Asia-Pacific Journal for Arts Education

Special Issue

Unfold the Future of Music Education through Technology

http://www.ied.edu.hk/ccaproject/apjae/apjae.htm ISSN: 1683-6995

Volume 22 Number 1 December 2023

Recognition of Piano Score Difficulty Level and Application in Music Teaching

Wei Gan

Central China Normal University

musicgw@ccnu.edu.cn

Bo He

Central China Normal University

hb925822@126.com

Abstract

Scientifically and accurately grading the difficulty of music scores is an essential foundation to realizing personalized music education. At present, the classification of music score difficulty levels in piano teaching is mainly done by manual labeling, which faces the double challenges of inefficiency and tendentiousness, whereas the performance of the previous difficulty level recognition algorithm is not enough to be applied to teaching practice. In this paper, we first introduce a piano score difficulty level recognition model based on an LSTM Neural Network. We then propose a data extension method through pitch offset, which can effectively resolve the over-fitting issue that appeared in training on a small dataset. Experimental results clearly demonstrate that our model represents a significant improvement compared with existing methods (78%), as well as achieving the highest accuracy rate on the three difficulty dataset levels (88%). Finally, in a piano tutoring platform independently developed by us, we utilize this model to recommend suitable scores for adaptive practice and personalized assessment according to teaching targets and learners' mastery, both of which effectively improve learners' achievements.

Key words

difficulty recognition, classification, piano instruction, music score recommendation

Introduction

The piano is the instrument with the most learners in the world. The purpose of piano learning is to improve an individual's playing ability, which is reflected in fingering, speed and other skills indicated within the structure of the music. In the graded piano tests, assessment is based on the degree to which the examinee's performance satisfies the requirements based on the difficulty level. Music scores are likewise divided into levels of difficulty, the appropriate materials being utilized according to the abilities of the individual concerned. The aim of this near-universal teaching mode is to enable learners to gradually master various piano playing skills from easy to difficult in a hierarchical and personalized way. Stratified teaching is an effective strategy for teachers to solve the enormous differentiation of students' ability levels in practice. In order to achieve stratified teaching, however, it is not only necessary to divide learning resources according to students' basic levels, but also to specify the corresponding evaluation methods for people with different foundations, so as to "teach students according to their aptitude" and stimulate their enthusiasm for piano learning. The difficulty of the music score not only reflects learners' ability levels, but also gives an objective benchmark for the evaluation of learners. When teachers consider the difficulty of the music score itself as part of the evaluation of a student's performance, this highlights their recognition of their students' learning processes, enabling them to discover learner shortcomings and improving the effect of evaluation on learning.

Due to the lack of unified quantitative standards, the current classification of the difficulty level of music scores mainly depends on the subjective judgment of experts or professionals in the field, meaning that learners are forced to rely on existing teaching materials, resulting in a lack of diversity and autonomy in the choice of learning resources. In the process of selecting tracks for learners, teachers are inevitably influenced by their own musical preferences. Although the tracks recommended to students can determine the difficulty coefficient, it is obvious that the recommended tracks are narrow in scope and personal music preferences are obvious when they are in line with learners' musical preferences or else reflect changing musical trends. Making full use of big data and artificial intelligence technology to objectively and accurately classify and grade piano music scores, which can effectively balance teachers' or students' preferences in the difficulty coefficient of music and music categories in the teaching process, can enable learners to independently choose appropriate learning resources and contribute to personalized piano teaching.

At present, although relevant studies have made preliminary progress in the difficulty identification of piano music scores, the piano is a multi-voice instrument and the complexity of compositions, compounded by the differences in the demands of different teaching scenarios, requires a more scientific, efficient and objective intelligent analysis algorithm to optimize difficulty classification. Only by improving the accuracy and precision of "analysis in difficulty application scenarios" can we truly meet the requirements of real teaching situations. In this study, a difficulty level recognition model of piano music was established using an LSTM neural network based on the timing characteristics of the music score. At the same time, in order to solve the overfitting problem of the model for the small data set, we also propose a data optimization method based on pitch migration. The experimental results show that the recognition accuracy of our proposed model is better than those of existing methods, the recognition accuracy of the three categories of data sets reaching 88% (the existing method is 78%). Finally, in the independently developed piano teaching platform, based on the requirements of teaching objectives and the current ability levels of learners, we use this model to recommend suitable music scores for learners to practice and test, effectively improving the learning effect.

