Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **3**, August **2024**

Intelligent Recognition of Musical Score Difficulty Based on Improved Deep Belief Networks

Meng Lin*

Graduate School of Christian Studies
Baekseok University, Seoul 06695, South Korea
lemon19910711@163.com

*Corresponding author: Meng Lin

Received November 16, 2023, revised February 17, 2024, accepted April 22, 2024.

Abstract. *Piano sheet music difficulty artificial intelligence recognition can be applied to music software and applications to automatically recommend suitable tunes according to the user's level. Therefore, in order to solve the problem that it is difficult to assign massive piano sheet music difficulty level labels manually, this work proposes an intelligent recognition method of piano sheet music difficulty based on improved Deep Belief Network. First, in order to find more difficulty-related features to expand the feature space, eight new difficulty-related features are proposed by analyzing the difficulty classification guidelines provided by music websites, and the effectiveness of these new features is verified by comparison experiments and the I-RELIEF weighting algorithm. Then, the eight new difficulty-related features are input into multiple Restricted Boltzmann Machine (RBM) layers of Deep Belief Network (DBN) for deep training, so as to extract deep abstract features. Considering that the number of neurons in the hidden layer has a large impact on the effect of feature extraction, and it is difficult to set the number of neurons artificially to bring out the best performance of the network, the number of neurons in each layer of the network is optimized within a specified range using the Artificial Bee Colony (ABC) algorithm to obtain the optimal stacking structure of the network. Finally, an extreme learning machine is used to classify and recognize the difficulty level, which compensates for the time-consuming feature extraction process of the network. Simulation experiments are commonly conducted on the music score dataset. The results show that the proposed method has 100% accuracy in classifying and recognizing the four difficulty levels.*

Keywords: difficulty level recognition; feature selection; deep belief network; artificial bee colony algorithm; learning rate

1. **Introduction.** Artificial intelligence recognition of piano sheet music difficulty can provide learners with personalized sheet music recommendations to help them choose the right piece of music with the right level of difficulty, thus improving the efficiency of practice. It can also be applied to music software and applications to automatically recommend suitable tunes according to the user's level.

With the increase of music learning population, learners need personalized sheet music recommendation. However, traditional piano sheet music difficulty assessment is highly subjective and cannot accurately fit different needs. Therefore, the development of piano sheet music difficulty artificial intelligence recognition technology can more accurately assess the difficulty of the sheet music, personalized recommendation, and improve the learning efficiency [1, 2]. Music teaching requires scientific and reasonable progress planning. However, the traditional empirical method is difficult to accurately grasp the learning ability of students. Intelligent recognition technology can analyze the level of students, assess the difficulty of piano sheet music, recommend appropriate sheet music, and plan a scientific learning route. Other music application scenarios also require intelligent difficulty recognition. For example, intelligent sorting of sheet music for music software, designing levels for games, etc. [3]. Intelligent recognition technology can give products a better user experience and greater commercial value.

Different instruments and different performers may have different perceptions of the difficulty of the same piece of music, making it difficult to objectively assess the difficulty. Melody, harmony, rhythm, and technique are all related, so it is difficult to establish a systematic evaluation system. It is necessary to refine the musical features that can effectively represent the difficulty, and to adjust the parameters for different instruments and styles [4]. However, manually labeling a large number of sheet music samples requires professional knowledge and a lot of manpower, and the scale and quality of labeling are difficult. It is necessary to select appropriate machine learning or deep learning models, and to design the structure and hyper-parameter optimization for the music domain [5]. Therefore, establishing a robust and widely applicable piano score difficulty assessment model requires comprehensive cross-domain research to achieve a breakthrough.

Neural networks have powerful pattern recognition capabilities and can be trained to learn the visual and semantic features of sheet music, modeling the complex mapping relationship of the difficulty of piano sheet music. Pre-trained models such as BERT [6] can extract the semantic information of music scores; Convolutional Neural Networks (CNN) [7] can learn the visual features of music scores, and Recurrent Neural Networks (RNN) [8] can learn the time series features of music scores. Neural networks can fuse heterogeneous music data from multiple sources, such as audio, score images, and symbol representations, to perform multimodal modeling and improve recognition robustness. Therefore, the main research objective of this work is to automatically assign difficulty level labels to a large number of digitized musical scores without difficulty level labels by using the techniques of deep learning theory and classification to achieve fully automatic recognition of difficulty levels of digital piano scores.

1.1. **Related Work.**

more and more studies focus on piano sheet music difficulty level recognition. The main research team comes from the fields of music information retrieval, music education, and computer vision.

In terms of datasets, some standard datasets have been proposed and used, such as the Lakh MIDI dataset and the MAESTRO dataset [9, 10, 11], but they are far from being sufficient. The scale and diversity of data still need to be improved. In terms of model effect, most of the current studies can achieve 70-85% recognition accuracy. However, there is still a certain gap, and it is difficult to meet the practical requirements. Overall, this research field is still in the primary development stage.

From the point of view of recognition methods, the research on piano score difficulty level recognition includes: methods based on traditional machine learning, methods based on deep learning, methods based on multi-modality, methods based on augmentation learning and methods based on transfer learning. The main method studied in this work is the deep learning based method.

Earlier feature engineering+classifier approaches were used, such as by manually designing musical features such as pitch, rhythm, etc., and inputting them into models such as Support Vector Machine (SVM), k nearest neighbors (KNN), etc. Phanichraksaphong and Tsai [12] proposed a method for assessing the difficulty of piano compositions based on an SVM model. Nearly 12-dimensional musical features such as pitch, melodic interval, and rhythm were extracted, and the classification results of 250 piano songs reached about 74% accuracy. This study demonstrated the importance of musical feature design. Sudarma and Harsemadi [13] used the KNN model to achieve tune prediction by counting the interval characteristics of melody, and achieved an accuracy rate of about 60%. This study extracted low-level audio features such as beat and rhythm, and music syntax features such as chord and pitch, resulting in an accuracy rate of about 60%. This type of approach demonstrates that machine learning models can learn music syntax knowledge, but the features and models need to be improved, such as manual feature extraction, large amount of work and poor scalability.

Costa et al. [14] proposed a method to automatically recognize music difficulty using convolutional neural networks (CNN). A large dataset of music scores was used, and the CNN model was trained and tested to successfully achieve automatic recognition of music difficulty. The method can help music educators better assess students' skill levels in order to provide them with more appropriate practice repertoire. Dua et al. [15] proposed a method for music difficulty estimation using recurrent neural networks (RNNs) and long-short-term memory networks (LSTMs) and utilized large-scale music datasets for model training. The results show that the method can accurately estimate the difficulty level of music and achieves good performance in different types of music. Ashraf et al. [16] used Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for feature extraction of musical scores and trained the model to predict the difficulty level. By combining CNN and RNN for feature extraction and temporal modeling of musical scores, automatic recognition of music difficulty is achieved. Experimental results show that the method achieves high accuracy on several music datasets and can be applied to music teaching and music recommendation systems.

