Journal of Network Intelligence
Taiwan Ubiquitous Information

©2025 ISSN 2414-8105
Volume **10**, Number **3**, August **2025**

Personalised Music Recommendation Algorithm Based on Multi-source Data Fusion and Attention Mechanism

Wen-Lu Luo[1,*], Yu-Han Wang[2]

[1]Nanchang Institute of Technology, Nanchang 330000, P. R. China
18607096886@163.com

[2]International College, Krirk University, Bangkok 10220, Thailand
zitong0825@gmail.com

[*]Corresponding author: Wen-Lu Luo

Received July 10, 2024, revised December 3, 2024, accepted February 29, 2025.

Abstract. *As the Internet becomes more and more integrated with people's productive lives, the total amount of data and information around the world has exploded. To save users' time in searching for their favourite music among many music, music recommendation services have emerged. The traditional music recommendation method has the issues of low recommendation accuracy and poor real-time performance. To address the above issues, this article suggests a personalised music recommendation algorithm relied on multi-source data fusion and attention mechanism. Firstly, the matrix factorization is used to optimize the multi-source data fusion algorithm, and the weights are introduced into the association network of the music homogeneous data sources, and the similarity of the weights as well as the relationship is optimised in the solving process, so that the heterogeneous information can be retained maximally. Then, the multi-source data of music is preprocessed and the homogeneous data is fused, followed by feature extraction of the fusion result using deep neural network to establish the feature system of multi-source data, and enhance the critical weight of the multi-source data features in the recommendation through multi-layer attention to achieve more accurate user music preference. Finally, the probability of the user's preference for the target music is predicted and the recommendation list is generated. Experimental outcome on dataset MSD implies that NDCG@20, MRR@20 and HR@20 of the model reach 0.5685, 0.5194 and 0.8124, which is superior to the existing algorithms and has excellent recommendation performance.*

Keywords: Personalized music recommendation; Multi-source data fusion; Attention mechanism; Deep neural network; Matrix factorization

1. **Introduction.** In the era of big data, the rapid growth of information on the Internet has aggravated the phenomenon of "information overload", and recommender systems, which can build a personalized interest model for users by analyzing their past behavioral characteristics and interest preferences, and help users find information quickly and accurately, have been playing an increasingly important role in recent years [1, 2]. Music is an important way for human beings to express their emotions, and it is a kind of art to support the emotions of life. In modern society, people tend to listen to music as a form of daily leisure and entertainment. In the field of music, it is suitable to introduce a recommendation system, accurate music recommendation can not only improve the user experience, but also bring traffic for music websites to create better business value

[3]. However, the recommendation function of traditional music software suffers from the problems of lack of personalization, cold start and sparse data [4, 5]. These problems affect the accuracy of recommendation results, which in turn affects the user experience. Therefore, it is significant to study personalized music recommendation methods that suit users' tastes.

1.1. **Related work.** Zheng et al. [6] suggested a dynamic music recommendation framework that dynamically integrates label information of music tracks with temporal dynamics into user-item interactions to achieve personalized music recommendation. Bogdanov et al. [7] performed personalized music recommendation based on Mel Frequency Cepstrum Coefficients (MFCC) and traditional Gaussian mixture models. Idrissi et al. [8] used latent factor recommendation for video to recommend many music songs. Chordia et al. [9] adopted Hidden Markov Model for prediction of music successions to personalize music recommendation for users.

Recently, deep learning models have been gradually utilized in music recommendation. Poulose et al. [10] fused deep belief networks and probabilistic matrix decomposition to extract vectorized features from spectrograms for personalized recommendation. Xia [11] offered a user-engrafted personalized music recommendation algorithm relied on BPNN. Dai et al. [12] modeled the music preference of users by feeding their personal information and music listening history into a deep neural network to push music to the users in accordance with their interests. Abdul et al. [13] used Convolutional Neural Networks (CNNs) to model the music preference of users to match their interests. Costa et al. [14] used a CNN to extract the temporal and frequency features from the music audio. Bai [15] suggested a convolutional matrix factor decomposition model to analyze the obscured features of music audio signals to address the system cold-start issue. However, CNNs do not have memory and cannot extract effective features for data with temporal connections, which are important for music recommendation systems.

The birth of Recurrent Neural Network (RNN) solved this issue. Zhang [16] built an RNN model for personalized music recommendation by learning user's historical preferences. Shafqat and Byun [17] used LSTM to capture characteristics of music audio and lyrics for the goal of computing music similarity to gain recommendation. However, owing to the variety of features decomposed from the audio signal, how to reasonably allocate the arithmetic power to the user's favourite features is also an urgent issue to be addressed. Wang et al. [18] offered a deep music recommendation algorithm combining music features based on the attention mechanism. Wang et al. [19] incorporated the attention mechanism relied on the deep network DNN to achieve personalized recommendation of music, but there is sparsity in the data. However, the data is sparse. In recommender systems, fusion of multi-source auxiliary information is the main feasible algorithm to address the issue of data sparsity, but there are only a few researches in the field of music recommendation that introduce multi-source data fusion. Li and Gan [20] suggested a method of fusion of multi-source user interest data under a tree network, to efficiently push the personalized music to the users.