Related work

Although artificial intelligence technology is commonly used in music analysis, conversion, generation and other, similar problems, there are few studies on the automatic recognition of musical score difficulty level. Sébastien et al. (2012) regarded the recognition of musical score difficulty level as a problem of musical score classification. They took the ‘MusicXML’ score as the analysis object and defined seven characteristics of musical score difficulty, such as the rhythm, fingering, and hand displacement required for performance, chord, rhythm and polyphony. Since these features interact with each other in a complex way with regard to the difficulty level of the score, they first obtained the key features by principal component analysis (PCA), and then realized the difficulty classifier by using an unsupervised clustering algorithm. The score difficulty classifier, based on the quintessential features, achieved a classification accuracy of 0.66 on the data set of 50 samples.

Chiu & Chen (2012) regarded the recognition of musical score difficulty level as a regression problem. Based on statistical features such as pitch entropy, range, average pitch and performance speed were added. The ‘RreliefF’ algorithm was used to filter the feature set, and other algorithms, such as linear regression (Robnik-Šikonja & Kononenko, 2003), multilinear regression, step-up regression and support vector regression (Smola & Schölkopf, 2004), were used to construct the music difficulty prediction model on the optimal special solicitation. The results of the model evaluation experiment showed that the performance of the support vector regression model was the best, and the regression coefficient (R2) of the model was close to 0.4 on the data sets of the 4 difficulty and the 9 difficulty categories.

The study by Nakamura et al. (2014) showed that scores with more unusual fingerings were very difficult to play. Consequently, Ramoneda et al. (2022) used fingering characteristics in piano playing techniques to analyze the difficulty of the music score; by using two kinds of finger-extracting systems, namely knowledge-driven and data-driven, four kinds of finger-extracting sequences were obtained from the music score, with ‘XGBoost’ and ‘DeepGRU’ deep learning models being utilized to construct music score difficulty classifiers on different finger-extracting sequences. At the same time, in order to make the study more relevant to the educational setting, they constructed a three-difficulty level data set that utilized the musical score used in piano teaching. In this dataset, the performance of the ‘DeepGRU’ model was the best, the classification accuracy reaching 0.78.

Ghatas et al. (2022) believed that the features based on manual definition had limitations and were not applicable to the end-to-end deep learning model. They therefore divided the music score into equal-length fragments and used the convolutional network to extract the difficulty features, and then combined the features obtained by the neural network and the manually defined features to form the feature input of music score difficulty. Finally, a multiple perceptron classifier was used to classify the music. The highest accuracy of this method was 0.80 and the highest F1 score 0.76 on the data set of three difficulty-level categories.

The evaluation of music score difficulty itself has a high degree of subjectivity, and the factors that need to be considered in different subdivision themes are uniform, so the construction of a music score difficulty evaluation model requires a certain complexity. There are subjective deviations in the artificially defined features; at the same time, the normalization, filtering, and other numerical calculation processes will also produce errors. More importantly, some statistical characteristics (such as vocal range, rhythm, and intervals) cannot describe the dynamic changes in the time sequence of the music itself. The difficulty of analyzing musical scores from the performance process is primarily reflected in the changes in distinct nodes, such as those to unit beat, post-bar rhythm, scale, and rotation. The model built on a time series has a natural structural advantage in capturing such information.

Approach

The construction of a time series classification model usually involves two main tasks: firstly, it is necessary to find a time series representation that can fully describe the research object, specifically one that solves the problem of time dimension division and the feature representation of a single time step; secondly, the time series is used as the input to construct the classification model. In order to more comprehensively represent the changes of musical notation in time sequence and make the data extensible, this chapter proposes improvements to the existing musical notation time sequence representation method, together with designing an automatic recognition model of musical notation difficulty features based on the improved representation method.

Sequential representation of piano scores

MIDI (Musical Instrument Digital Interface) (David, 2019), as the international standard of digital music electronic synthetic instruments, is the most extensive music standard format in the field of music programming. MIDI records music through the digital control signals of notes, and uses ‘MIDI Message’ to describe the information of the music to be played, such as at the specific moment, what type of instrument will be used, which notes will start to sound, which notes will sound at the end, whether the tone of the beat changes, etc. For any keyboard instrument, including the piano, each note message contains a key number and force. The key number corresponds to the pitch, indicating the frequency at which the note is made, and ranges from 0 to 127; the force represents the weight of the “down” key, and the “stop” key is represented by a message with a force of 0. [9] In addition, MIDI files contain other metadata that are used to control playback.