Deep Belief Network (DBN) is a neural network model that consists of a multilayer stacked structure of multiple Restricted Boltzmann Machines (RBMs). DBN can be used for unsupervised learning and feature extraction. Gao et al. [17] DBN-based math problem difficulty level prediction method. By constructing a DBN model containing word embedding layer, hidden layer and output layer, the math problem is described as a multi-categorization problem, and the DBN model is used to predict the difficulty level of the problem. The experimental results show that the method has high accuracy and robustness in predicting the difficulty level. Wang et al. [18] used the DBN model to predict the programming difficulty level. The researchers performed the prediction of programming difficulty level by inputting the features of programming problems into the DBN model. The experimental results show that the method can predict the difficulty level of programming problems more accurately.

1.2. **Motivation and contribution.** Compared to CNN and RNN, DBN can effectively process sequence data in musical scores by analyzing the arrangement of notes in the score as input. In contrast, CNNs are more suitable for processing image data, while RNNs are mainly used for processing sequence data [19, 20]. DBN can capture complex musical patterns in a score by stacking multiple hidden layers for nonlinear feature extraction [21, 22]. This fact allows DBN to better portray the relationships and patterns between notes in piano score difficulty assessment.

The deep structure of DBN allows it to handle more complex score data, including longer score sequences and more input features. This makes DBN more scalable and adaptable in the field of piano score difficulty level evaluation. Therefore, in order to solve the problem that it is difficult to assign difficulty level labels for massive piano sheet music manually, this work proposes an intelligent recognition method for piano sheet music difficulty based on improved DBN. The main innovations and contributions of this work include:

(1) In order to find more difficulty-related features to expand the feature space, eight new difficulty-related features are proposed by analyzing the difficulty classification guidelines provided by music websites, and the effectiveness of these new features is verified by comparison experiments and the I-RELIEF weighting algorithm [23, 24].

(2) Eight new difficulty-related features are input into the DBN for deep training, so as to extract deep abstract features. Considering that the number of neurons in the hidden layer has a large impact on the effect of feature extraction, and it is difficult to set the number of neurons artificially to bring out the best performance of the network, the number of neurons in each layer of the network is optimized by using the Artificial bee colony (ABC) algorithm [25] within the specified range to obtain the optimal stacking structure of the network.

(3) The use of Extreme Learning Machine (ELM) [26] for classification and identification of difficulty levels compensates for the time-consuming process of network feature extraction.

2. Establishment of the feature space.

2.1. The role of feature space.

The vectors in the feature space can be directly used as inputs to machine learning and deep learning models for end-to-end modeling and training.

Each dimension in the feature space can represent a musical semantic feature such as pitch, rhythm, and harmony. These features can reflect the difficulty level of the score. Different lengths and forms of musical scores are uniformly mapped to a fixed dimension feature space, which is convenient for model input and processing. The high-dimensional music data can be abstractly represented in the feature space at low latitude, which reduces the computational burden and highlights the main features. The feature space can visualize the high-dimensional music data and understand the data distribution more intuitively. It also facilitates hyper-parameter tuning. Scores with different difficulty levels can be mapped to different regions of the feature space to achieve differentiated recognition.

2.2. Difficulty feature extraction.

Existing difficulty-related features of piano scores mainly involve score information such as beat, pitch, time, score length, note alteration and hand displacement [27].

On the basis of comprehensively analyzing the score information contained in the existing difficulty features, and referring to the difficulty level classification standards provided by music websites, this work found some other score information closely related to difficulty. These specific quantitative features can effectively reflect multiple aspects of the difficulty of piano scores, which can be combined to realize a comprehensive assessment of difficulty. Overall, the difficulty feature space after this paper is 14(6+8) dimensions, i.e., there are 14 numerical features. The eight new features adopted in this work which are related to the assessment of the difficulty level of piano music are as follows.

(1) Pitch Range (PR): Measures the size of the pitch range used in a melody.

$$PR = \max(P) - \min(P) \tag{1}$$

where P is the set of pitches.

(2) Highest Pitch (HP): Reflects the pitch of the highest note of the tune.

$$HR = \max(P) \tag{2}$$

(3) Largest Pitch Interval (LPI): the largest pitch interval in the melody.

$$LPI = \max\{|P_{i+1} - P_i|\} \tag{3}$$

where P_i and P_{i+1} are neighboring pitches.

(4) Polyphonic Techniques (PT): Evaluates the number and complexity of polyphonic techniques used.

$$PT = \frac{N_{poly}}{N_{notes}} \tag{4}$$

where N_{poly} is the number of notes using the polyphony technique and N_{notes} is the total number of notes.

(5) Chord Change Speed (CCS): The frequency of chord change per unit time.

$$CCS = \frac{N_{chord}}{T} \tag{5}$$

where N_{chord} is the total number of chord changes and T is the total time.

(6) Melodic Rhythm Complexity (MRC): Reflects the complexity of rhythmic changes of notes in the melody [28]. This information entropy index reflects the randomness and complexity of rhythmic changes in the melody, and the higher the value, the more complex the melodic rhythm. Let there are n types of notes (e.g., whole notes, quarter notes, etc.), denoted as $R_1, R_2, \ldots, R_n$, and the number of times each type of note appears in the melody is $N(R_i)$, then the probability of occurrence of each type of note is:

$$P(R_1) = \frac{N(R_1)}{N_{notes}}, \quad P(R_2) = \frac{N(R_2)}{N_{notes}}, \ldots, \quad P(R_n) = \frac{N(R_n)}{N_{notes}} \tag{6}$$

Thus, MRC is defined as the information entropy of the probability of occurrence of each type of note:

$$MRC = -\sum P(R_i) \log P(R_i) \tag{7}$$

where $\sum$ is a summation operation for all note types R_i.

(7) Number of Notes Simultaneously (NNS): Reflects the characteristics of vertical complexity in the tune.

$$NNS = \max(N_{simult}) \tag{8}$$

where N_{simult} is the number of simultaneous notes occurring at any time.

(8) Proportion of Repetition Structures (PRS): Evaluates the proportion of repetition structures in the structure of the piece.

$$PRS = \frac{T_{repetition}}{T_{total}} \tag{9}$$

where $T_{repetition}$ is the repetition structure time and T_{total} is the total time.

2.3. **Data preprocessing.** Prior to difficulty recognition, it is necessary to quantify the collected scores, normalize the extracted feature data as well as pre-processing operations to resolve data imbalances.

First of all, due to the time resolution of the music file is generally higher, resulting in some notes do not appear in the correct rhythmic position, so before extracting the features to quantize the music file, so that the note onset time (onset time) and duration (duration) can appear in the correct rhythmic position, the specific quantization process is shown in Figure 1.

After feature extraction, this paper uses the Min-Max normalization algorithm to normalize the feature vector to the interval [0, 1].

$$x_i^* = \frac{x_i - \min}{\max - \min} \tag{10}$$

where min and max denote the minimum and maximum values of the feature x_i respectively, and x_i^* denotes the feature x_i after normalization.