1.2. **Contribution.** In summary, the existing personalized music recommendation algorithm has the issues of large recommendation error, poor real-time performance and cold start of the system. To cope with the above issues, this article suggests a personalized music recommendation algorithm relied on multi-source data fusion and attention mechanism. Firstly, based on the similarity-based matrix factorization, the multi-source data fusion algorithm is optimized to maximize the retention of heterogeneous data. Then, the audio data of music, the text emotion data and the time series data of user behavior are preprocessed, and the optimized multi-source data fusion algorithm is used to fuse the homogeneous data, and feature extraction is performed on the fusion results to establish the feature system of multi-source data, and then the attention mechanism is used to enhance the critical weights of the multi-source data features in the recommendation in order to obtain more accurate users' music preference. Finally, the probability of user's choice for the target music is predicted, and a music recommendation list is gained by softmax.

2. Theoretical analysis.

2.1. Recurrent neural network.

RNN is a special neural network framework that can process sequence-type data, such as time series, speech sequences, etc [21]. However, RNN does not have the ability to process text of excessive length. Thus, the processing of long sequence data usually requires the use of techniques such as Long Short-Term Memory (LSTM) or Gate Recurrent Unit (GRU).

LSTM is a commonly used variant of RNN basis, which can well solve the above shortcomings of RNN, not only can learn more long-distance context information, but also can reduce the gradient explosion problem of RNN when dealing with high-dimensional information [22]. The gating mechanism is added to LSTM to control the transmission of information, which are the input gate, the output gate, and the oblivious gate, in which the oblivious gate is used to prevent information overload, as shown in Equation (1).

$$f_t = \delta(V_f x_t + W_f g_{t-1} + a_f) \tag{1}$$

The input gate is used to feed new information to the obscured level, as implied below.

$$i_t = \delta(V_i x_t + W_i g_{t-1} + a_i) \tag{2}$$

The output gate decides which information in the obscured level is output to the outside, as implied below.

$$o_t = \delta(V_o x_t + W_o g_{t-1} + a_o) \tag{3}$$

where V, W, and a are the weight parameters of the different gating mechanisms, and $\delta(\cdot)$ is the activation function, in general a Sigmoid function.

2.2. Attention mechanism.

The core aim of the attentional mechanism is to choose from a large amount of the data that is more important to the existing task aim as a way to improve accuracy and reduce algorithmic complexity by removing the redundancy of information [23], and its internal structure is implied in Figure 1.

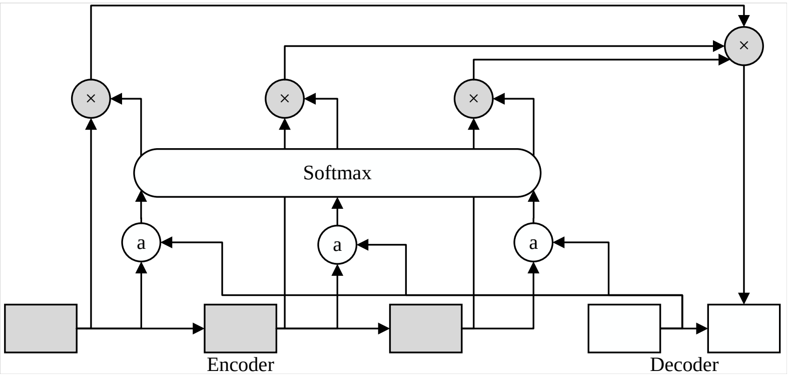

Figure 1. Internal structure of attentional mechanisms

AM boils down to a redistribution of the weights of the input data, whereas the input data between layers is processed with respect to the hermitian state of the Encoder at all time steps, as implied in the following equation.

$$O = \operatorname{softmax}(QK^T)V \tag{4}$$

where Q denotes query, K denotes Key and V denotes Value.

Each element in the source target is understood as a set of key-value pairs of Key and Value. Given a data in the target, by calculating the correlation between this data and each key, the weight coefficients of each key corresponding to value are obtained, and then the values are weighted and summed to finally obtain the weight value of the attention mechanism, as implied in Equation (5).

$$Attention(Tar, Sour) = \sum_{i=1}^{L_x} \mathrm{Sim}(Tar, K_i)V_i \tag{5}$$

where $L_x = \mathrm{Sour}$ represents the length of the source target.

3. **Similarity-based fusion of music multi-source data.** Intending to the lack of differentiation of data sources in multi-source data fusion algorithms, this article introduces weights to the correlation network of music homogeneous data sources, and optimizes the weights of the homogeneous network and the similarity of the relationship in the solution process, so as to maximize the retention of value information. The overall flow of the music multi-source data fusion algorithm is implied in Figure 2.

The music data source is distinguished into different object types, labelled $\sigma_1, \ldots, \sigma_s$, indicating that there are s object types that have interactions. The relationship between type i and type j is denoted (σ_i, σ_j). The data related to (σ_i, σ_j) in the data source is denoted as a sparse relation matrix $r_{ij} \in \mathbb{R}^{n_i \times n_j}$. Data of different object types are heterogeneous and data of the same object type are homogeneous. A multi-source homogeneous relationship between the same type σ is denoted by $\vartheta^{(t)} \in \mathbb{R}^{n_i \times n_j}$, where $t \in \{1, 2, \ldots, t\}$, is a homogeneous associated data source of type i with t sources.