Even though the “MIDI message” stream accurately captures the temporal characteristics of the score, algorithmic models cannot use it as an input directly. After being divided by time steps in related studies, it is typically encoded as a two-dimensional Piano Roll matrix (Dervakos et al., 2021). Taking the matrix D_{k*t} ($k \in \{0,1,2...127\}$; $t \in N$ is the time step) to express the piano roll, each column of the matrix corresponds to a time step (e.g., quarter note duration), the row number corresponds to the key number in MIDI, and the element $d_{ij} = 1$ of the matrix indicates that the key number i is pressed at the jth time step, while $d_{ij} = 0$ indicates that it has not been pressed.

There are two segmentation techniques for the time dimension that are currently used: fixed step and fixed duration. Fixed step represents the duration of each unit beat by the same number of steps. Fixed duration divides the duration of notes by the base time. The challenge with fixed duration lies in determining the base duration, whereas the division approach of fixed steps frequently results in excessively lengthy coding sequences that hinder the algorithm’s ability to converge. In order to achieve convergence, we created a relative reference coding, selecting 1/n of a unit beat as the time step for any score (n is typically 16, 32, or 64).

Considering that splitting by time step will truncate the notes with large durations into multiple small fragments, and in order to distinguish whether the keys with the same number and in the pressed state in adjacent time steps are the same notes in the corresponding score, a matrix S_{k*t} ($k \in \{0,1,2...127$; $t \in N$ is the time step length$\}$) was defined to record this state. The element $s_{ij}=1$ of the matrix indicated whether the pressed state of the ith key in the j_{th} time step was a continuation of the j-1 step; $s_{ij} = 0$ if it was not a continuation of the pressed state. The score was finally represented as a piano roll by aligning the matrices D and S in time steps and then splicing them together as P = (DS), where P is a two-dimensional matrix of m*n, m is the double of the number of keys, and n is the overall number of time steps after splitting. M = 88*2 if the MIDI representation is a piano score (generally, a piano score has only 88 keys, corresponding to the MIDI key numbers 21 to 108). Figure 1 shows a score fragment of the Piano Roll sequence.

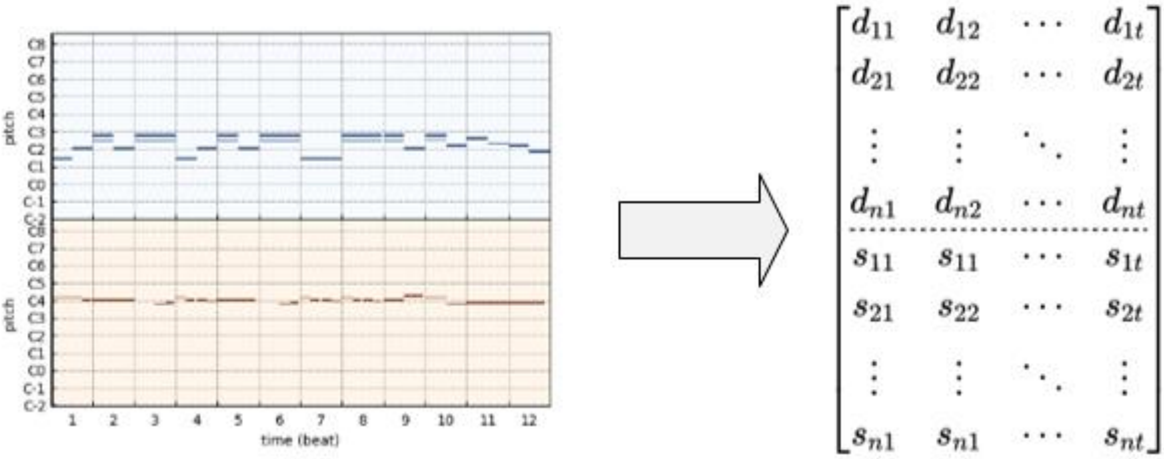


Figure 1 *Music sheet encode example by proposed method*

Classification method

An LSTM network is a recursive network designed to optimize the gradient disappearance problem of a recurrent neural network (RNN) (Pascanu et al., 2013). A logic gate module is utilized in neurons in order to replace activation function in RNN, so that it can store and transmit context information between, and realize the long-term dependence of, time series (Cheng et al., 2016). In LSTM-based sequence prediction and classification problems, LSTM output state sequence $\{h_1,h_2... h_t\}$ is often regarded as the characteristic information of the original sequence, which is seen as the difficulty related information implied in the musical score sequence in this study. Since the original LSTM only considers the sequential dependence of the time series, and the reverse characteristics of adjacent notes in the score and the difficulty of the combined sequence may also be related to the score, the experiment of this research also uses the ‘Bi-LSTM’ and ‘Attention-LSTM’ network models improved from LSTM.

