overall performance of the intelligent library personalized information recommendation system.

III. B. 2) Test environment setup

At present, only the library personalized information recommendation system is still in the laboratory testing stage, according to the laboratory environment to build the software testing environment, based on the B/S architecture of the library personalized information recommendation system consists of the foreground website and the background management of two parts. Among them, the user is the user of the foreground website, which is mainly responsible for the display of commodity information and the collection of user behavior. The administrator is the user of the background management, which is responsible for the management of the data in the system and the realization of the recommendation function. The server side of the system is deployed on Apache server, and the database adopts MySQL. The specific test environment network topology diagram of the system.

III. B. 3) Performance testing

This subsection mainly focuses on AB test to test the performance of the system. This system uses Apache's AB test to perform performance testing of the frontend website.The performance metrics of AB test mainly include the number of users, throughput rate, average server request time and average user waiting time. Run the ab -c10 -n1000 http://localhost/Supermart/index.php command to access the homepage of the foreground website. Where -

c10 means that the number of concurrent users is 10 and -n2000 means that the total number of requests executed is 2000. When the number of concurrent users is 30, the throughput rate of the server is 869.328reqs/s, the average request waiting time of the users is 35.326ms, and the average request waiting time of the server is 1.381ms. For better performance testing, next increase the number of concurrent users from 1 to 200 to test the performance metrics under each condition, and the results of different concurrent user numbers are shown in Table 1. As can be seen from the data performance in the table, the server throughput rate changes continuously as the number of concurrent users increases from 1 to 200, and the peak throughput rate occurs when the number of concurrent users is 50. Both the average server request waiting time and the average user request waiting time show an increasing trend with the number of concurrent users. When the number of concurrency reaches 200, the average user request waiting time is 271.829ms (about 0.27s), the average user request waiting time of 0.27s does not affect the user experience too much, then the server's ability in this test basically meets the initial requirements of the system.

Table 1: Test results of different concurrent user numbers

Concurrent number of users	Throughput rate (reqs/s)	Average request waiting time of the server (ms)	User average request waiting time (ms)
1	270.367	1.917	3.094
10	862.603	1.158	12.439
30	869.328	1.381	35.326
50	905.784	2.124	56.22
70	808.928	0.05	85.886
90	803.894	0.655	111.151
100	801.08	2.128	126.936
120	807.095	2.295	149.974
140	841.14	0.909	166.892
160	799.412	2.23	199.272
180	737.406	1.765	245.346
200	735.114	1.722	271.829

IV. Conclusion

The intelligent library personalized recommendation system based on time-series data analysis proposed in this paper has achieved remarkable results. Through the local characterization of convolutional neural network, the system outperforms the characterization of Pearson's correlation coefficient method in the interval of data sparsity 0.2-0.5. The performance test results show that when the number of concurrent users is 50, the system throughput rate reaches a peak of 905.784reqs/s, which demonstrates a good concurrent processing capability; when the number of concurrency increases to 200, the average user request waiting time is only 271.829ms, which meets the actual application requirements. The ablation experiment proves that the two-way propagation mechanism of BiLSTM plays an important role in improving the recommendation performance, which verifies the key position of temporal modeling in book recommendation. This system provides an effective technical solution for library personalized service and promotes the in-depth development of intelligent library construction.

References

[1] Asemi, A., Ko, A., & Nowkarizi, M. (2021). Intelligent libraries: a review on expert systems, artificial intelligence, and robot. Library Hi Tech, 39(2), 412-434.

[2] Panda, S., & Chakravarty, R. (2022). Adapting intelligent information services in libraries: A case of smart AI chatbots. Library Hi Tech News, 39(1), 12-15.

[3] Kato, A., Kisangiri, M., & Kaijage, S. (2021). A review development of digital library resources at university level. Education Research International, 2021(1), 8883483.

[4] Anyim, W. O. (2018). E-Library resources and services: Improvement and innovation of access and retrieval for effective research activities in university e-libraries in Kogi State Nigeria. Library Philosophy and Practice, 0_1-21.

[5] Atanda, L. A., & Adeyemi, S. A. (2018). Information retrieval tools, catholicon to accessing library resources. Research Journal of Library and Information Science, 2(2), 30-35.

[6] Roy, B. K., Mukhopadhyay, P., & Biswas, A. (2022). Discovery layer in library retrieval: VuFind as an open source service for academic libraries in developing countries. Journal of Information Science Theory and Practice, 10(4), 3-22.

