International Journal of Web-Based Learning and Teaching Technologies
Volume 19 • Issue 1

Application of Recommendation Algorithms Based on Social Relationships and Behavioral Characteristics in Music Online Teaching

Chunjing Yin, School of Humanities and Arts, Jiangsu Shipping College, China*

ABSTRACT

This research designed an improved collaborative filtering algorithm to be responsible for music recommendation tasks in the online music teaching platform. This algorithm integrates the user's social trust into the similarity calculation formula. Then, the algorithm uses behavioral feature data driven by preferences, music tags, and popularity as the basis for recommendation calculation. It adopts user data testing on an online music teaching platform. The results showed that when the number of recommended music was eight, the recommended recall rates of XCF, CTR, TSR, and UB-CF recommendation models reached their maximum, reaching 97.82%, 95.26%, 93.95%, and 88.72%, respectively. The AUC and average computational time of the ROC curves for XCF, CTR, TSR, and UB-CF recommended models are 0.7, 0.68, 0.64, 0.57, and 160ms, 136ms, 114ms, and 88ms, respectively. The experimental data shows that the recommendation accuracy of the music recommendation model designed in this study is significantly higher than that of traditional recommendation models.

KEYWORDS

Behavioral Characteristics, Collaborative Filtering Algorithm, Music Recommendations, Online Teaching

In today's world, science and technology represented by next-generation communication technology, Internet of Things, and artificial intelligence are causing drastic changes in the industry related to network information exchange. People can conveniently access resources of interest anytime and anywhere they need them. However, in this process, people also face information overload. That is, there is too much data and information available for one to choose and use, making it difficult for individuals to make decisions quickly or smoothly. Users can utilize different types of search tools to help them quickly find the products they need. However, these products themselves also have various drawbacks and shortcomings (Choi et al., 2022; Li et al., 2021; Mohammadi et al., 2022). For example, most search engines require users to clarify their search needs and conduct searches based on clear keywords. However, the search results obtained using this method are lacking in individuality (Peng, 2021). And in many cases, users are not very clear about their

DOI: 10.4018/IJWLTT.338319 *Corresponding Author

preferences and needs. This leads to the inability to search for keywords and conduct searches effectively (Zeeshan et al., 2021). There are similar problems in online music teaching platforms, where teachers and students cannot have direct contact, making it more difficult for teachers to meet students' learning needs. At the same time, each student's knowledge foundation and interest preferences are also different. Using the same music teaching materials to teach different types of students will inevitably not achieve good teaching results. Therefore, it is necessary to select more suitable continuous music and appreciation music as auxiliary teaching materials for students. It is necessary to design a system that can accurately identify user music preferences to help teachers carry out online teaching. The problem that needs to be studied is the low accuracy of the recommendation system in online music teaching. To investigate the research question, a similarity calculation method for the social relationship trust of mixed users was designed; this is the study's first point of innovation. The second innovation in the study is using behavioral feature data driven by preferences, music tags, and popularity as the basis for the recommendation calculation and designing an improved recommendation system. The goal of this study is to design a recommendation model with higher accuracy in online music teaching, in order to improve the quality of online music teaching in China.

RELATED WORKS

A recommendation system is a system that recommends items or information that best meet users' needs by analyzing their historical behavior and interest preferences. The workflow mainly includes data collection, data preprocessing, feature extraction, similarity calculation, and recommendation calculation. The working principle of a recommendation system is to collect users' historical behavior records, collect basic information and needs, extract user features, calculate similarity between users, find similar users, and recommend items or information that best meet their interests and needs to improve the accuracy and satisfaction of recommendations. A recommendation system that includes music recommendations has practical value in saving users' time in searching for more accurate products and information. Therefore, recommendation systems have also become a key research object for domestic and foreign experts and practitioners.

Some researchers have found that the problems or characteristics inherent in the data used to train recommendation systems can affect the recommendation results of the system and have designed improved recommendation systems that can solve corresponding problems. Rabiu et al. (2022) found that data sparsity and category imbalance in recommendation systems have a significant impact on recommendation results. Therefore, a recommendation system based on adaptive short-and long-term memory neural network algorithm and collaborative filtering algorithm is designed. The test indicates that the accuracy of the recommendation system on the TOP5 recommendation results of the dataset has been improved by 12.8% compared to the previous method (Rabiu et al., 2022). Gwadabe et al. (2022) proposed a processing method based on an improved generative graph neural network algorithm to address the issue of poor processing and recommendation performance of traditional recommendation systems for long sequence data. This method is used to preprocess data that requires recommendation calculations, and the test indicates that the recommendation results processed by this algorithm have higher recommendation accuracy than those without processing (Gwadabe et al., 2022).

Zhang et al. (2021) found that implicit recommendations have attracted the attention of many scholars. However, the uncertainty of implicit feedback data poses significant challenges to recommendation. Therefore, the authors propose a causal neural fuzzy inference algorithm to model missing data in implicit recommendations. The test showcases that this algorithm has better recommendation performance on three real datasets, and the recommendation calculation speed is faster (Zhang et al., 2021). Zy et al. (2020) believe that the unknown entries in the rating matrix actually contain a significant amount of useful information for prediction. This information is usually discarded in traditional methods. Therefore, Zy et al. designed an improved recommendation strategy based on the idea of semi supervised learning. The experiment showcases that this method significantly outperforms the reference method in recommendation accuracy and has a certain degree of robustness to the diversity of the dataset (Zy et al., 2020). Zhao et al. (2020) found that traditional recommendation techniques are hindered by the simplicity and sparsity of user project interaction data. Most studies focus on a single type of external relationship, without fully utilizing the potential relationship between users and items. Therefore, they propose a recommendation algorithm that integrates heterogeneous networks and applies it to multiple real recommendation task datasets (Zhao et al., 2020). The test indicates that the recommendation accuracy of this recommendation method is significantly higher than the algorithm before improvement, and it can consider more potential relationship information between users and items (Zhao et al., 2020).

Some scholars have found that recommendation system algorithms applied to certain specific scenarios have their own shortcomings and attempted to address these shortcomings. Vo et al. (2020) overcame the adverse impact of cold start on the performance of recommendation systems and designed an improved recommendation system using an implicit random gradient descent algorithm. The experiment illustrates that after applying this method, the recommendation calculation time and accuracy of the recommendation system are shortened by 15.2% and improved by 19.7% compared to the previous method (Vo et al., 2020). Sturluson et al. (2021) found that if the product customer evaluation matrix in the recommendation system is incomplete, there will also be more errors in the recommendation results. Therefore, the authors adopt a covalent organic framework to design a new recommendation system (2021). The test demonstrates that this recommendation system can greatly ensure the reliability of recommendation results when the user rating matrix is missing by less than 50%, but the calculation speed is significantly slower than traditional methods (2021). The authors believe that this drawback may be improved and improved in subsequent research (Sturluson et al., 2021).