The state sequence {h_1,h_2,... h_t } containing music difficulty information is obtained through the LSTM network. After h_t, the classification task can be expected to be completed only after subsequent processing. Figure 2 describes the general structure of the LSTM-based classifier, which is composed of the fully connected LSTM network layer, ‘SoftMax’ function and ‘Argmax’ function, where ‘hlast’ represents the output state of the last neuron in the LSTM network. In other words, as it pertains to the difficulty feature vector of the score, the full connection layer is the high-dimensional state vector ‘hlast’ transformed into a vector of the same dimensions as the category, where the ‘SoftMax’ function is used to calculate the probability of each category and the ‘Argmax’ function selects the maximum probability category from the probability distribution.

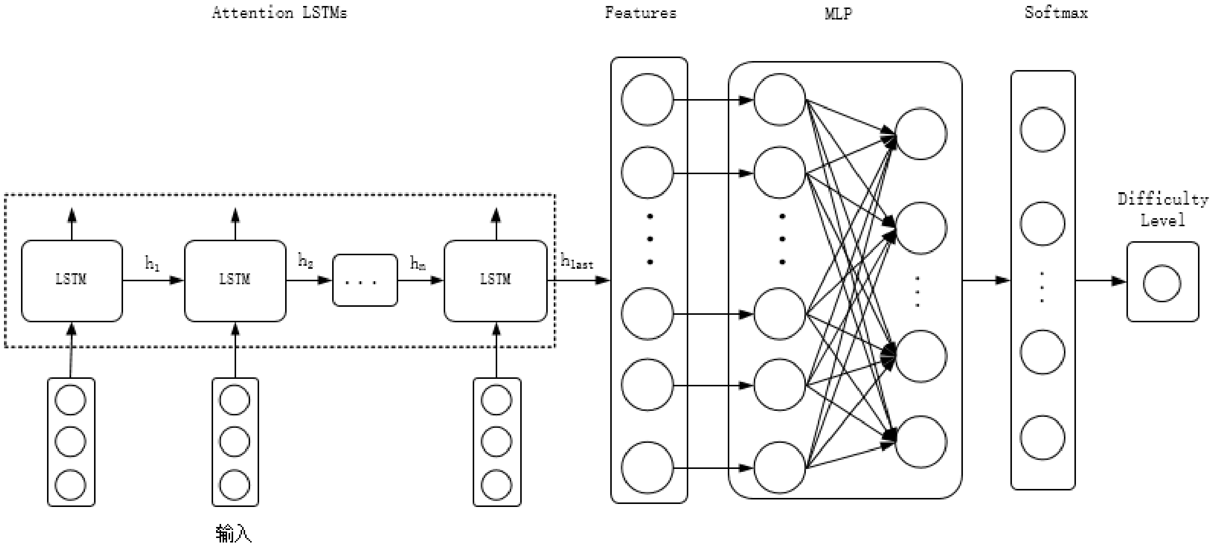


Figure 2 *Framework of proposed music sheet classification model*

Since the LSTM network in our model is used to extract the difficulty features from the score sequence, and there is no Encoder-Decoder structure, we use an attention mechanism that was initially proposed in the Encoder-Decoder structure for obtaining the contextual information of the input sequence (Sundermeyer et al., 2012) and which can be propagated in one direction (Elliot, 2016). As seen in Figure 3, when determining the output of the current time step, the output state of the previous n steps is taken into full consideration, with the attention part being calculated as shown in Equations 1-3:

$$u_i^t = v^T \tanh(W_h h_i + W_c c_t) \quad (1)$$

$$a_i^t = \mathrm{softmax}(u_i^t) \quad (2)$$

$$h_t' = \sum_{i=t-n}^{t-1} a_i^t h_i \quad (3)$$

where vectors v, W_h, and W_c are the parameters to be learned in the model, c_i is the state of the LSTM cell at the current time step, h_i is the output of the LSTM cell at the previous n time steps (h_{t-n},... ,h_{t-1}); the weights of each time step are calculated using the state of the current time step and the outputs of the previous n time steps.