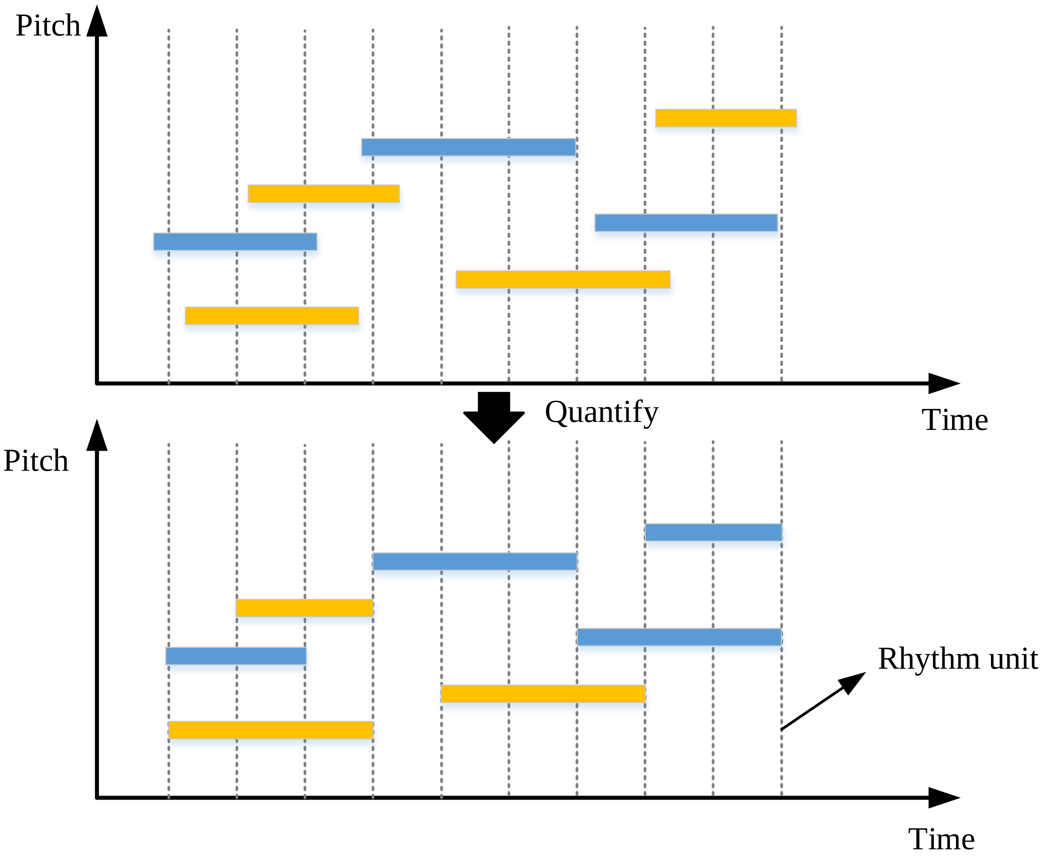

Figure 1. Basic flow of filtered feature selection methods

In addition, in order to solve the data imbalance problem, this paper adopts the Threshold Moving (ThM) algorithm. The prediction threshold of the classifier is adjusted so that the majority class samples are categorized into a minority class, thus balancing the number of samples in each class.

2.4. **Assessing the validity of difficulty-related features.** To verify the validity of the proposed feature space, it was compared with the existing feature space.

For the same MAESTRO dataset, three regression algorithms [29, 30] (Linear Regression, Decision Tree Regression, Random Forest Regression) were used to fit the data in the two different feature spaces and the fit was measured by the Root Mean Squared Error (RMSE). The RMSE was computed as shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

where $\hat{y}_i$ is the predicted difficulty level of the i-th score, y_i is the actual difficulty level of the i-th score, and n is the number of score samples.

Three regression algorithms were used to fit the data in the two feature spaces and the RMSE results are shown in Table 1. It can be seen that all three regression algorithms have better RMSE values under the expanded feature space, indicating that the regression algorithms can fit the data better in the expanded feature space. In addition, the newly proposed difficulty-related features can improve the discrimination between categories. However, despite the better fit in the expanded feature space, the best RMSE value is still not very good, indicating that the fit needs to be further improved or the problem is not suitable to be solved by the regression algorithms.

Table 1. RMSE for three regression algorithms.

Eigenspace	**linear regression**	**decision tree regression**	**Random Forest Regression**
12-dimensional feature space	2.68	2.19	1.96
The proposed 14-dimensional feature space	2.51	1.87	1.62

To further illustrate that the newly proposed features are helpful in achieving the recognition of the difficulty level of piano scores, the I-RELIEF weighting algorithm is further utilized to identify the Lakh MIDI dataset and the MAESTRO dataset to obtain the magnitude of the weights of the individual features and thus measure the effectiveness of the new features. Table 2 lists the eight features with the highest weights assigned by the I-RELIEF weighting algorithm and their corresponding weights.

Table 2. The eight most heavily weighted features and their corresponding weights.

No.	**Features**	**Lakh MIDI**	**MAESTRO**
1	PR	0.0232	0.0223
2	HP	0.0218	0.0209
3	LPI	0.0184	0.0185
4	PT	0.0175	0.0174
5	CCS	0.0168	0.0159
6	MRC	0.0119	0.0115
7	NNS	0.0081	0.0052
8	PRS	0.0065	0.0087

For both datasets, six of the eight features with the largest weights are the same, with PR having the largest weight in all of them. This also shows that the feature space proposed in this paper is effective and has a strong ability to discriminate the difficulty level.

3. **Intelligent recognition of piano score difficulty based on ABC-DBN-ELM.**

3.1. **DBN-based deep abstract feature extraction.** . After verifying the validity of the proposed new feature space, eight new difficulty-related features are input into multiple RBM layers of DBN for deep training to extract deep abstract features.

The DBN hierarchical structure mainly consists of a visible layer $\mathbf{v} = (v_1, v_2, \cdots, v_m)$ and a hidden layer $\mathbf{h} = (h_1, h_2, \cdots, h_n)$, where the h-layer contains multiple levels of RBM. The structure of RBM is shown in Figure 2.

Let $\theta = \{w, c, b\}$, where w is the weight between $\mathbf{v}$ and $\mathbf{h}$, c and b are the bias between $\mathbf{v}$ and $\mathbf{h}$, then the effect of a node in the $\mathbf{h}$ layer on the $\mathbf{v}$ layer is shown below:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{12}$$

where $E(\mathbf{v}, \mathbf{h})$ obeys the Bernoulli distribution.