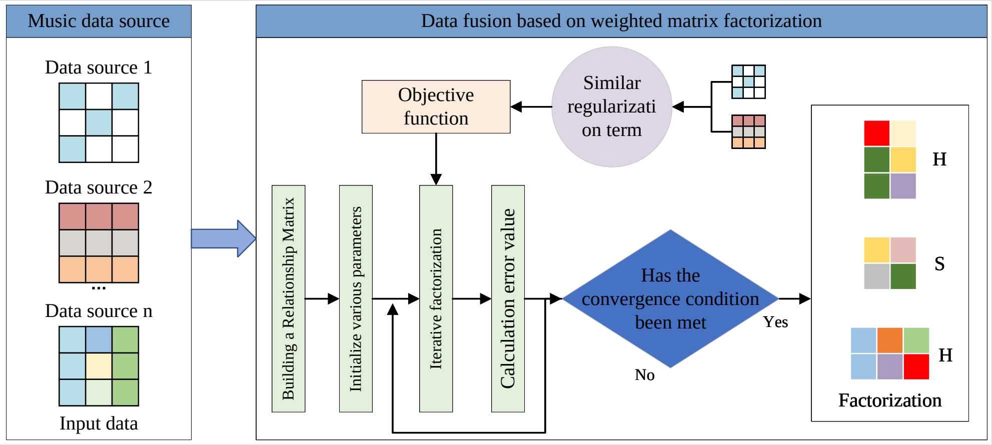
Music data source
Data source 1
Data source 2
...
Data source n
Input data
Data fusion based on weighted matrix factorization
Objective function
Similar regularization term
Building a Relationship Matrix
Initialize various parameters
Iterative factorization
Calculation error value
Has the convergence condition been met
Yes
No
H
S
H
Factorization

Figure 2. The whole flow of the music multi-source data fusion

(1) Object similarity data fusion for different music data types. Suppose that M similarity matrices are constructed for the object types by the similarity function, $S = \{S^{(1)}, \ldots, S^{(M)}\}$. For example, for σ_i and σ_j, whose combined similarity $\sum_{S=1}^{M} S_{i,j}^{(S)}$ value

defined on S is larger, the low-rank matrices H_i and H_j containing their attribute information should also be similar. The addition of similarity is constrained by introducing a similarity regular term, which is shown as follow:

$$O_1 = \frac{1}{s} \sum_{i,j=1}^{s} \|H_i - H_j\|^2 S_{i,j}^{(S)} \tag{6}$$

This article defines the similarity function as the mean of the Euclidean distance similarity [24] and the cosine similarity as follows.

$$S_{ij} = \frac{1}{2} \left(1 + \frac{H_i H_j}{\|H_i\| \|H_j\|}\right) \tag{7}$$

Define the combined similarity as a linear combination of the individual similarity matrices after normalising them, and define the diagonal matrix $D_{i,j}^{(S)}$ to construct the consistent Laplacian matrix $L^* = D^{(S)} - S^{(S)}$. According to Equation (7), the regular terms of the Laplacian are transformed into the trace of the matrix as in Equation (8), and tr() denotes the computation of the trace of the matrix.

$$O_2 = \operatorname{tr}(H^T L^* H) \tag{8}$$

(2) Objective function setting for multi-source data fusion algorithm. In the proposed fusion algorithm, the input data is updated under constraints to minimize the objective function to obtain an approximation of the matrix.

$$O_3 = \sum_{i=1}^{s} \|S_{ij} - H_i S_{ij} H_j^T\|_F^2 + \sum_{i=1}^{s} \sum_{t=1}^{t_i} \operatorname{tr}(H_i^T \vartheta^{(t)} H_i) \tag{9}$$

where $\|S_{ij} - H_i S_{ij} H_j^T\|_F^2$ denotes the approximate computation of the reconstructed relation matrix after decomposition of heterogeneous data sources and the second term is the constraint term for homogeneous data sources.

To add the similarity information between different types of objects to the algorithm, the final objective function is obtained as follows by adding the similarity regular term described as below.

$$O_{\text{SiMF}} = O_3 + \beta O_2 \tag{10}$$

where β is a tuning parameter intended to guide the contribution of the similarity regularity term and the underlying objective function to the overall loss.

For the optimal solution of O_{SiMF}, the iterative formula of H and S is used to iterate the values until the objective function gets the minimum value of convergence, which maximizes the retention of the value information of the heterogeneous data sources.

4. Personalized music recommendation based on multi-source data fusion and attention mechanism.

4.1. Music multi-source data pre-processing.

On the basis of the previous section, this paper suggests a personalized music recommendation algorithm based on multi-source data fusion and attention mechanism. Firstly, the multi-source data of music is preprocessed, secondly, the multi-source data of music is fused, and the fused result is feature extracted to establish the feature system of multi-source data, then the critical weight of the multi-source data features in recommendation is enhanced by the attention mechanism to obtain a more accurate user's music preference, and finally, the probability of the user's choice for the target music is predicted to generate the recommendation list. The model structure of the suggested algorithm is implied in Figure 3.