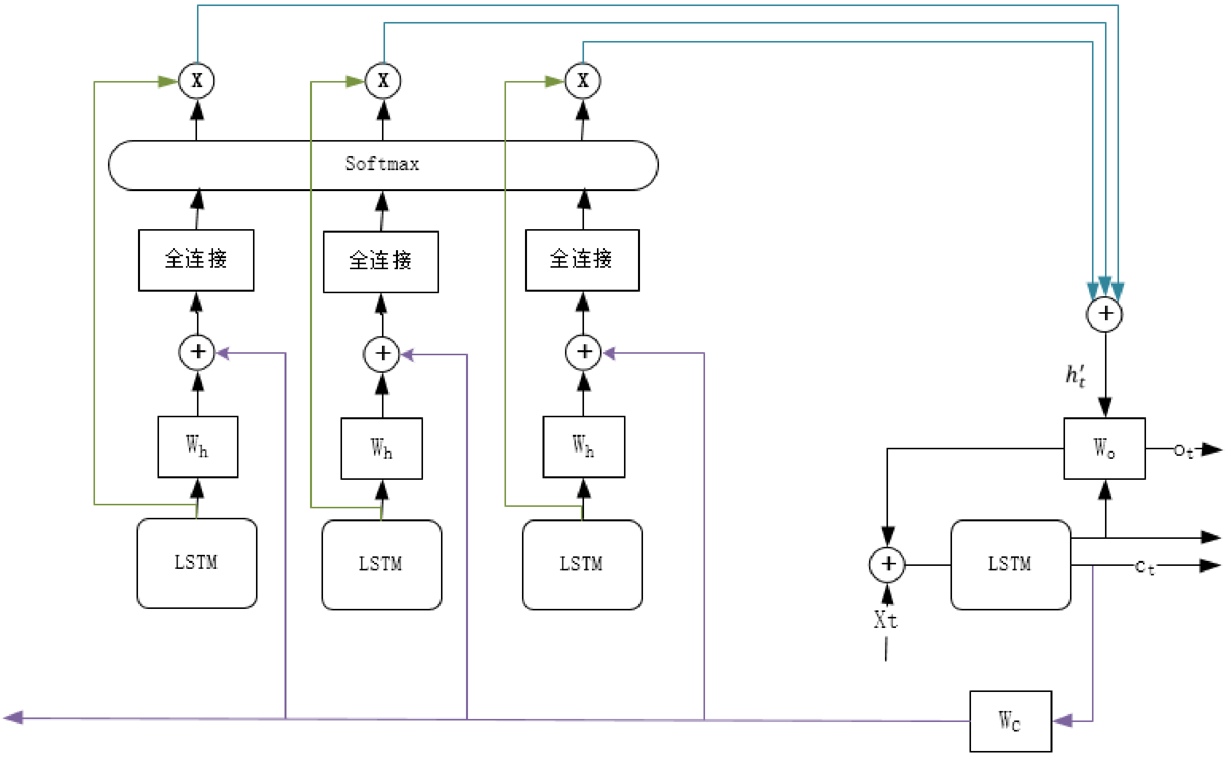


Figure 3 *The inner structure of attention LSTM cell unit*

The attention value corresponding to each time step is obtained after 'Softmax' normalization, following which the attention value is weighted and summed with the output to obtain the attention representation $h_t'$for the n time steps. The vector obtained by concatenating the attention h_t' with the hidden state $\boldsymbol{h}_t^*$ of the last time step is the difficulty feature value of the score sequence used in Equation 4:

$$\boldsymbol{h}_t = [\boldsymbol{h}_t', \boldsymbol{h}_t^*] \tag{4}$$

In the experimental section we will use the basic LSTM and 'Bi-LSTM' models as benchmarks to verify the performance of the attention mechanism in the score difficulty classification problem.

Experiment

Dataset

We selected two sheet music datasets, ‘80notes’ (8notes.com, 2020) and ‘Mikrokosmos’ (Pedro, 2021/2022), which have been used in previous literature as experimental data. The ‘80notes’ dataset contains four difficulty levels (Beginner, Easy, Intermediate, Advanced) and ‘Mikrokosmos’ contains three difficulty levels (Beginner, Moderate, Professional). Due to the large length of the segmented scores in the Advanced category of ‘80notes’ and the performance problems of the LSTM network in dealing with very long sequences, only the first three levels are used in this study.

Preprocess

The number of scores and the average length of each difficulty level in the two datasets are shown in Table 1, where the average length of scores is counted in the number of sixteenth notes.