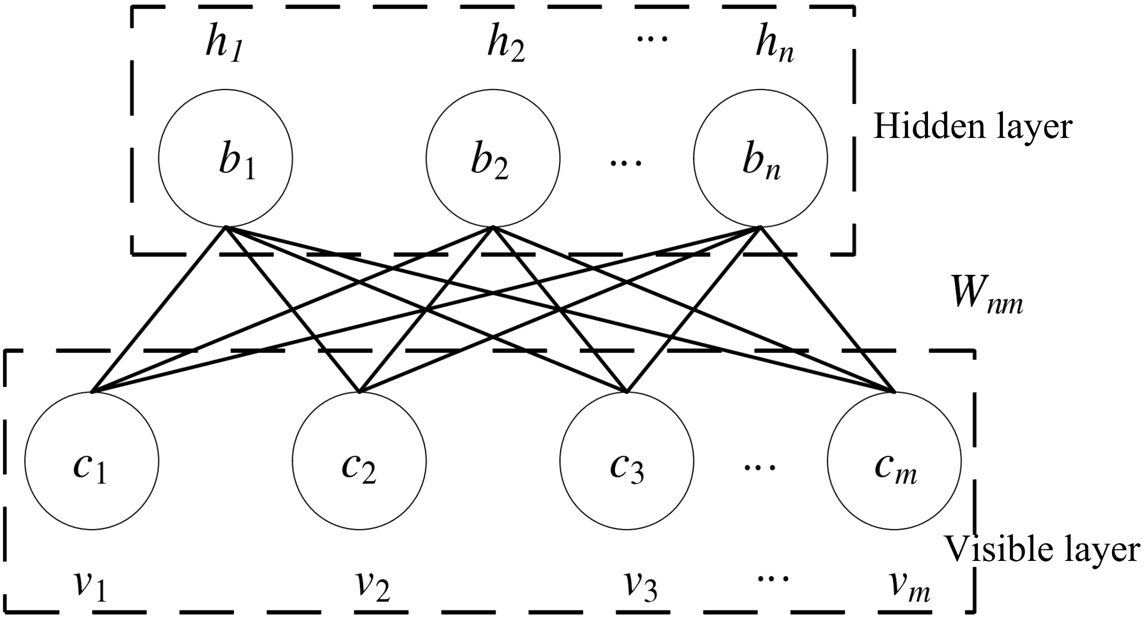

Figure 2. Structure of the RBN

Let the three coefficients of the Bernoulli distribution of the i-th hidden layer and the j-th visible layer be c_j, b_j and w_j.

$$E(\mathbf{v},\mathbf{h}) = -\sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} v_j h_i \tag{13}$$

Then the role of all nodes in layer $\mathbf{h}$ on layer $\mathbf{v}$ is shown below:

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{14}$$

In turn, all the nodes in the $\mathbf{v}$-layer act on the $\mathbf{h}$-layer as follows:

$$P(\mathbf{h}) = \sum_{\mathbf{v}} P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\sum_{\mathbf{v}} e^{-E(\mathbf{v},\mathbf{h})} \tag{15}$$

The effect of all $\mathbf{v}$-layer nodes on the i-th $\mathbf{h}$-layer node is shown below:

$$P(h_i = 1|\mathbf{v}) = \sigma(c_i + \sum_{j=1}^{m} w_{ij} v_j) \tag{16}$$

The effect of all $\mathbf{h}$-tier nodes on the i-th $\mathbf{v}$-tier node is shown below:

$$P(v_j = 1|\mathbf{h}) = \sigma(b_j + \sum_{j=1}^{n} w_{ji} v_i) \tag{17}$$

where σ is $\sigma(x) = 1/(1+e^{-x})$.

If $v_0, v_1, \ldots, v_m$ in $\mathbf{v}$ obeys an independent homogeneous distribution, then the maximum likelihood estimate is obtained by taking the natural logarithm of Equation (14).

$$\overset{\wedge}{\theta} = \arg\max_{\theta} \sum_{t=0}^{m} \ln P(v_t|\theta) \tag{18}$$

$$\theta^{*} = \theta + \eta\frac{\partial \ln P(\mathbf{v})}{\partial\theta} \tag{19}$$

where $\eta\,(\eta > 0)$ is the learning rate.

The effect of the l-th visual layer on the h-layer is shown below:

$$\ln P(v_0) = \ln\sum_{h} e^{-E(v_0,h)} - \ln\sum_{v,h} e^{-E(v,h)} \tag{20}$$

In order to obtain the key parameters of DBN, the derivations are performed for w_{ij}, b_j, and c_j respectively.

$$\left\{\begin{array}{c} \frac{\partial \ln P(\mathbf{v_0})}{\partial w_{ij}} = P(h_i = 1|\mathbf{v_0})v_{0j} - \sum\limits_{\mathbf{v}} P(v)P(h_i = 1|\mathbf{v}) \\ \frac{\partial \ln P(\mathbf{v_0})}{\partial b_j} = v_{0j} - \sum\limits_{v} P(\mathbf{v}) \\ \frac{\partial \ln P(\mathbf{v_0})}{\partial c_i} = P(h_i = 1|\mathbf{v_0}) - \sum\limits_{\mathbf{v}} P(\mathbf{v})P(h_i = 1|\mathbf{v}) \end{array}\right. \tag{21}$$

The relationship between the results of the current iteration and the last iteration is shown below:

$$\left\{\begin{array}{c} w_{ij}{}^{*} = w_{ij} + \eta\frac{\partial \ln P(\mathbf{v_0})}{\partial w_{ij}} \\ b_j{}^{*} = b_j + \eta\frac{\partial \ln P(\mathbf{v_0})}{\partial b_j} \\ c_i{}^{*} = c_i + \eta\frac{\partial \ln P(\mathbf{v_0})}{\partial c_i} \end{array}\right. \tag{22}$$

Finally the v-layer weight parameters of the DBN are solved inversely to determine the DBN network structure.

3.2. **ABC-DBN.** Considering that the number of neurons in the hidden layer has a large impact on the effect of feature extraction, and it is difficult to set the number of neurons artificially to bring out the best performance of the network, the ABC algorithm is used to optimize the number of neurons in each layer of the network in a specified range in order to obtain the optimal stacking structure of the network.

The ABC algorithm mainly takes the honey source as the optimal solution of the algorithm, and leads the detector bees and the follower bees to complete the search of the honey source together. Let the nectar source be i and the d-dimensional initial random position of the detector bee be $\boldsymbol{X}_{id}$.

$$\boldsymbol{X}_{id} = L_d + \mathrm{rand}(0,1)(U_d - L_d) \tag{23}$$

where U_d and L_d are the maximum and minimum values of the boundary of the nectar source in the d-dimensional property, respectively.

The detector bees start the nectar search at $\boldsymbol{X}_{id}$ and set the new nectar source as $\boldsymbol{V}_{id}$.

$$\boldsymbol{V}_{id} = \boldsymbol{X}_{id} + \varphi(\boldsymbol{X}_{id} - \boldsymbol{X}_{jd}) \tag{24}$$

where φ is a random value in the range [1,1] and $\boldsymbol{X}_{jd}$ is any position in the d-th dimension of the search range except $\boldsymbol{X}_{id}$.