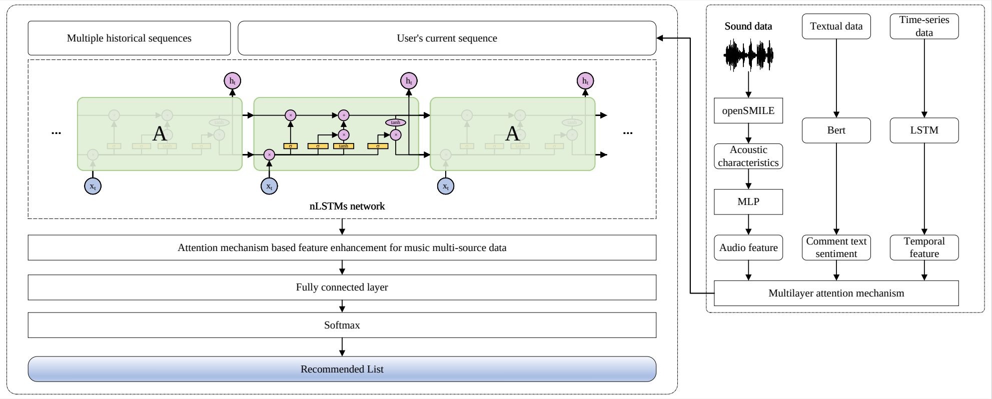

Figure 3. The model structure of the suggested algorithm

There are three types of data in the input layer of the proposed algorithm respectively, the first one is the music related audio data, the time-frequency of the audio is converted using short time Fourier transform as implied in Equation (11). Then the audio amplitude data is captured using the Hamming window method, and finally, the converted spectrogram is downscaled using the Mel filter [25] to generate the Mel spectrogram. Finally, the preprocessed audio data is gained from the Mel filter's energy output as bellow.

$$\mathrm{STFT}_{\lambda}(t,f) = \int_{-\infty}^{+\infty} [\lambda(u)h(u-t)]e^{-j2\pi f u}du \tag{11}$$

where $\lambda(t)$ is the original signal, $h(t)$ is the window function.

The second part is the user's music-related textual sentiment data, including user's gender, user's age, user's nationality, music name, music artist, music label and music duration, etc. This part of the data is converted into M-dimensional word vectors by the word embedding model BERT, and is combined into a matrix through vector splicing to serve as the subsequent input. When using BERT to construct word vectors for the words in the music static data, the vector representations of the corresponding words are obtained from the pre-trained vector space of the corpus data provided by Wikipedia, as implied below.

$$L = -\frac{1}{L}\sum_{l=1}^{L} y_l \log(\mathrm{Softmax}(BAx_l)) \tag{12}$$

where x_l denotes the textual representation of the document based on word vectors, y_l denotes the corresponding feature labels, and A and B are both weight matrices.

The third is the time series data generated by user behavior, taking the user's recent k sequences of music playback as the prediction sequence, i.e., $T_X = [X_{t-l}, X_{t-l+1}, X_{t-l+2}, \dots, X_{t-1}]$, $T \in \mathbb{R}^{k\times n}$ where n denotes the dimension of each sequence, which is a collocation of word vectors of the user's preferences for different music and the names of music, music artists and music labels in the recent k sequences of music playback. For example, if the user plays music A at time t, then $X_A = [\mathrm{score}_A, x_A, p_A, t_A] \in \mathbb{R}^n$, where score_A denotes the preference score, and x_A, p_A, t_A denote the word vectors of music names, music singers, and music labels, respectively, is computed as $\mathrm{score}_i = m_A/k$, where m_A denotes the number of times music A occurs in sequence k.

4.2. **Music multi-source data fusion and feature extraction.** Based on the above pre-processed multi-source data, this section uses the data fusion algorithm in Section III to fuse the homogeneous data from multiple sources of music, assuming that the final fusion result obtained by iteratively solving Equation (10), which contains audio data, text data, and temporal data, and in this paper, we respectively carry out the feature extraction of these multi-source data.

(1) Feature extraction of audio data. In this article, audio-related features are extracted using multilayer perceptron, and the splicing vector x of intensity, loudness, pitch, rhythm, etc. is used as the input to the MLP. Since each neuron in the MLP belongs to a different level, the neurons in each level are able to receive the signals from the previous layer and generate signals for output to the next level. Thus, the audio feature a extracted using MLP is implied bellow.

$$a = \delta(V_d x_h + e_d) \tag{13}$$

where V_d and e_d are the obscured level parameters, x_h denotes the obscured level state of the h-th level, and δ is the activation function.