In summary, although existing research has achieved certain results in various aspects of recommendation systems, there are still some shortcomings. These shortcomings include inadequate data processing in the recommendation system algorithm itself, resulting in low accuracy of recommendation results; in specific scenarios, recommendation algorithms have issues such as cold start, sparse data, and incomplete evaluation matrices, which directly affect the effectiveness and performance of recommendation systems. Additionally, some studies have focused too much on a single type of external relationship and have not fully utilized the potential relationship between users and items. These problems also exist in the field of music online teaching. Based on this, in order to overcome the shortcomings of existing research in the field of music online teaching, a recommendation algorithm that combines students' social relationships and behavioral characteristics is proposed. This study attempts to carry out work from the following aspects: developing a recommendation algorithm that can fully utilize user social relationships and behavioral characteristics, in order to improve the accuracy and satisfaction of recommending courses and learning resources in music online teaching; fully considering the interactive relationships and learning behaviors among students and combining individual differences and interest preferences to provide personalized music courses and learning resource recommendations for students; solving the cold start problem in music online teaching; utilizing user social relationships and behavioral characteristics to provide effective music course and learning resource recommendations for new students; and helping students adapt to the music online teaching environment more quickly. The final gap in this study is the inability to collect diverse research project materials to fully validate the universality of the designed model in music online teaching recommendations, which may limit the market-oriented application of this study. Future research will improve on this point.

DESIGN OF MUSIC RECOMMENDATION ALGORITHM THAT INTEGRATES SOCIAL RELATIONSHIPS AND USER BEHAVIOR CHARACTERISTICS

In this study, a music recommendation algorithm that integrates social trust indicators and user behavior characteristics was designed. This algorithm can be applied to online music teaching (Ravakhah et al., 2021). In this algorithm, random walk based on social relationship trust will be used to mine user trust relationships in the data to be recommended. The user behavior characteristics will be used to alleviate the cold start problem in recommendation algorithms (Jiang et al., 2021).

Design of Music Recommendation Algorithm Integrating Social Relationship Trust

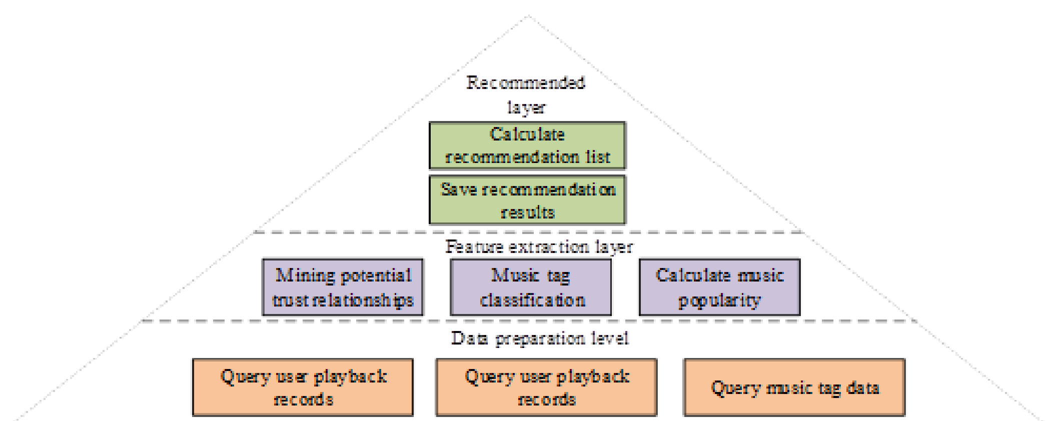
The structure of the entire recommendation system is shown in Figure 1, and it includes a recommendation module, a feature extraction model, and a data preparation model. The main content of the recommendation module is the improved recommendation model designed in this study. The main content of the feature extraction model is to use the topic model to cluster music tags and calculate music popularity. The main content of the data preparation model is to calculate or check the music information played and followed by users.

This study designs the core modules of the recommendation layer shown in Figure 1 as follows. The random walk algorithm designed by the opportunity graph model is of great significance to the recommendation system. There are two modes for conducting recommendation work in this mode (Da'U et al., 2021). In the first mode, the algorithm maps users, their historical behaviors, and items into a bipartite graph, and performs recommendation work on the graph (Wang, 2021). The second mode is carried out based on social networks, as users' close social relationships with peers can have a significant impact on their judgment. This is the reason why the second pattern design recommendation model was chosen for this study. Restarting random walk is an important improvement of random walk. The core principle can be described according to equation (1).

$$p^{(t+1)} = (1-a)p^{(t)}S + aq \tag{1}$$

In equation (1), $p^{(t)}$ represents the probability of random walk jumping from step t to node; S is the state transition matrix; q is the initialization vector of random walk; a represents the probability of returning to the starting point x with each step of the walk. Random walk can only be carried out after the final probability distribution is calculated according to equation (1). The following is a

Figure 1. Network teaching music recommendation system



method of mapping users and their accompanying trust relationships into bipartite graphs. Assuming there is a set of users U, music S, and user likes data C, G is a weighted directed graph, and V is the vertex set of the directed graph. There is a direct attention relationship between users i and j. Therefore, there is also a directed edge $\langle i, j \rangle$ between vertices i and j, with a corresponding weight of w_{ij}. Then the weighted adjacency matrix A of G can be calculated according to equation (2).

$$A = \left(a_{i,j}\right)_{i,j \in E} \tag{2}$$

In equation (2), E is the edge set of the graph. The element $a_{i,j}$ of adjacency matrix A is calculated according to equation (3).

$$a_{i,j} = \begin{cases} w_{i,j}, i, j \in E \\ 0, else \end{cases} \tag{3}$$

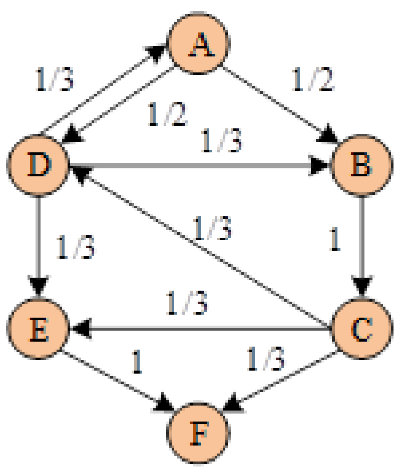
It should be noted that, considering that the concerns of users in G are unidirectional, the adjacency matrix A has asymmetry, as shown in Figure 2.

The transfer matrix S is obtained by normalizing G, and the calculation method is shown in equation (4).

$$S = D^{-0.5} A D^{-0.5} \tag{4}$$

In equation (4), D represents the degree of exit of each vertex in the adjacency matrix. After iterating k times according to the above method, the transition probability from the starting point to other points will stabilize. The next step is to sort all vertices according to the stable probability value. This can obtain the set of most trusted user vertices or the set of most similar user vertices corresponding to the starting point (Sharmila et al., 2021).