When a new nectar source is detected by a detector bee, the old and new nectar sources are compared in terms of fitness. The adaptation f_i of the new nectar source $\boldsymbol{V}_i = [\boldsymbol{V}_{i1}, \boldsymbol{V}_{i2}, \cdots, \boldsymbol{V}_{id}]$ is calculated as shown below:

$$fit_i = \left\{\begin{array}{c} 1/(1+f_i), f_i \geq 0 \\ 1 + abs(f_i), otherwise \end{array}\right. \tag{25}$$

If the fitness value of $\boldsymbol{V}_i$ is better than $\boldsymbol{X}_i$, the original nectar source is replaced with the new nectar source, otherwise it is not replaced. The detecting bee communicates the nectar source data to the following bee, and when the following bee finds more than one nectar source within its search range, it selects the nectar source according to the probability p_i.

$$p_i = \frac{fit_i}{\sum_{i=1}^{S} fit_i} \tag{26}$$

where S denotes the total number of nectar sources.

During nectar search, when the number of iterations G is less than or equal to the maximum number of iterations G_{max}, then jump to Equation (23) and re-execute, otherwise the algorithm stops.

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{L}_d + rand(0,1)(\mathbf{U}_d - \mathbf{L}_d), G \geq G_{\max} \\ \mathbf{X}_i^t, G < G_{\max} \end{cases} \tag{27}$$

In this paper, the ABC algorithm is used to optimally solve the DBN core parameters, i.e., updating the RBM key parameters in accordance with Equation (24) to Equation (26), and obtaining the optimally adjusted RBM network structure parameters $\theta^* = (w_{ij}^*, b_j^*, c_i^*)$. Finally, the stable DBN structure model is obtained.

3.3. **ELM-based difficulty level classification.** Compared with traditional neural network classification methods, ELM has faster learning speed, good generalization performance, and is less likely to fall into local minima, which makes it more suitable for difficulty level classification [31]. The neural network structure of ELM is shown in Figure 3 The deep abstract features extracted by ABC-DBN are fed into the ELM for final difficulty level classification. For an ELM network with a hidden layer containing L neurons, the output of the network can be expressed as:

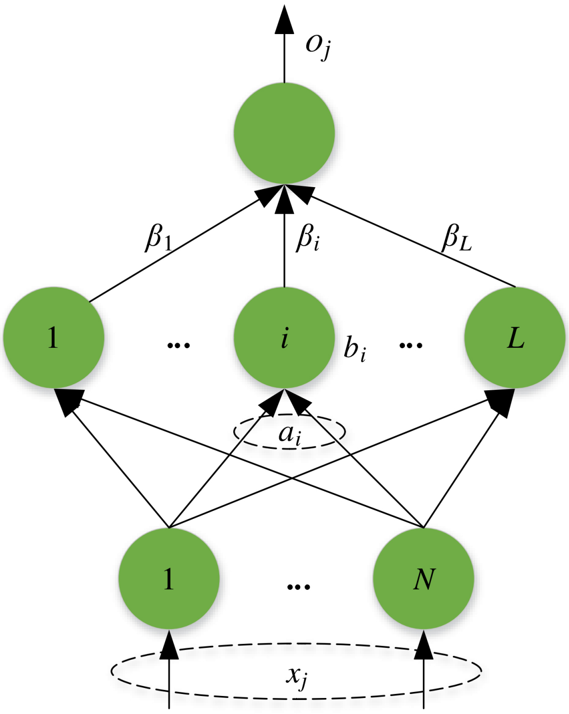

Figure 3. Neural Network Architecture for ELMs

$$\sum_{i=1}^{L} \beta_i g\left(W_i \cdot X_j + b_i\right) = O_j, j = 1, 2, ..., N \tag{28}$$

where $g(x)$ is the activation function, $\boldsymbol{W}$ is the weight matrix of the input, $\boldsymbol{\beta}$ is the weight of the output, and b_i is the bias of the i-th hidden layer neuron.

The learning objective of the ELM network is to minimize the error of the output [32].

$$\sum_{i=1}^{N} \|o_j - t_j\| = 0 \tag{29}$$

The process of training the ELM is to find $\boldsymbol{\beta}$, $\boldsymbol{W}$ and b_i to make the output as close as possible to the classification label of the sample. Therefore, the structure of ABC-DBN-ELM is shown in Figure 4.

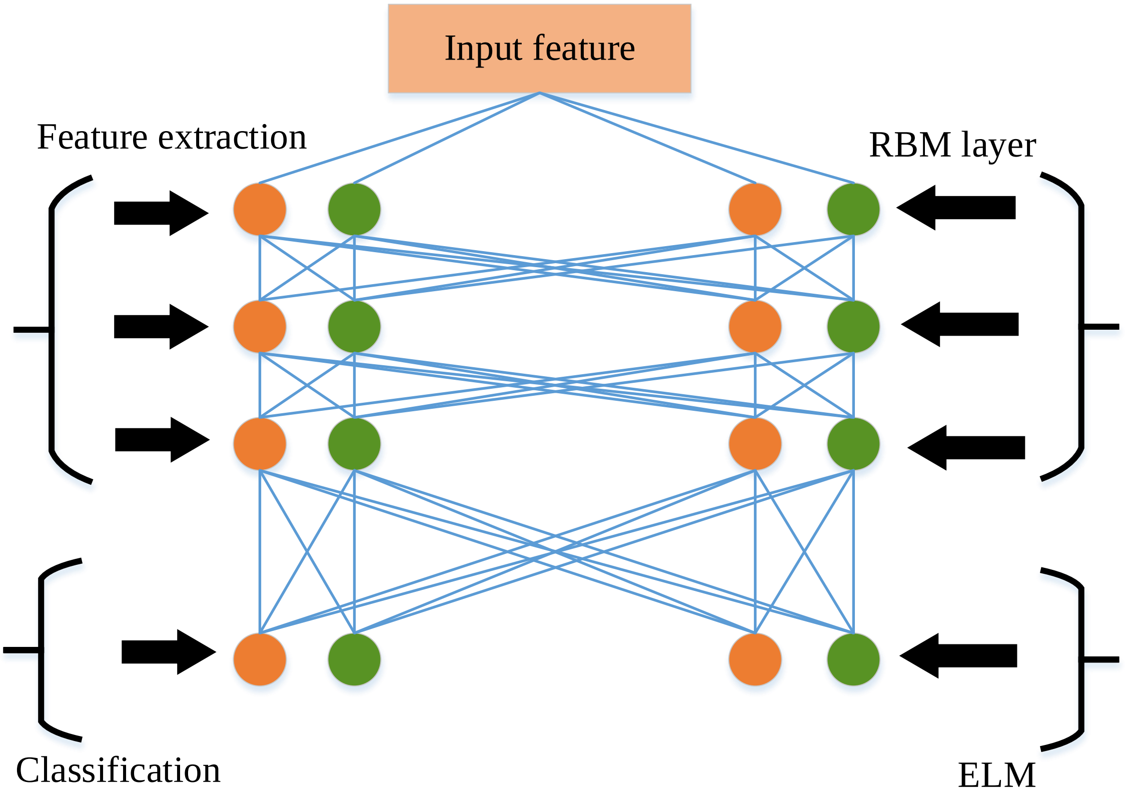

Figure 4. Structure of ABC-DBN-ELM

4. Experimental results and analysis.

4.1. Data sources and data preprocessing. In order to verify the performance of ABC-DBN-ELM in the field of intelligent recognition of piano sheet music difficulty, Matlab is used for example simulation. The computer environment used is a 32-bit Windows 10 system with a built-in Intel I5-4200M processor and 8 G of RAM.