(2) Feature extraction of text emotion data. In this paper, TextCNN is used for text emotion feature extraction, and the word vectors obtained from the preprocessing part are input into TextCNN, and multiple convolution kernels are used for convolution to learn the local features between sentences and adjacent multiple words.

$$b_i = f(W \cdot x_{i+g-1} + l) \tag{14}$$

where W is the weight of convolution kernel, l is the bias, f is the activation function, and g is the window size. The final feature vectors obtained from the text are implied below.

$$b = [b_1, b_2, \ldots, b_{m-g+1}] \tag{15}$$

(3) Feature extraction of time series data. Bi-LSTM is used for temporal feature extraction. Firstly, each time series data $C \in \mathbb{R}^m$ is inputted to obtain the implied state $c_t^{ur} \in \mathbb{R}^z$ of the forward sequence and the implied state $c_t^{su} \in \mathbb{R}^z$ of the inverse sequence, which are spliced to obtain $c \in \mathbb{R}^{2z}$, where z represents the dimension of the feature vector of the obscured level.

$$c_t^{ur} = \mathrm{LSTM}_{\mathrm{forward}}(C_t, c_{t-1}^{ur}) \tag{16}$$

$$c_t^{su} = \mathrm{LSTM}_{\mathrm{backward}}(C_t, c_{t-1}^{su}) \tag{17}$$

$$c = c_t^{ur} \oplus c_t^{su} \tag{18}$$

4.3. **Attention mechanism based feature enhancement for music multi-source data.** The above extracted feature vectors a, b, and c are first mapped to a dense representation space to obtain a d-dimensional embedding as implied bellow respectively.

$$a_i' = U_a a_i + k_a \tag{19}$$

$$b_j' = U_b b_j + k_b \tag{20}$$

$$c' = U_c c + k_c \tag{21}$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$, $U_a, U_b, U_c, k_a, k_b, k_c$ are embedding parameters.

To better distinguish the influence of features in different directions of music on the preferences of different users, multi-layer attention [26] is applied to the multi-source data, and the contributions of features in different domains are identified and enhanced. Firstly,

$a_{\text{att}} = \sum_{i=1}^{N} \beta_i a'_i$ and $b_{\text{att}} = \sum_{j=1}^{M} \alpha_j b'_j$ are used to weight and sum the audio features and text emotion features respectively, where β_i and α_j are the attention scores of embedded feature vectors a'_i and b'_j respectively. The attention scores are calculated by two-layer attention network as bellow.

$$\beta'_i = u_a^T \delta(U_a a'_i + k_a) + k_a \tag{22}$$

$$\alpha'_j = u_b^T \delta(U_b b'_j + k_b) + k_b \tag{23}$$

where $U_a, U_b \in \mathbb{R}^{t\times d}$, $k_a, k_b \in \mathbb{R}^t$ are the parameters of the first layer, $u_a, u_b \in \mathbb{R}^t$, $k_a, k_b \in \mathbb{R}$ are the parameters of the second layer, and t is the number of layers in the hidden layer. The attention scores β_i and α_j are further normalized by Softmax, $\beta_i = \frac{\exp(\beta'_i)}{\sum_{i=1}^{N}\exp(\beta'_i)}$ and $\alpha_j = \frac{\exp(\alpha'_j)}{\sum_{j=1}^{M}\exp(\alpha'_j)}$ respectively.

While distinguishing the overall contribution of audio sentiment data from textual sentiment data, temporal features were further combined to obtain the final music attention representation.

$$z_{\text{att}} = \gamma_a a_{\text{att}} + \gamma_b b_{\text{att}} + \gamma_c c \tag{24}$$

where γ_a, γ_b, and γ_c are the attention scores of the embedding vectors of audio features, textual sentiment features, and temporal features, respectively, which are computed as $\gamma_a = \frac{\exp(\gamma'_a)}{\exp(\gamma'_a)+\exp(\gamma'_b)+\exp(\gamma'_c)}$, where $\gamma'_a = u_z^T \delta(U_z a_{\text{att}} + k_z) + k_z$. Similarly, γ_b and γ_c can be computed by using the above computation method.

4.4. **Preferred recommendation forecast.** After obtaining the above music attention representation z_{att}, this article uses an LSTM with a fusion attention mechanism to model the recently played music, then the fusion attention representation of multi-source data is implied as bellow.

$$h'_{t-1} = \text{LSTM}(z'_{t-1}) \tag{25}$$

where h'_{t-1} is the hidden layer state of the LSTM incorporating the attention mechanism.

Based on this, at the moment $t-1$, h_{t-1} and h'_{t-1} are learnt according to the traditional LSTM and the LSTM incorporating the attention mechanism, respectively, and the final representation of the user's preference is the average of the two vectors $h^*_{t-1} = \frac{h_{t-1}+h'_{t-1}}{2}$.

After obtaining the user's preference representation, the probability p_{t-1} of the user's preference for the target music is calculated by the Softmax function, and the top N music with higher probability are selected to generate the recommendation list.

$$p_{t-1} = \text{softmax}(U h^*_{t-1}) \tag{26}$$

where $U \in \mathbb{R}^{|M|\times 2d}$ is the trainable projection matrix for all music.

If the probability of music is denoted as $p \in P$, the objective function can be expressed as a log-likelihood as follows.

$$L = -\sum_{k=1}^{N} \log(p_k) \tag{27}$$

where N is the total amount of training instances and p_k is the probability that the model generates music for the k-th training sample.