Figure 2. Weighted directed graph



This study fully considered the likes behavior characteristics of users in the computational dataset. This is also a point of concern in the random walk, so the corresponding adjacency matrix A of Figure G can be calculated according to equation (5).

$$\begin{cases} a_{i,j} = w_{i,j} + {c_{ij}} \big/ {\sum_{k \in U_i} c_{i,k}} \\ A = \left(a_{i,j}\right), i, j \in E \end{cases} \tag{5}$$

In equation (5), c_{ij} is the number of likes from user i to user j; $\sum_{k \in U_i} c_{i,k}$ is the sum of likes from user i to other users. If a user likes another user, it can be considered that the former trusts the latter more. Therefore, by modifying random walk through likes, the shortcomings of rough description of trust degree can be improved in the system (Da'U et al., 2021).

After optimizing the trust calculation method, there are still two issues in social networks. One is that some users do not like socializing and their data on social networks is relatively scarce; the second issue is that trusted users may not necessarily be similar users, and algorithms that combine trusted users with similar users will result in some incorrect recommendations (Zhu, 2021). In response to the above issues, this study combines trust and user scoring to construct similarity, which is called trust similarity (TS). The calculation method is shown in equation (6).

$$TS\left(u_i, u_j\right) = sim\left(u_i, u_j\right)\theta + (1-\theta)t\left(u_i, u_j\right) \tag{6}$$

Among them, u_i and u_j are the scoring data of users i and j; $sim\left(u_i, u_j\right)$ is the similarity in rating between two users. Expressed as Pearson coefficient, $t\left(u_i, u_j\right)$ represents the user's modified social network trust level; θ represents the weight distribution coefficient. The calculation method of Pearson coefficient $sim\left(u_i, u_j\right)$ is shown in equation (7).

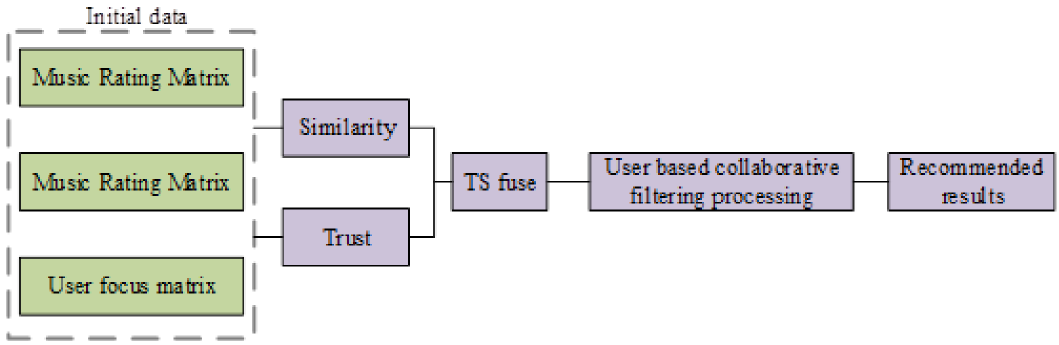
$$sim\left(u_i, u_j\right) = \frac{\left|N(u) \cap N(v)\right|}{\sqrt{\left|N(u)\right| \cup \left|N(v)\right|}} \tag{7}$$

$N(u)$ is the variance of scoring u. Finally, equation (8) is used to calculate the recommendation results of each user using the collaborative filtering algorithm. The calculation equation of the collaborative filtering algorithm based on the improved similarity is shown in equation (8).

$$p(u,i) = \sum_{v \in S(u,k) \cap N(i)} TS_{uv} r_{vi} \tag{8}$$

u, i represents the u-th user and the i-th item, respectively; $p(u,i)$ is the recommended value for the corresponding item - user; r_{vi} is the number of times the u-th user has touched the i-th item. To sum up, a music recommendation algorithm can be obtained, which integrates improved random walk trust, and its calculation is shown in Figure 3.

Figure 3. Improved recommendation algorithm



Design of Music Recommendation Algorithm Integrating User Behavior Characteristics

In a recommendation system, user behavior is influenced by surrounding users. The user's feedback consists of implicit and explicit feedback. Explicit feedback is easy to collect and calculate, but implicit feedback contains important information to understand user preferences and is difficult to collect and understand. This study proposes another music recommendation algorithm that integrates classified user behavior features. Analyzing the recommended dataset of this study, it was found that user behavior is triggered in three ways: preference driven, music label driven, and popularity driven. Therefore, it is necessary to extract these three types of feature data and perform correlation calculations to build an improved recommendation system. The preference driven activity feature (PAF) is a record of users' direct and effective actions. In the music recommendation theme of this study, the representative indicators are the user's music playback volume, collection volume and number of music likes. According to user usage habits, it can be inferred that the more times a user plays a specified song, the higher their liking for the song. Therefore, $PAF(u,s)$ can be calculated according to equation (9).

$$PAF(u,s) = \frac{count(u,s)}{\sum_{s' \in S(u)} count(u,s')} \tag{9}$$

$count(u,s)$ represents the number of times user u played on song s; $S(u)$ represents the total collection of historical music played by user u. In terms of music tag driven activity feature (TAF), users in the community can be regarded as sensors, and music tags can be considered as the recognition results of sensors. Therefore, it can be considered that the more certain labels a song receives, the more suitable the label is to describe the song. So here the number of tags were used to describe the weight of each tag in the song. The dataset used in this study already carries the tag weight $Poss(t,s)$ of each song. Meanwhile, it can be seen that TAF also implies the degree of symbolization between song labels and user preferences. Therefore, equation (10) can be used to calculate TAF.

$$TAF(u,s) = \sum_{t \in T(s)} Poss(t,s) \cdot \mathrm{Pr}\,ef(u,t) \tag{10}$$

In equation (10), $\Pr ef(u,t)$ represents the user's preference for labels. In this study, the tag t contained in songs played by users was used to represent their level of identification with the tag, as shown in equation (11).

$$\Pr ef(u,t) = \frac{count(u,t)}{total(u)} \tag{11}$$

$count(u,t)$ is the count of user u on label t ; $total(u)$ is the count of all tags contained by user u . Finally, it analyzes the Popularity driven Activity Feature (PopAF) aspect. Cold start is a common obstacle in recommendation systems. Therefore, using only TAF and PAF for recommendations is more likely to result in difficulties in cold starting. This study used popularity to fill in user data for cold start difficulties, believing that if a song is played more frequently by users over a period of time, it is considered a popular song. So equation (12) can be used to calculate $PopAF$.

$$PopAF(u,s) = \frac{\sum_{u' \in P(s)} play(u')}{\sum_{u \in S} play(u)} \tag{12}$$