The experimental data is the MAESTRO dataset, a large music dataset used for music information retrieval and related tasks, containing tens of thousands of classical music works covering multiple musical periods and styles. A random sample of 500 MIDI-formatted score files containing four difficulty levels were selected from the MAESTRO dataset as the source of experimental data. After data preprocessing, piano sheet music difficulty level classification was recognized using ABC-DBN-ELM. Each experiment was repeated 5 times, each time with 5-fold cross-validation, and the average value was taken as the final result.

4.2. **Selection of optimal learning rate.** In order to verify the effect of the DBN core parameter learning rate η on the difficulty level classification of piano scores, the classification accuracy at different η values is verified, as shown in Table 3.

Table 3. Classification Accuracy for Different η Values

Difficulty level category	learning rate η	Classification accuracy		
		Minimum	Average	Maximum
1	0.02	0.8912	0.8914	0.8935
	0.06	0.8998	0.9021	0.9047
	0.08	0.9246	0.9255	0.9284
	0.10	0.9131	0.9151	0.9172
2	0.02	0.9003	0.9042	0.9067
	0.06	0.9111	0.9142	0.9175
	0.08	0.9417	0.9442	0.9464
	0.10	0.9306	0.9338	0.9361
3	0.02	0.8992	0.9017	0.9045
	0.06	0.9092	0.9145	0.9166
	0.08	0.9392	0.9408	0.9443
	0.10	0.9391	0.9403	0.9432
4	0.02	0.8978	0.9004	0.9028
	0.06	0.8992	0.9035	0.9064
	0.08	0.9322	0.9346	0.9368
	0.10	0.9232	0.9247	0.9261

It can be seen that at different η values, although ABC-DBN-ELM obtains a classification accuracy of more than 0.9 in all four difficulty levels, there are still some differences. This is mainly because under the condition of a certain maximum number of iterations, the η value determines the rate and range of the search for the optimal value, which affects the accuracy of ABC-DBN detection. Cross-sectional comparisons revealed that both ABC-DBN-ELM achieved the highest classification accuracy at $\eta = 0.08$. Therefore, the learning rate of $\eta = 0.08$ was chosen as the optimal parameter.

4.3. **DBN feature extraction results.** After the feature extraction by three layers of RBM and the inverse fine-tuning of the ABC algorithm, the 14-dimensional features obtained after DBN feature extraction of the original data of the sheet music file. The feature extraction results are shown in Table 4.

Table 4. DBN Network Feature Extraction Results

ID	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	...	Feature 14
1	0.602929	0.570384	0.61914	0.554068	0.526342	0.530688	0.589821	...	0.565637
2	0.594544	0.556673	0.603014	0.538962	0.524628	0.519579	0.583409	...	0.554929
3	0.5534	0.509959	0.537609	0.491562	0.514099	0.484331	0.54839	...	0.514437
4	0.601131	0.565632	0.61453	0.548556	0.52607	0.526847	0.588407	...	0.562459
5	0.587817	0.540709	0.58423	0.51795	0.523726	0.50403	0.577866	...	0.54223
6	0.586026	0.537923	0.580529	0.514465	0.523365	0.501547	0.576291	...	0.53989
7	0.595655	0.55453	0.601728	0.535074	0.525144	0.516961	0.584093	...	0.553897
8	0.58309	0.529033	0.570751	0.503417	0.523032	0.492806	0.574328	...	0.33364
9	0.602111	0.571302	0.618996	0.555321	0.526038	0.531526	0.589072	...	0.565876

In order to further demonstrate the feature extraction effect of the DBN network, the results extracted in the previous step need to be visualized. Using LLE to reduce the features to a two-dimensional space and normalize the reduced data, the visualized results are shown in Figure 5. F0-F3 represent four piano music score difficulty level states, respectively. It can be seen that although several states in the original data are also roughly distributed, there are many overlapping points between different states, which is not convenient for direct classification. As can be seen in Figure 5, the four state features automatically extracted by DBN are all clearly distinguished, proving DBN's very good feature extraction ability.

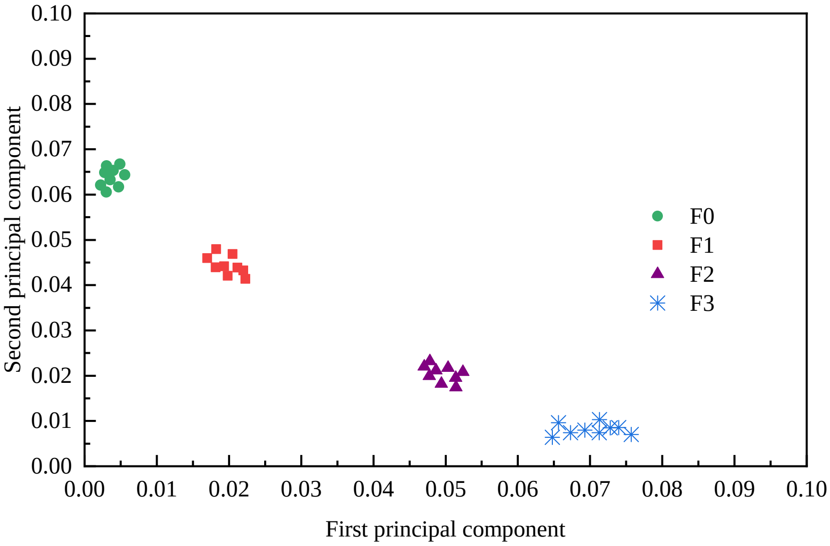

Figure 5. 2D Feature Visualization

4.4. **Selection of the number of ELM neurons L.** Since L has a large impact on both the training time and classification accuracy of ELM, in order to determine the optimal L, the relationship between L and the accuracy of the test set is analyzed in this paper, as shown in Figure 6.

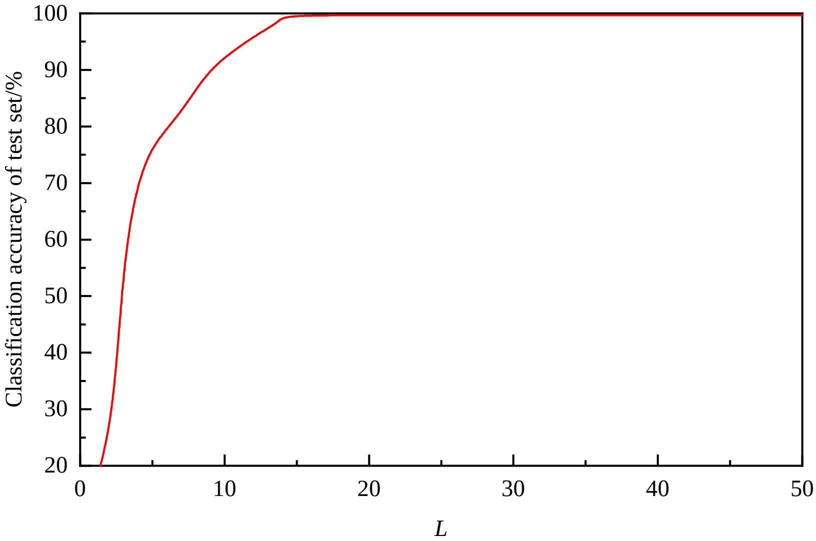

Figure 6. Effect of L on test set accuracy

It can be seen that when $L = 13$, the accuracy of the test set reaches 100% for the first time. As L continues to increase, the accuracy basically stabilizes between 95% and

100%, but considering the effect of L on the speed of the ELM, the number of neurons in the hidden layer of the ELM is finally chosen to be 13.