5. **Performance testing and analysis.**

5.1. **Analysis of the accuracy of the experimental results.** This article uses MSD [27], a dataset provided by Last.fm website, which has a large user group and rich music data. MSD mainly contains user IDs, genders, ages, listening timestamps, music artist IDs, and so on, with a total of 18,172,536 records, and the processing of the above data adopts the file processing of Python, which converts the original file data into the appropriate format and stores it in the database. To make the experimental outcome fair, the parameter settings of the participating experiments are all the same. In this paper, the dimensionality of the dataset vectors is set to 100, the size of the batch-size is set to 100, the size of the memory encoder is set to 100, and the Adam optimizer is adopted with an initial learning rate of 0.001. The processor used in this experiment is AMD Ryzen 5 3550H 2.1GHZ, and the graphics card is AMD Radeon (TM) Vega 8 Graphics, 16G of RAM, the programming language is Python 3.7, and the deep learning framework used is Tensorflow 1.14.0.

To evaluate the recommendation accuracy of the proposed algorithm and the compared models, this paper uses the evaluation metrics Precision@N, Recall@N, NDCG@N, MRR@N and HR@N, which are commonly used in music recommendation methods, to evaluate the recommendation performance of the VLHM algorithm [9], DLAN algorithm [11], RSCN algorithm [13], MCUM algorithm [17], BLCR algorithm [20] and the recommended performance of the proposed model MSAM are evaluated. For the goal of eliminating the instability of individual experimental results, this paper repeats 20 experiments, and the average of the experimental results is used as the final evaluation results. The results of different recommended methods are implied in Table 1. From Table 1, it can be seen that MSAM significantly outperforms the other five comparison models in all evaluation indicators. The Precision@20, Recall@20, NDCG@20, MRR@20, and HR@20 of MSAM are 0.8962, 0.8643, 0.5685, 0.5194, and 0.8124, respectively. The quantitative evaluation results clearly prove the superiority of the suggested method.

Table 1. Performance comparison of MSAM with other models

Algorithm	Precision@20	Recall@20	NDCG@20	MRR@20	HR@10
VLHM	0.6458	0.5842	0.3117	0.1539	0.5729
DLAN	0.6921	0.6194	0.3961	0.2386	0.6352
RSCN	0.7284	0.6617	0.4395	0.3027	0.6971
MCUM	0.7739	0.7481	0.4842	0.3844	0.7283
BLCR	0.8241	0.8055	0.5127	0.4267	0.7514
MSAM	0.8962	0.8643	0.5685	0.5194	0.8124

VLHM does not differentiate between historical and current sequences and exploits their correlation, resulting in poor recommendation performance. DLAN is a hybrid music recommendation method that combines BP neural networks with users' music playback records, but it does not consider the multi-source data of the music, and the features are not sufficiently extracted, resulting in inefficient recommendation. RSCN inputs a single music score into CNN for personalised music recommendation, which neither considers the diversity of data nor enhances the features. MCUM and BLCR are methods based on sequence analysis, which use listening records and other dynamic modelling of user preferences to improve recommendation performance, but do not highlight key features and have feature redundancy, resulting in poorer recommendation than MSAM, which fuses music's multi-source data into sequence modelling and highlights important features, and exhibits the best recommendation performance.

In addition, the weighted summed average F1@N of Precision@N and Recall@N can reflect the recommendation performance of each model more intuitively. The F1@N results