$play(u')$, $play(u)$ represents the number of times the user played s and all songs. After the above three indicators are calculated, the user song bipartite graph can be constructed. This is a complete bipartite graph with weighted values, including user set and song set vertices. Considering the characteristics of the three indicators, the following definition is made. The first definition: For a given music set S and user set U , if user u has played song s , the edge weight of the graph can be calculated according to equation (13); otherwise, the edge weight $w_{PAF}(u,s)$ of the graph can be calculated according to equation (14).

$$w_{PAF}(u,s) = PAF(u,s) \tag{13}$$

After $w_{PAF}(u,s)$ is determined, the bipartite graph in Figure 4 can be obtained. The elements A, B, C, D, and a, b, c, d, e in Figure 4 represent the user vertex set and music vertex set, respectively.

$$w_{PAF}(u,s) = PopAF(u,s) \tag{14}$$

Similarly, this study sets the following definitions. According to the user set and music set, a weighted undirected graph can be obtained, and its corresponding edge weight $w_{TAF}(u,s)$ is calculated according to equation (15).

$$\begin{cases} w_{TAF}(u,s) = TAF(u,s) \\ w_{TAF}(u,s) = PopAF(u,s) \end{cases} \tag{15}$$

At this point, the bipartite graph of the behavior characteristics driven by the tag ca be gotten, as shown in Figure 5.

Figure 4. Schematic diagram of dichotomy

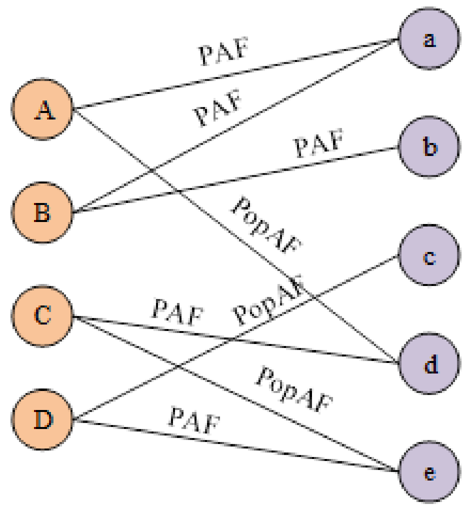
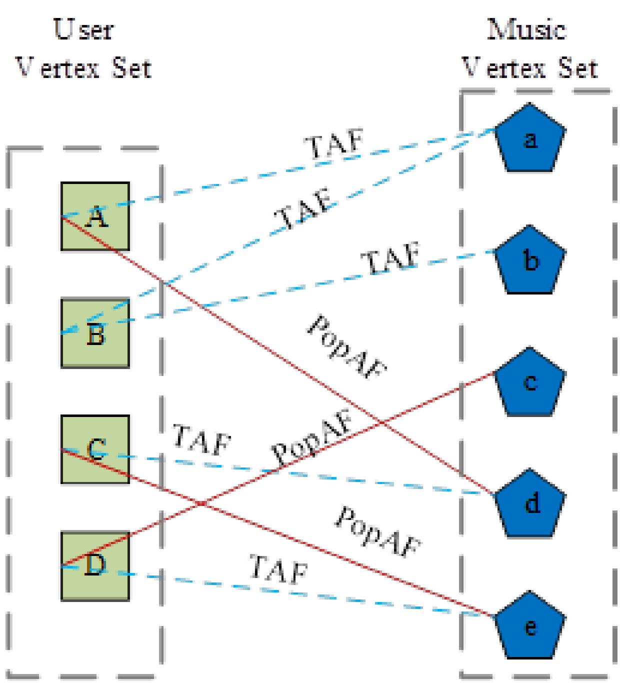


Figure 5. Bipartite graph of behavior characteristics



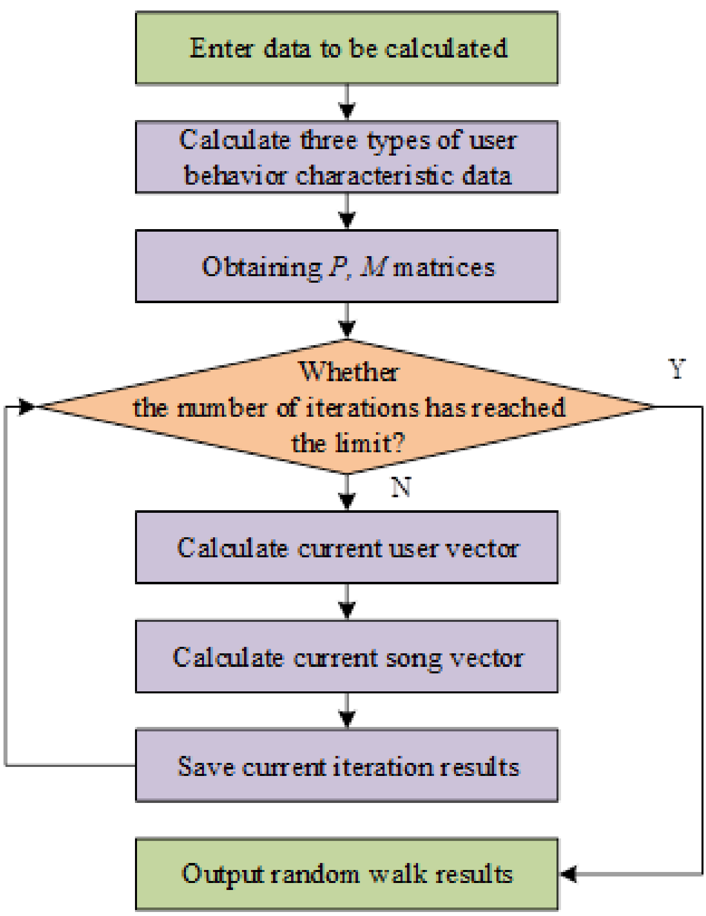
Most current random walk algorithms fail to take into account the behavior characteristics of the three driving modes concerned in this study. Therefore, this research fuses these three behavior characteristics in an equal weight manner. And it proposes an improved random walk algorithm based on user behavior characteristics. The following will elaborate on it (Politis D et al., 2015).

Assuming that there are m users and n songs, the generated adjacency matrix is M, and w_{ij} represents the edge weight of the j-th song of user i. Therefore, the random walk algorithm based on user behavior characteristics can be expressed in equation (16).

$$\begin{cases} x_s^{k+1} = \left(\alpha M_{col}^T + \left(1-\alpha\right)\delta_1\right) x_u^k \\ x_u^k = \left(\alpha M_{row} + \left(1-\alpha\right)\delta_2\right) x_s^k \end{cases} \tag{16}$$

x_s^{k+1}, x_u^k is the calculated value of song s after random walk in step $k+1$, and the calculated value of user u after random walk in step k; M_{col}^T, M_{row} represents the transpose of matrix M and the row random matrix of M, respectively; δ_1, δ_2 represents the $m \times n$ size matrix with all elements $1/m$ and $1/n$, respectively; α is the transmission probability. To sum up, the calculation flow of the random walk algorithm based on user behavior characteristics can be shown in Figure 6.