4.5. **Effect of ABC on classification performance.** To further verify the improvement effect of ABC on DBN-ELM, the classification accuracy RMSE and classification time were compared, as shown in Table 5.

Table 5. Performance comparison between DBN-ELM and ABC-DBN-ELM

Algorithm	**Category**	**RMSE mean**	**Detection time/s**
DBN-ELM	1	6.226×10^{-2}	23.664
	2	7.283×10^{-2}	23.739
	3	5.417×10^{-2}	25.25
	4	5.325×10^{-2}	24.916
ABC-DBN-ELM	1	4.734×10^{-2}	23.678
	2	5.624×10^{-2}	23.833
	3	4.987×10^{-2}	25.175
	4	5.012×10^{-2}	24.991

It can be seen that the classification RMSE value of DBN-ELM algorithm is significantly higher than that of ABC-DBN-ELM algorithm, which indicates that the stability of its classification accuracy is significantly improved after the optimization of DBN using ABC algorithm, which also reflects that the optimization of the DBN weight parameter solution using ABC algorithm is able to obtain a more stable value of the weight parameter.

In terms of classification time, the difference between the classification time of the DBN algorithm and the ABC-DBN algorithm at steady state is small, which is due to the fact that after adding the ABC optimization, the DBN saves the time of solving the RBM parameter $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{c}, \boldsymbol{b}\}$ one by one inversely. The difference in classification time between the two is not large because the DBN algorithm is able to obtain a better parameter $\boldsymbol{\theta} = \{\boldsymbol{w}, \boldsymbol{c}, \boldsymbol{b}\}$ after the introduction of ABC optimization. Under the same classification accuracy threshold, the ABC algorithm's own search optimization consumes about the same amount of time as the DBN's solution time, but the accuracy is clearly superior to ABC-DBN.

4.6. **Difficulty level classification results.** The classification accuracies of the training and test sets are simulated and analyzed respectively, and the classification results of the test set are shown in Figure 7. Where the horizontal coordinate represents the number of samples and the vertical coordinate represents the four difficulty level labels. 80% (400) of the original samples are taken for each of the training set and 20% (100) for the test set.

The classification and recognition results of evaluating ABC-DBN-ELM are judged based on whether the true labels of the data and the labels diagnosed by the model overlap. The confusion matrix is used to visualize the correct classification rate of each class, as shown in Figure 8.

It can be seen that the diagnostic accuracy of the four difficulty levels of classification and recognition is 100%, and the classification results are consistent with the feature visualization results extracted by DBN.