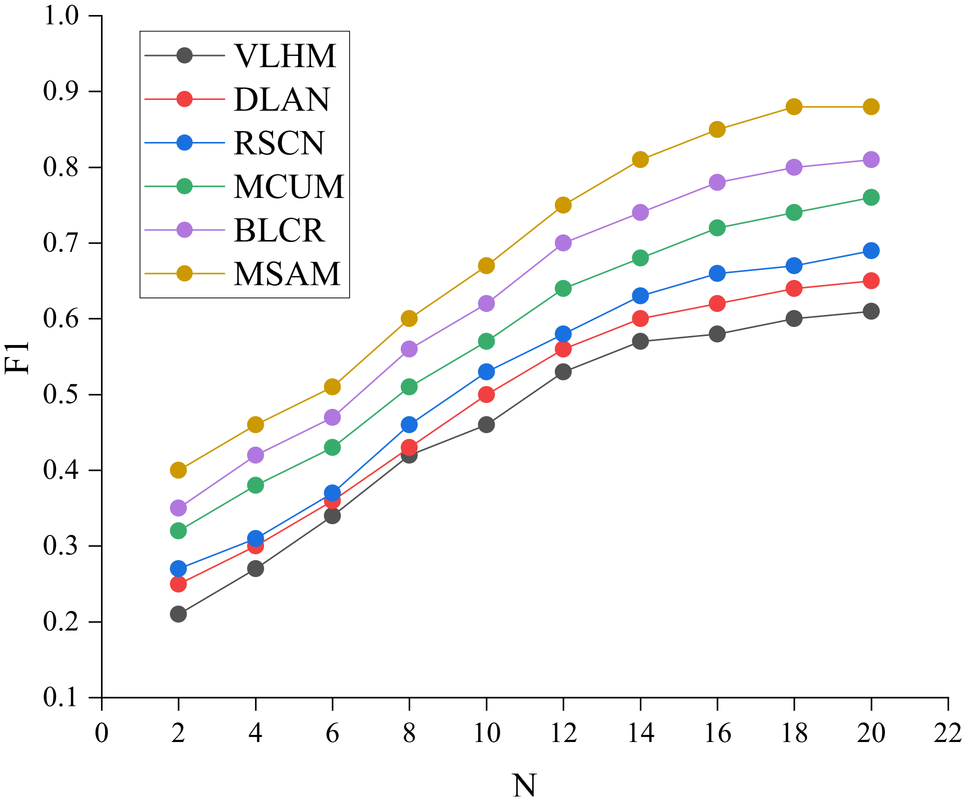

Figure 4. Comparison of F1 values for different algorithms

when N takes different values are implied in Figure 4 for generating a user's recommendation list using the Top N recommendation strategy. When $N = 20$, all six models have the highest F1 values. In addition, from the experimental results of different values of N, it can be seen that when N is taken from 2 to 20, and the range of variation of the F1 value is the smallest among all the models, and with the increase of N, the F1 value of MSAM exceeds that of other models, which indicates that the model has stronger stability.

5.2. **Recommended accuracy analysis.** The capability of the algorithm to forecast scores is an important part of the recommendation performance, and this article uses MSE, RMSE, MAE, MAPE, and MSLE [28] to estimate the prediction effect of the algorithm. A comparison of the recommended accuracies of the different models is implied in Figure 5. The MSE, MAE, and MSLE of MSAM are reduced by 25.73%, 29.61%, and 36.41% compared to VLHM, 20.92%, 19.65%, and 31.05% compared to DLAN, 19.13%, 21.75%, and 27.35% compared to RSCN, 11.02%, 15.20%, and 21.56% compared to MCUM.

VLHM is a collaborative filtering recommendation based on traditional machine learning, and its error is large. DLAN and RSCN both use a single music data source as the input of deep neural network to predict ratings without fusing multi-source data, and the prediction effect is unsatisfactory. MCUM adequately captures the temporal information but does not fuse multi-source data such as the release year of the music, labels, audio, etc. BLCR fuses multi-source user interest data through tree structure fuses multi-source user interest data, but does not enhance important features, and the error is larger than that of MSAM. In summary, MSAM fuses audio data, review text data, and temporal information, and outperforms the other models on the prediction task.

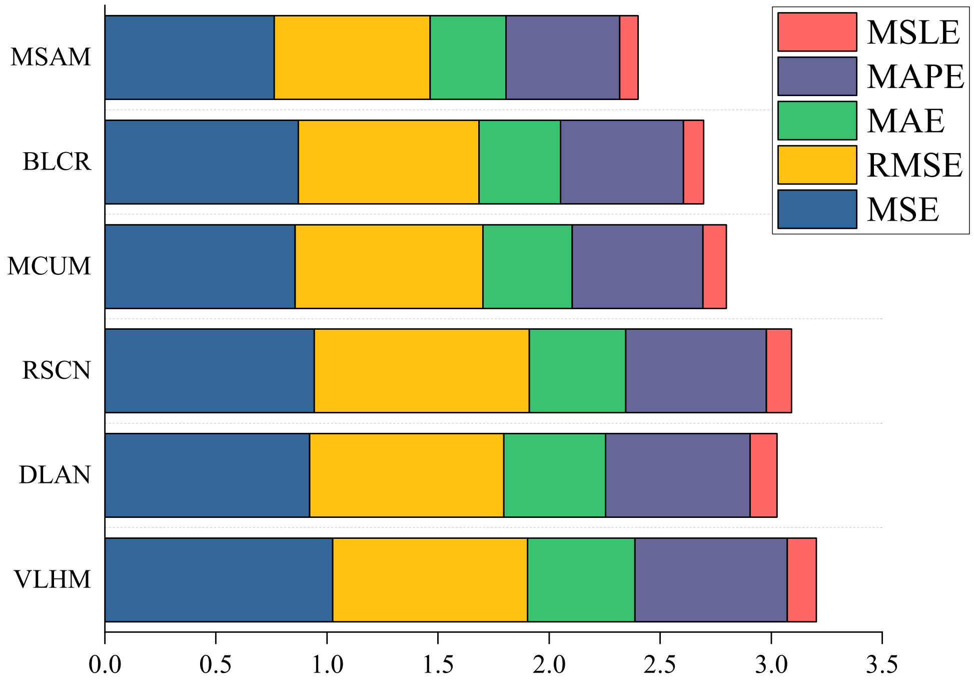

Figure 5. The comparison of the recommended accuracies of the different algorithms

6. **Conclusion.** For the existing personalized music recommendation model has the problems of low recommendation accuracy and poor real-time performance. This article suggests a personalized music recommendation algorithm relied on multi-source data fusion and attention mechanism. Firstly, based on the similarity-based matrix factorization algorithm, the multi-source data fusion algorithm is optimized to maximize the retention of heterogeneous data. Then, audio data of music, text emotion data and time series data of user behavior are preprocessed and the preprocessed homogeneous data are fused, followed by feature extraction of the fusion results using deep neural network algorithm to establish the feature system of multi-source data, enhancement of key features of multi-source data by multi-layer attention to obtain more accurate user music preferences, and finally prediction of the user's probability of the target music preference, select the first N music with higher probability, and generate the recommendation list. The experimental outcome implies that the suggested V has high recommendation accuracy, while the introduction of multi-source data fusion and attention mechanism also positively affects the recommendation effect of the algorithm.