Figure 6. Random walk algorithm



PERFORMANCE TESTING OF IMPROVED RECOMMENDATION ALGORITHMS IN ONLINE MUSIC TEACHING

In this study, the commonly used dataset for music based machine learning tasks, Last.fm, was selected for performance testing experiments. This dataset contains rich song tags and playback information of a group of users on these songs. The dataset is extracted from the backend of a certain online music teaching platform. The selected dataset contains 9,330 songs, 992 users, and 3,769,957 pieces of user playback data generated by these users. This study divided all data into experimental and test sets in a 7:3 ratio. The testing experiment was conducted on a Unix operating system with 8GB of memory and a CPU running in the Intel Core I7 2G Hz frequency model. The computational logic in the experiment was developed using Python 3.0.

Experimental Plan Design

To compare the recommendation performance of the designed music recommendation systems in online teaching environments, three comparison recommendation systems were designed: user based collaborative filtering (UB-CF) algorithm, tag driven item similarity recommendation (TSR) algorithm, and collaborative tag based recommendation (CTR) algorithm. The recommendation algorithm designed in this study is abbreviated as XCF. The core idea in the UB-CF recommendation system is to calculate user similarity and find neighboring users of vertex users based on their shared ratings. Then the system determines the recommended item items of the target user through domain users. The co-occurrence label of music is used in the TSR algorithm to replace the user's co-scoring and is used to calculate similarity. The algorithm also uses the song playback count indicator as implicit feedback on the user's level of liking for the song. The CTR algorithm is similar to the TSR algorithm, but the former considers user preference implicit feedback more and uses the singer's own attributes to repeatedly label information. To fully compare the performance of various recommendation systems, this study selected accuracy, precision, recall, computational time, and computational memory consumption as comparison indicators. The parameters of each recommended model are determined through a combination of industry experience and multiple debugging. The parameter θ in XCF needs to be determined through multiple trial runs, and the trial run should be carried out according to the accuracy calculation method recommended in the first three. The trial run results are shown in Table 1.

The visual effect of Table 1 is not sufficient, so we can apply Table 1 to draw Figure 7. The horizontal axis of Figure 7 represents the parameter values, while the left and right vertical axes represent the precision and recall values of the calculation scheme. Analyzing Figure 7, it can be seen that as the parameter values increase, the recommended precision and recall values of the model gradually increase. However, when the parameter value is 0.7, the two indicators reach their maximum values. Therefore, a value of 0.7 is more appropriate for parameter θ. The other parameters in XCF are also obtained in this way.

Table 1. Calculation results of parameter θ

Parameter Value	Precision	Recall	Parameter Value	Precision	Recall
0.1	0.58	0.49	0.5	0.68	0.59
0.2	0.60	0.51	0.6	0.70	0.61
0.3	0.62	0.53	0.7	0.71	0.60
0.4	0.65	0.55	0.8	0.68	0.57

Comparison of Performance Test Results

The comparison of training effects of various recommendation systems on the training set is shown in Figure 8. In Figure 8, the horizontal axis represents the number of iterations, the vertical axis represents the objective function value, and different line styles represent different recommendation system models. To increase the reliability of the comparison results, we import the generative adversarial networks (GAN) code on the GitHub platform to build a recommendation algorithm as a comparison method. Analyzing Figure 8, it can be seen that there was no significant rebound in the objective function values of each recommended model during the training process, indicating that the parameter settings of each model were relatively reasonable, and the training results were comparable. As the number of iterations increases, the objective function values of each model first rapidly decrease, then the rate of decline gradually slows down, and finally converges to a certain value. When the number of iterations reaches 200, all models complete convergence, and the objective function values of XCF, CTR, TSR, UB-CF, and GAN recommended models are 0.41, 0.65, 0.69, 1.83, and 0.49, respectively. From the perspective of training set data alone, it can be seen that the XCF online music recommendation teaching system designed in this study has the best training effect; the recommendation model generated from GitHub platform code has the slowest training speed.

The performance of each recommendation model on the test set is analyzed below, as shown in Figure 9. The horizontal axis in Figure 9 represents the number of users participating in the testing experiment, with a maximum value of 297 users, which is the complete test set. The subgraph (a) represents the accuracy comparison results of each recommendation model. The subgraph (b) represents the comparison results of TOP5 recommendation accuracy for each model. The vertical axes of the two subgraphs also represent different test indicators. Analyzing Figure 8, it can be seen that as the number of user samples participating in the test increases, the fluctuation range of recommendation accuracy indicators of each recommendation model gradually decreases. This is because when there are few recommended users, due to the user's own characteristics, the algorithm may not be able to accurately recommend their favorite music according to the overall method. For