[7] Xiang, C. Z., Fu, N. X., & Gadekallu, T. R. (2022). Design of resource matching model of intelligent education system based on machine learning. EAI Endorsed Transactions on Scalable Information Systems, 9(6).

[8] Bi, S., Wang, C., Zhang, J., Huang, W., Wu, B., Gong, Y., & Ni, W. (2022). A survey on artificial intelligence aided internet-of-things technologies in emerging smart libraries. Sensors, 22(8), 2991.

[9] Zhang, X. (2022). On the innovative work and development of library reader service in the era of artificial intelligence. Wireless communications and mobile computing, 2022(1), 3779660.
[10] Shahzad, K., Khan, S. A., Iqbal, A., & Javeed, A. M. D. (2024). Identifying university librarians’ readiness to adopt artificial intelligence (AI) for innovative learning experiences and smart library services: an empirical investigation. Global Knowledge, Memory and Communication.
[11] Li, X. (2024, August). Research on the Application of Artificial Intelligence in Library Reader Behavior Analysis and Personalized Service. In 2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE) (pp. 902-907). IEEE.
[12] Wang, C., & Sha, Z. (2021, November). Research on intelligent information system of library under big data and digitization technology. In Journal of Physics: Conference Series (Vol. 2083, No. 4, p. 042063). IOP Publishing.
[13] Hamad, F., Al-Fadel, M., & Fakhouri, H. (2023). The provision of smart service at academic libraries and associated challenges. Journal of Librarianship and Information Science, 55(4), 960-971.
[14] Zhuang, Y. (2021). Optimization of the personalized service system of university library based on internet of things technology. Wireless Communications and Mobile Computing, 2021(1), 5589505.
[15] Lin, N. (2024). Personalized book intelligent recommendation system design for university libraries based on IBCF algorithm. IEEE Access.
[16] Zhang, M. (2025). Design and implementation of intelligent library personalized information recommendation model based on reinforcement learning. J. COMBIN. MATH. COMBIN. COMPUT, 124(351), 375.
[17] Pang, N. (2022). A personalized recommendation algorithm for semantic classification of new book recommendation services for university libraries. Mathematical Problems in Engineering, 2022(1), 8740207.
[18] Marzuki, M., Azero, S. F. Z., Zamzuri, N. A. A. M., Razilan, M., Kadir, A., & Campus, S. A. (2025). A Systematic Literature Review of User Behavior and Personalization in Digital Libraries. International Journal of Research and Innovation in Social Science, 9(1), 4830-4842.
[19] Kowsick, N. P., & Ramasamy, K. (2024). Implementing AI-based recommendation systems for personalized financial services in libraries. In Improving library systems with AI: Applications, approaches, and bibliometric insights (pp. 235-243). IGI Global.
[20] Zhao, L. (2021). Personalized recommendation by using fused user preference to construct smart library. Internet Technology Letters, 4(3), e273..
[21] Sun, H. (2020, June). Research on interest reading recommendation method of intelligent library based on big data technology. In Web Intelligence (Vol. 18, No. 2, pp. 121-131). Sage UK: London, England: SAGE Publications
[22] Sa’ari, H., Sahak, M. D., & Skrzeszewskis, S. (2023). Deep learning algorithms for personalized services and enhanced user experience in libraries: a systematic review. Mathematical Sciences and Informatics Journal (MIJ), 4(2), 30-47.
[23] Vidya K. Sudarshan,Reshma A. Ramachandra,Smit Ojha & Ru San Tan. (2024). DCEnt-PredictiveNet: A novel explainable hybrid model for time series forecasting. Neurocomputing,608,128389-128389.
[24] Sinha Kaushlendra Kumar & Suvvari Somaraju. (2023). Repetition Dynamics-based Deep Learning Model for Next Basket Recommendation. SN Computer Science,5(1),
[25] Tegene Abebe,Liu Qiao,Gan Yanglei,Dai Tingting,Leka Habte & Ayenew Melak. (2023). Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems. Applied Sciences,13(2),726-726.
[26] Qing Liu. (2024). Design of University Student Status Management System Based on Web B/S Architecture. Advances in Computer, Signals and Systems,8(7),
[27] Jian Wei Leng & Xiang Lian Fu. (2014). Design and Application of Energy Monitoring System Based on Serial Port Server and B/S Framework Configuration. Applied Mechanics and Materials,2948(496-500),1613-1616.