REFERENCES

[1] R. Sharma, and S. Ray, "Explanations in recommender systems: an overview," *International Journal of Business Information Systems*, vol. 23, no. 2, pp. 248-262, 2016.

[2] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.

[3] H. Yu, "A personalised recommendation method of pop music based on machine learning," *International Journal of Reasoning-based Intelligent Systems*, vol. 15, no. 2, pp. 120-127, 2023.

[4] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, pp. 95-116, 2018.

[5] Y. Deldjoo, M. Schedl, and P. Knees, "Content-driven music recommendation: Evolution, state of the art, and challenges," *Computer Science Review*, vol. 51, pp. 100618, 2024.

[6] H.-T. Zheng, J.-Y. Chen, N. Liang, A. K. Sangaiah, Y. Jiang, and C.-Z. Zhao, "A deep temporal neural music recommendation model utilizing music and user metadata," *Applied Sciences*, vol. 9, no. 4, pp. 703, 2019.

[7] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, and P. Herrera, "Semantic audio content-based music recommendation and visualization based on user preference examples," *Information Processing & Management*, vol. 49, no. 1, pp. 13-33, 2013.

[8] N. Idrissi, A. Zellou, and Z. Bakkoury, "KFDBN: Kernelized Finetuned Deep Belief Network for recommendation," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 23599-23634, 2024.

[9] P. Chordia, A. Sastry, and S. Şentürk, "Predictive tabla modelling using variable-length markov and hidden markov models," *Journal of New Music Research*, vol. 40, no. 2, pp. 105-118, 2011.

[10] A. Poulose, C. S. Reddy, S. Dash, and B. J. R. Sahu, "Music recommender system via deep learning," *Journal of Information and Optimization Sciences*, vol. 43, no. 5, pp. 1081-1088, 2022.

[11] J. Xia, "Construction and implementation of music recommendation model utilising deep learning artificial neural network and mobile edge computing," *International Journal of Grid and Utility Computing*, vol. 13, no. 2-3, pp. 183-194, 2022.

[12] S. Dai, X. Ma, Y. Wang, and R. B. Dannenberg, "Personalised popular music generation using imitation and structure," *Journal of New Music Research*, vol. 51, no. 1, pp. 69-85, 2022.

[13] A. Abdul, J. Chen, H.-Y. Liao, and S.-H. Chang, "An emotion-aware personalized music recommendation system using a convolutional neural networks approach," *Applied Sciences*, vol. 8, no. 7, 1103, 2018.

[14] Y. M. Costa, L. S. Oliveira, and C. N. Silla Jr, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 28-38, 2017.

[15] H. Bai, "Convolutional neural network and recommendation algorithm for the new model of college music education," *Entertainment Computing*, vol. 48, 100612, 2024.

[16] Y. Zhang, "Study on improved personalised music recommendation method based on label information and recurrent neural network," *International Journal of Information and Communication Technology*, vol. 24, no. 1, pp. 48-59, 2024.

[17] W. Shafqat, and Y.-C. Byun, "A context-aware location recommendation system for tourists using hierarchical LSTM model," *Sustainability*, vol. 12, no. 10, 4107, 2020.

[18] D. Wang, X. Zhang, Y. Wan, D. Yu, G. Xu, and S. Deng, "Modeling sequential listening behaviors with attentive temporal point process for next and next new music recommendation," *IEEE Transactions on Multimedia*, vol. 24, pp. 4170-4182, 2021.

[19] S. Wang, C. Xu, A. S. Ding, and Z. Tang, "A novel emotion-aware hybrid music recommendation method using deep neural network," *Electronics*, vol. 10, no. 15, 1769, 2021.

[20] D. Y. Li, and M. X. Gan, "Music Recommendation Method Based on Multi-Source Information Fusion," *Data Analysis and Knowledge Discovery*, vol. 5, no. 2, pp. 94-105, 2021.

[21] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.

[22] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68-75, 2017.

[23] D. Soydaner, "Attention mechanism in neural networks: where it comes and where it goes," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13371-13385, 2022.

[24] K. L. Elmore, and M. B. Richman, "Euclidean distance as a similarity metric for principal component analysis," *Monthly Weather Review*, vol. 129, no. 3, pp. 540-549, 2001.

[25] H. K. Kathania, S. Shahnawazuddin, W. Ahmad, and N. Adiga, "Role of linear, mel and inverse-mel filterbanks in automatic recognition of speech from high-pitched speakers," *Circuits, Systems, and Signal Processing*, vol. 38, pp. 4667-4682, 2019.

[26] S. Zhang, X. Xu, Y. Pang, and J. Han, "Multi-layer attention based CNN for target-dependent sentiment classification," *Neural Processing Letters*, vol. 51, pp. 2089-2103, 2020.

[27] J. Lee, and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208-1212, 2017.

[28] W. Yan, D. Wang, M. Cao, and J. Liu, "Deep auto encoder model with convolutional text networks for video recommendation," *IEEE Access*, vol. 7, pp. 40333-40346, 2019.