Figure 7. Calculation results of parameter θ

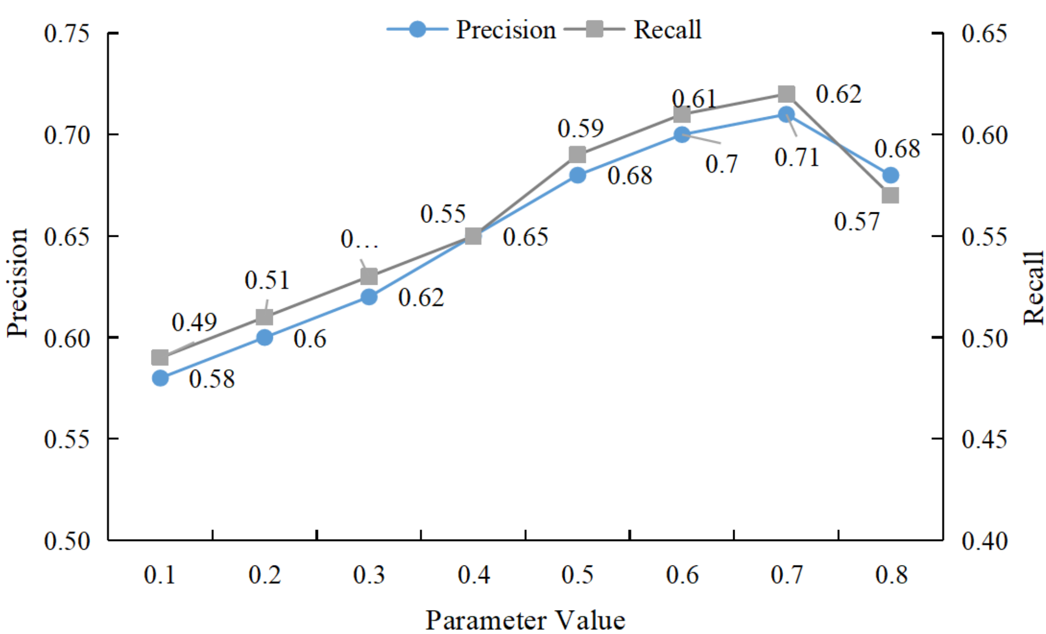


Figure 8. Training effect of recommendation system

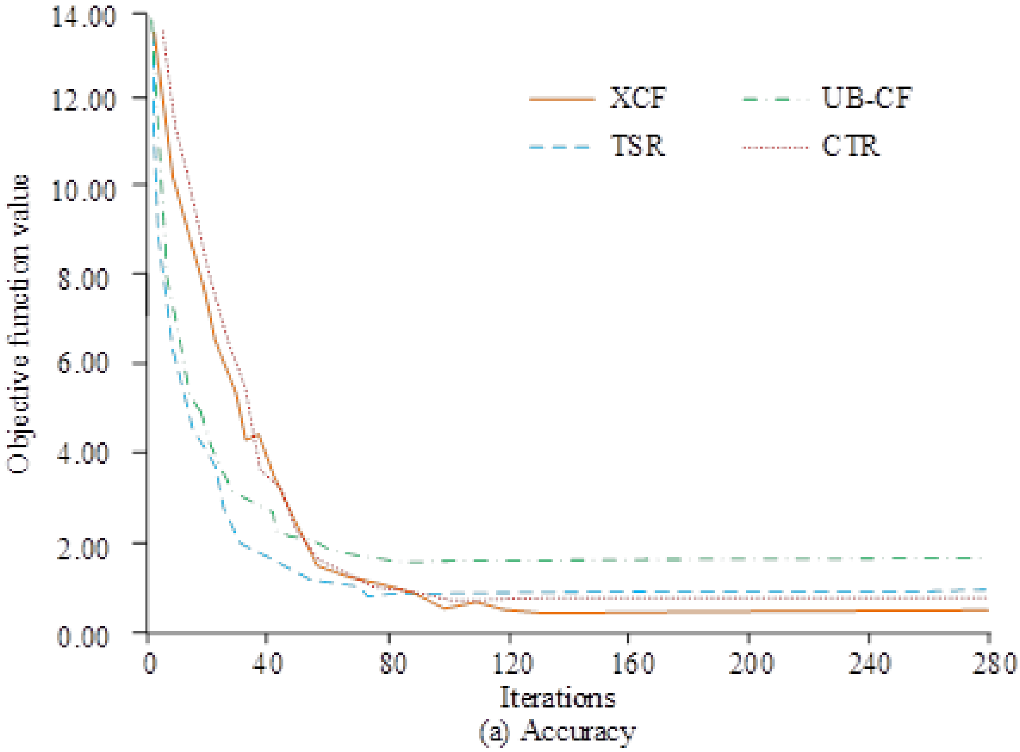
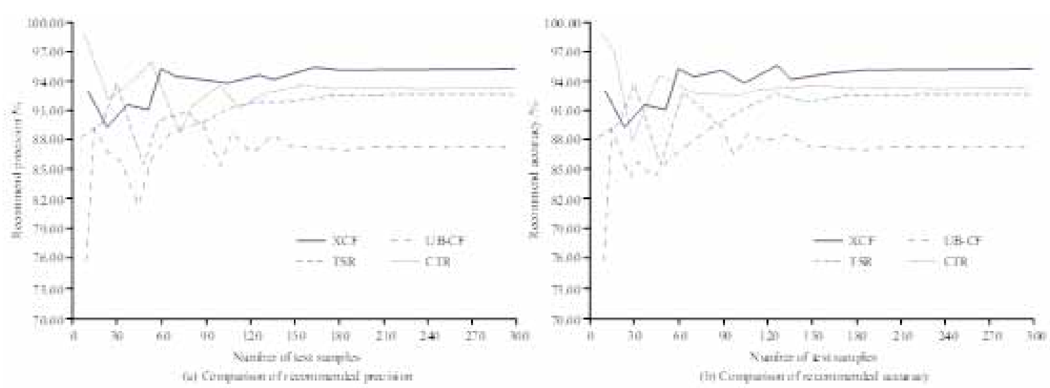


Figure 9. Comparison of TOP5 recommendation accuracy



example, when the number of users to be tested is 6, the TOP5 recommendation accuracy and accuracy of the TSR recommendation model are 75.63% and 75.88%, respectively. But after the convergence of the recommendation accuracy of each algorithm, the recommendation model designed in this study has the highest recommendation accuracy. When the number of users to be tested is the entire test set, the recommendation precision of XCF, CTR, TSR, and UB-CF recommendation models is 95.82%, 92.51%, 91.74%, and 87.26%, respectively. At this time, the recommendation accuracy of XCF, CTR, TSR, and UB-CF recommendation models is 95.75%, 92.42%, 91.66%, and 87.17%, respectively. The research results obtained by Zou et al. (2021) also indicate that teaching recommendation systems that consider user behavior characteristics have higher recommendation accuracy.

This study further analyzes the accuracy of each recommendation model under different recommended music quantity conditions. The statistical results are illustrated in Figure 9. All experimental schemes in Figure 10 were conducted on the entire test set. The horizontal axis in Figure 10 represents different recommended music quantity schemes; the vertical axes of subgraphs (a) and (b) represent recall and accuracy, respectively. Accuracy is represented as a percentage. Different flags represent different recommendation algorithms; different styles of lines represent regression curves of recommendation accuracy data for different recommendation algorithms. Observing Figure 10, it can be concluded that when the number of recommended music samples is small, the recall rate increases rapidly, while the accuracy rate slowly decreases. When the number of recommended music samples exceeds 4, the growth rate of recall rate significantly slows down, while the decrease rate of accuracy gradually accelerates. When the number of recommended music samples exceeds 7, the recall rate begins to slowly decrease, while the accuracy rate rapidly decreases. Overall, when the number of recommended music samples is 8, the XCF, CTR, TSR, and UB-CF recommendation models all have the highest recommended recall rates, with 97.82%, 95.26%, 93.95%, and 88.72%, respectively. The recommendation accuracy is highest when the number of recommended music samples is 1, and the recommendation accuracy of XCF, CTR, TSR, and UB-CF recommendation models is 97.28%, 94.73%, 93.14%, and 88.09%, respectively. Li et al. (2021) found that when the number of recommended subjects is 8, the recommendation accuracy reaches its maximum value. This is because the research subjects of the reference are not limited to a specific course field, and therefore more recommendations are needed to maximize the recommendation effect (Li et al., 2021).