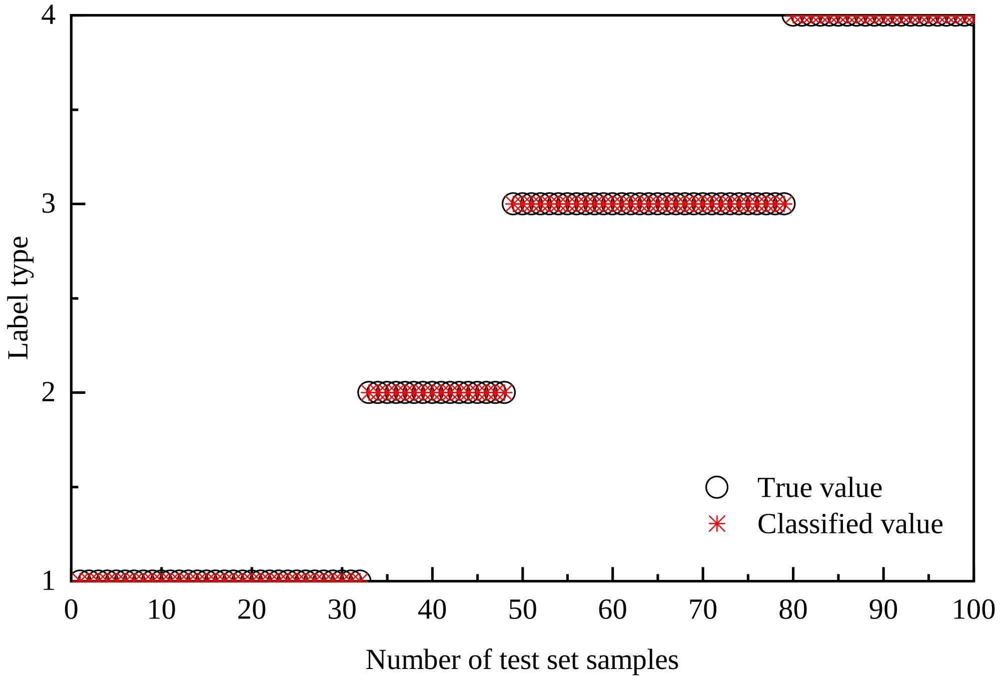

Figure 7. Test set classification results

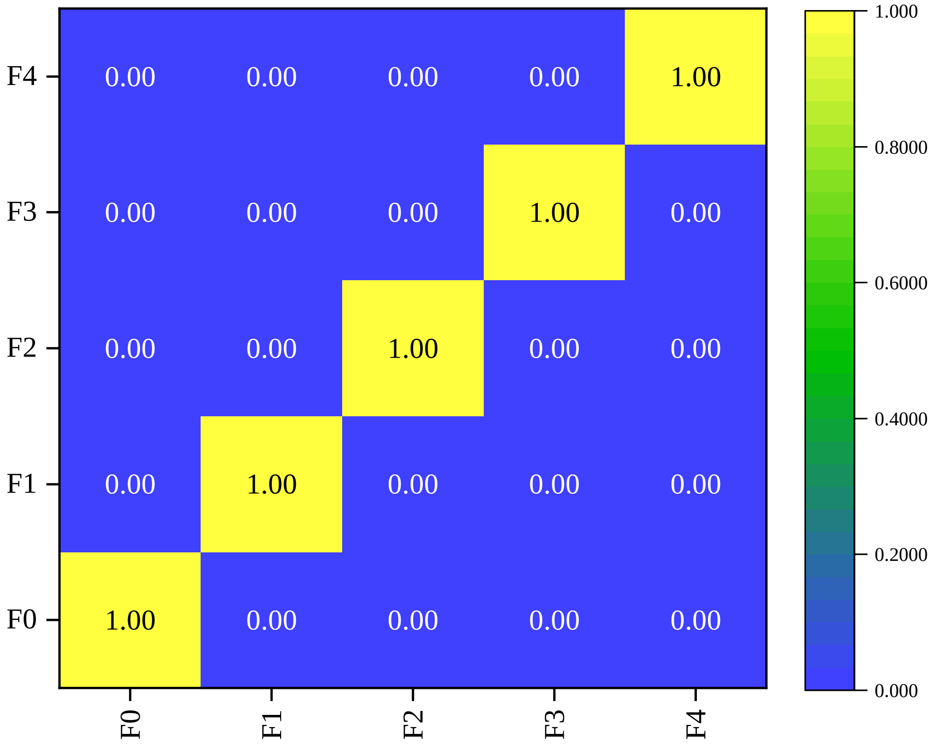

Figure 8. Heat map of test set classification results

5. **Conclusion.** In order to find more difficulty-related features to extend the feature space, this work proposes eight new difficulty-related features by analyzing the difficulty classification guidelines provided by music websites. Then, the eight new difficulty-related features are input into DBN for deep training so as to extract deep abstract features. Next, the ABC algorithm is taken to optimize the number of neurons per layer of the

network within a specified range to obtain the best stacking structure of the network. Finally, ELM is used to classify and recognize the difficulty levels, which compensates for the time-consuming feature extraction process of the network. The results show that the diagnostic accuracy of the classification recognition for all four difficulty levels is 100%, and the classification results are consistent with the feature visualization results extracted by DBN. However, the training process of DBN involves two stages: layer-by-layer pre-training and fine-tuning. These two stages of the training process are relatively cumbersome, and for large-scale datasets, the training time can be very long. Follow-up studies will attempt to improve the training speed by using distributed computing frameworks such as Dask or Spark.

REFERENCES

[1] K. Karma, "Musical aptitude definition and measure validation: ecological validity can endanger the construct validity of musical aptitude tests," *A Journal of Research in Music Cognition*, vol. 19, no. 2, 79, 2007.

[2] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[3] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[4] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, and T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[5] K. Wang, Z. Chen, X. Dang, X. Fan, X. Han, C.-M. Chen, W. Ding, S.-M. Yiu, and J. Weng, "Uncovering Hidden Vulnerabilities in Convolutional Neural Networks through Graph-based Adversarial Robustness Evaluation," *Pattern Recognition*, vol. 143, pp. 109745, 2023.

[6] D. Cao, K. Zeng, J. Wang, P. K. Sharma, X. Ma, Y. Liu, and S. Zhou, "BERT-Based Deep Spatial-Temporal Network for Taxi Demand Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9442-9454, 2022.

[7] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24-49, 2021.

[8] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 132306, 2020.

[9] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-Resolution Piano Transcription with Pedals by Regressing Onset and Offset Times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707-3717, 2021.

[10] S. Mukherjee, and M. Mulimani, "ComposeInStyle: Music composition with and without Style Transfer," *Expert Systems with Applications*, vol. 191, pp. 116195, 2022.

[11] C. Arriaga Sanz, and J.-M. Madariaga Orbea, "Is the perception of music related to musical motivation in school?," *Music Education Research*, vol. 16, no. 4, pp. 375-386, 2013.

[12] V. Phanichraksaphong, and W.-H. Tsai, "Automatic Evaluation of Piano Performances for STEAM Education," *Applied Sciences*, vol. 11, no. 24, 11783, 2021.

[13] M. Sudarma, and I. G. Harsemadi, "Design and Analysis System of KNN and ID3 Algorithm for Music Classification based on Mood Feature Extraction," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 1, 486, 2017.

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

[15] M. Dua, R. Yadav, D. Mamgai, and S. Brodiya, "An Improved RNN-LSTM based Novel Approach for Sheet Music Generation," *Procedia Computer Science*, vol. 171, pp. 465-474, 2020.

[16] M. Ashraf, F. Abid, I. U. Din, J. Rasheed, M. Yesiltepe, S. F. Yeo, and M. T. Ersoy, "A Hybrid CNN and RNN Variant Model for Music Classification," *Applied Sciences*, vol. 13, no. 3, 1476, 2023.

[17] H. Gao, Y. Duo, T. Sun, and X. Yang, "Dynamic Safety Management on the Key Equipment of Coal Gasification Based on Dbt-Dbn Method," *Mathematical Problems in Engineering*, vol. 2020, pp. 1-14, 2020.

[18] S. Wang, T. Liu, J. Nam, and L. Tan, "Deep Semantic Feature Learning for Software Defect Prediction," *IEEE Transactions on Software Engineering*, vol. 46, no. 12, pp. 1267-1293, 2020.

[19] Y. Yang, "Medical Multimedia Big Data Analysis Modeling Based on DBN Algorithm," *IEEE Access*, vol. 8, pp. 16350-16361, 2020.

[20] Y. Al-Hadhrami, and F. K. Hussain, "Real time dataset generation framework for intrusion detection systems in IoT," *Future Generation Computer Systems*, vol. 108, pp. 414-423, 2020.

[21] C.-H. Hu, H. Pei, X.-S. Si, D.-B. Du, Z.-N. Pang, and X. Wang, "A Prognostic Model Based on DBN and Diffusion Process for Degrading Bearing," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 10, pp. 8767-8777, 2020.

[22] L. Zhu, L. Chen, D. Zhao, J. Zhou, and W. Zhang, "Emotion Recognition from Chinese Speech for Smart Affective Services Using a Combination of SVM and DBN," *Sensors*, vol. 17, no. 7, pp. 1694, 2017.

[23] B. Tang, and L. Zhang, "Local preserving logistic I-Relief for semi-supervised feature selection," *Neurocomputing*, vol. 399, pp. 48-64, 2020.

[24] F. C. Grant, "Experiences with intramedullary tractotomy," *Archives of Surgery*, vol. 42, no. 4, 681, 1941.

[25] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21-57, 2012.

[26] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 41611-41660, 2021.

[27] X. Yang, Y. Dong, and J. Li, "Review of data features-based music emotion recognition methods," *Multimedia Systems*, vol. 24, no. 4, pp. 365-389, 2017.

[28] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, "Combining visual and acoustic features for music genre classification," *Expert Systems with Applications*, vol. 45, pp. 108-117, 2016.

[29] D. Maulud, and A. M. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140-147, 2020.

[30] M. Abdurohman, A. G. Putrada, and M. M. Deris, "A Robust Internet of Things-Based Aquarium Control System Using Decision Tree Regression Algorithm," *IEEE Access*, vol. 10, pp. 56937-56951, 2022.

[31] S. Lu, S.-H. Wang, and Y.-D. Zhang, "Detection of abnormal brain in MRI via improved AlexNet and ELM optimized by chaotic bat algorithm," *Neural Computing and Applications*, vol. 33, no. 17, pp. 10799-10811, 2020.

[32] S. Mugunthan, and T. Vijayakumar, "Design of improved version of sigmoidal function with biases for classification task in ELM domain," *Journal of Soft Computing Paradigm*, vol. 3, no. 2, pp. 70-82, 2021.