Figure 10 showcases that when the recommended music quantity is 5, the recommendation accuracy and recall of each model are higher. Therefore, in subsequent research, the recommended parameter for music quantity is fixed at 5. The receiver operating characteristic curve (ROC) of each model under the recommended music TOP5 conditions is analyzed as shown in Figure 11. The horizontal and vertical axes in Figure 11 represent the false positive rate and true positive rate, respectively, and different curve types represent different recommended models. Analyzing Figure 11, it was found that the ROC curve area under curve (AUC) of all recommended models was significantly lower than 0.50. This indicates that the TOP5 music recommendation accuracy of each recommendation model has a certain degree of authenticity. Specifically, the ROC curves AUC of XCF, CTR, TSR, and UB-CF recommended models are 0.7, 0.68, 0.64, and 0.57, respectively. The ROC curve AUC of the XCF recommended model designed in this study is higher than that of all comparative models.

Figure 10. Comparison of recommendation accuracy

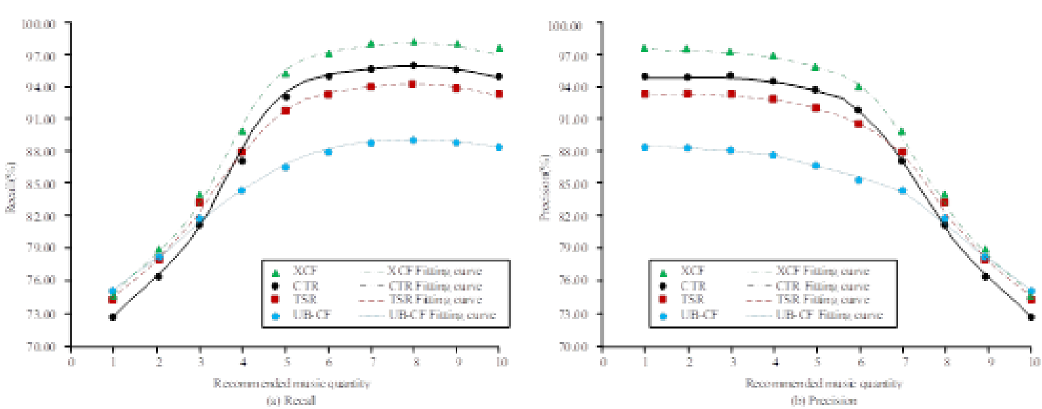
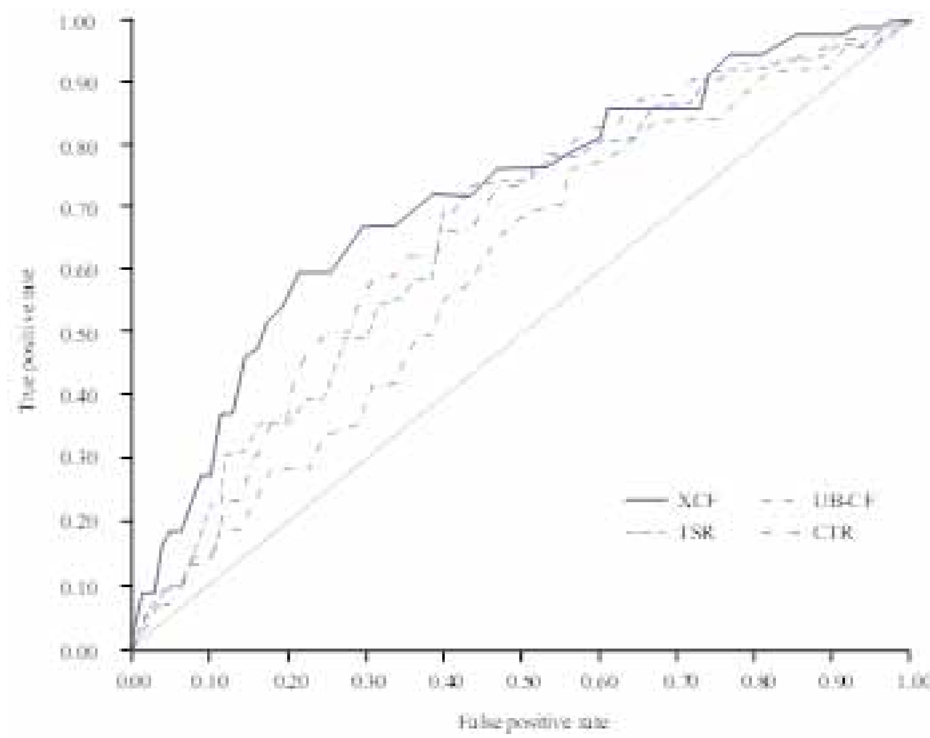


Figure 11. Recommended ROC curve for TOP5



Finally, this study analyzes the computational time and resource consumption levels of each comparative model, as shown in Table 2. Table 2 showcases that as the number of computing users increases, the average computing time and average computing resource consumption of each recommendation model show a linear growth trend. However, for the same model, the change slopes of the former are significantly higher than those of the latter. The XCF recommended model designed in this study has higher computational time and memory consumption compared to other models. This is related to incorporating three user behavior features into the model and using the fusion of trust based on social relationships to calculate similarity metrics. However, the computational efficiency and resource consumption of the XCF recommendation model are still within the range that most computing devices can meet. Specifically, when the number of samples to be calculated is 297 (i.e. the complete test set), the average calculation time for XCF, CTR, TSR, and UB-CF recommendation

Table 2. Calculation time and resource consumption

Number of Users to Be Recommended	Average Recommended Time/ms				Average Memory Consumption/MB			
	XCF	CTR	TSR	UB-CF	XCF	CTR	TSR	UB-CF
60	32	27	22	18	106.1	72.8	68.5	24.5
120	63	53	45	35	106.5	73.0	68.7	33.7
180	95	80	67	55	106.8	73.3	69.1	42.8
240	127	108	91	72	107.0	73.6	69.3	51.6
297	160	136	114	88	107.3	74.0	69.7	60.5

models is 160ms, 136ms, 114ms, and 88ms, respectively. When the number of samples to be calculated is 297, the memory consumption of XCF, CTR, TSR, and UB-CF recommendation models is 107.3MB, 74.0MB, 69.7MB, and 60.5MB, respectively.

CONCLUSION

This research designed a collaborative filtering algorithm that integrates user social relations and behavior characteristics to solve the insufficient accuracy of students' music appreciation recommendation in online music teaching. Then this study conducted a testing experiment based on user behavior data from real online music playback platforms. The research results are as follows: after all models have completed convergence, the objective function values of XCF, CTR, TSR, and UB-CF recommended models are 0.41, 0.65, 0.69, and 1.83, respectively. When the number of users to be tested is the entire test set, the recommendation precision of XCF, CTR, TSR, and UB-CF recommendation models is 95.82%, 92.51%, 91.74%, and 87.26%, respectively. At this time, the recommendation accuracy of XCF, CTR, TSR, and UB-CF recommendation models is 95.75%, 92.42%, 91.66%, and 87.17%, respectively. When the recommended music quantity exceeds 7, the recall rate begins to slowly decrease, while the accuracy rate rapidly decreases. When the recommended music quantity is 8, the recommended recall rates of XCF, CTR, TSR, and UB-CF recommendation models all reach their maximum, reaching 97.82%, 95.26%, 93.95%, and 88.72%, respectively. The recommendation accuracy is highest when the recommended music quantity is 1. At this time, the recommendation accuracy of XCF, CTR, TSR, and UB-CF recommendation models is 97.28%, 94.73%, 93.14%, and 88.09%, respectively. When the recommended music quantity is 5, the recommendation accuracy and recall of each model are relatively high. The ROC curves AUC of XCF, CTR, TSR, and UB-CF recommended models are 0.7, 0.68, 0.64, and 0.57, respectively. The average computation time and memory consumption of XCF, CTR, TSR, and UB-CF recommended models are 160ms, 136ms, 114ms, 88ms, and 107.3MB, 74.0MB, 69.7MB, and 60.5MB, respectively. The following is the discussion section of this study: the online music teaching recommendation model designed in this study is more accurate. This model is conducive to improving the enjoyment of online music teaching and provides students with more personalized and suitable learning materials. It plays an important role in reducing the dropout rate of students' music online courses and improving learning quality. However, the limitation of this study is that there is only one selected research project material, which fails to fully verify the universality of the designed model in music online teaching recommendations. This may limit the market-oriented application of this study, which is also the scope of further research in the future.

REFERENCES

Choi, Y. H., Lee, J., & Yang, J. (2022). Development of a service parts recommendation system using clustering and classification of machine learning. *Expert Systems with Applications, 188*(Feb), 116084.1-116084.12.

Cui, Y. (2021). Intelligent recommendation system based on mathematical modeling in personalized data mining. *Mathematical Problems in Engineering, 2021*(3), 6672036.1-6672036.11.

Da'U, A., Salim, N., & Idris, R. (2021). An adaptive deep learning method for item recommendation system. *Knowledge-Based Systems, 213*(8), 106681.1-106681.12.

Da'U, A., Salim, N., & Idris, R. (2021). Multi-level attentive deep user-item representation learning for recommendation system. *Neurocomputing*, *433*, 119–130. doi:10.1016/j.neucom.2020.12.043

Gwadabe, T. R., & Liu, Y. (2022). Improving graph neural network for session-based recommendation system via non-sequential interactions. *Neurocomputing*, *468*, 111–122. doi:10.1016/j.neucom.2021.10.034

Jiang, N., Gao, L., Duan, F., Wen, J., Wan, T., & Chen, H. L. (2021). SAN: Attention-based social aggregation neural networks for recommendation system. *International Journal of Intelligent Systems*, *37*(6), 3373–3393. doi:10.1002/int.22694

Li, H., Zhong, Z., Shi, J., Li, H., & Zhang, Y. (2021). Multi-objective optimization-based recommendation for massive online learning resources. *IEEE Sensors Journal*, *21*(22), 25274–25281. doi:10.1109/JSEN.2021.3072429

Li, L., Zhang, Z., & Zhang, S. (2021). Hybrid algorithm based on content and collaborative filtering in recommendation system optimization and simulation. *Scientific Programming, 2021,* 742709.1-7427409.11.

Mohammadi, N., & Rasoolzadegan, A. (2022). A two-stage location-sensitive and user preference-aware recommendation system. *Expert Systems With Applications, 191,* 116188.116188.25.

Peng, B. (2021). Research and implementation of electronic commerce intelligent recommendation system based on the fuzzy rough set and improved cellular algorithm. *Mathematical Problems in Engineering, 2021,* 6671219.1-6671219.8.

Politis, D., Piskas, G., Tsaligopoulos, M., & Kyriafinis, G. (n.d.). variPiano™: Visualizing musical diversity with a differential tuning mobile interface. *International Journal of Interactive Mobile Technologies (iJIM), 9*(3), 58-63.

Rabiu, I., Salim, N., Da'U, A., & Nasser, M. (2022). Modeling sentimental bias and temporal dynamics for adaptive deep recommendation system. *Expert Systems with Applications, 191,* 116262.1-116262.15.

Ravakhah, M., Jalali, M., Forghani, Y., Sheibani, R. (2021). Balanced hierarchical max margin matrix factorization for recommendation system. *Expert Systems, 39*(4), e12911.1-e12911.14.

Sharmila, A. H., & Jaisankar, N. (2021). Edge intelligent agent assisted hybrid hierarchical blockchain for continuous healthcare monitoring & recommendation system in 5G WBAN-IoT. *Computer Networks, 200,* 108508.1-108508.18.

Sturluson, A., Raza, A., Mcconachie, G. D., Siderius, D. W., Fern, X. Z., & Simon, C. M. (2021). A recommendation system to predict missing adsorption properties of nanoporous materials. *Chemistry of Materials*, *33*(18), 7203–7216. doi:10.1021/acs.chemmater.1c01201

Vo, N. D., Hong, M., & Jung, J. J. (2020). Implicit stochastic gradient descent method for cross-domain recommendation system. *Sensors (Basel)*, *20*(9), 2510–2525. doi:10.3390/s20092510 PMID:32365513

Wang, N. (2021). Ideological and political education recommendation system based on AHP and improved collaborative filtering algorithm. *Scientific Programming, 2021,* 2648352.12648352.9.

Zeeshan, Z., Ain, Q. U., Bhatti, U. A., Memon, W. H., Ali, S., Nawaz, S. A., Nizamani, M. M., Mehmood, A., Bhatti, M. A., & Shoukat, M. U. (2021). Feature-based multi-criteria recommendation system using a weighted approach with ranking correlation. *Intelligent Data Analysis*, *25*(4), 1013–1029. doi:10.3233/IDA-205388

Zhang, W., Zhang, X., & Chen, D. (2021). Causal neural fuzzy inference modeling of missing data in implicit recommendation system. *Knowledge-Based Systems, 222*(11), 106678.1-106678.12.

Zhao, Z., Zhang, X., Zhou, H., Li, C., Gong, M., & Wang, Y. (2020). HetNERec: Heterogeneous network embedding based recommendation. *Knowledge-Based Systems, 204*(8), 106218.1-106218.10.

Zhu, Q. (2021). Network course recommendation system based on double-layer attention mechanism. *Scientific Programming, 2021*, 7613511.1-7613511.9.

Zou, F., Chen, D., Xu, Q., Jiang, Z., & Kang, J. (2021). A two-stage personalized recommendation based on multi-objective teaching–learning-based optimization with decomposition. *Neurocomputing*, *452*(6), 716–727. doi:10.1016/j.neucom.2020.08.080

Zy, A., Feng, Z. A., Lyb, C., & Qian, Z. (2020). A new prediction method for recommendation system based on sampling reconstruction of signal on graph. *Expert Systems with Applications, 159,* 113587.1-113587.12.