Table 1 *Dataset statistical analysis*

Dataset Name	Difficulty Level	Count	Average steps(1/16$_{th}$ note per step)
‘80notes’	Beginner	91	245.54
	Easy	624	449.7

Dataset Name	Difficulty Level	Count	Average steps(1/16th note per step)
	Intermediate	1144	773.97
‘Mikrokosmos’	Beginner	52	299.12
	Moderate	53	399.62
	Professional	30	731.73

Data enhancement.

Data enhancement is a data preprocessing technique used in deep learning to improve the performance of models on small data sets and is commonly used in research related to computer vision, such as flipping, rotating, and the offsetting of images. From the features selected in the literature (Chiu & Chen, 2012), it can be seen that the difficulty of a musical score is related to features such as the length of the note itself and the change in the pitch of adjacent notes. Therefore, expansion of the sample size can be achieved by shifting the pitches in the whole score up and down in order to form a new score with the same difficulty level. Figure 4 shows a score fragment before and after pitch shifting, whereby throughout the process the duration of all the notes remains the same.

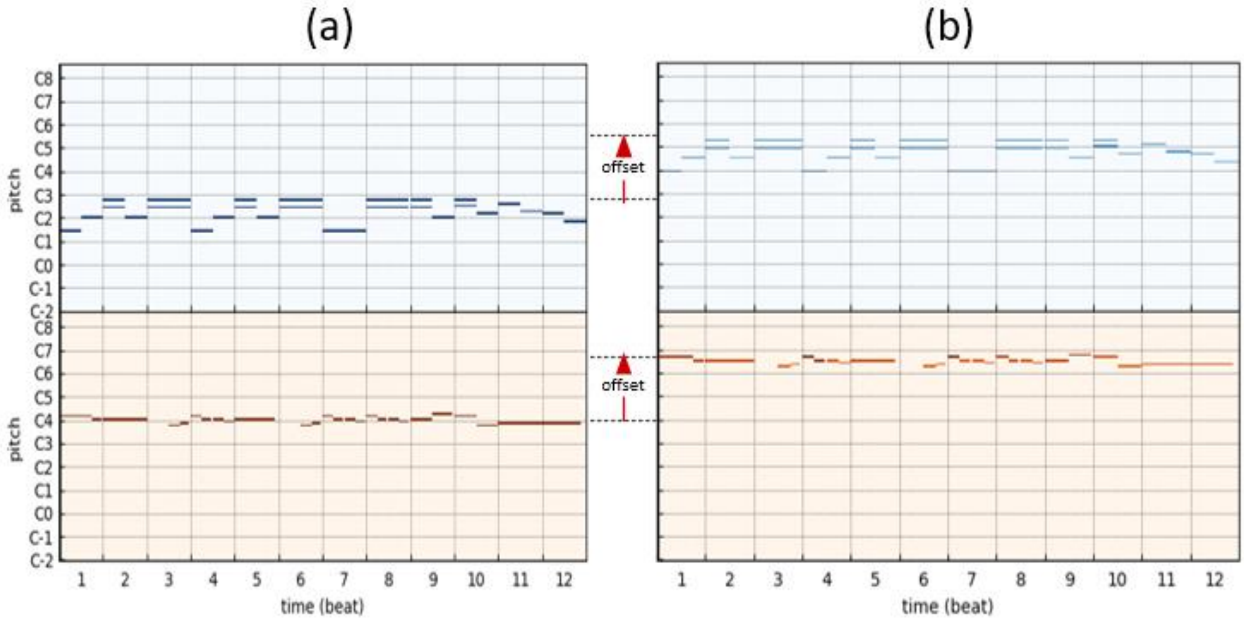


Figure 4 *Pitch offset example of one fragment, where (a) is the original score and (b) is the offset one*

According to this method, the dataset was expanded with different offset units and the pitches were guaranteed to be between 22 and 109, so the maximum offset unit was set as a threshold with the lowest note of the offset score not lower than 22 and the highest note not exceeding 109. After data selection and expansion, the number of scores in the final dataset was 701 for each level in ‘note80’, totaling 2804 scores; there were 292 scores for each difficulty level in the ‘Mikrokosmos’ dataset, totaling 876 scores. In each set of experiments, the original data was used as the test set and the extended data as the training set.

Experimental setup

The number of LSTM, ‘bi_LSTM’ and ‘att_LSTM’ networks in the experiment was 2, and the number of hidden state units in each layer 128 and 64, respectively; the attention window size was 64 in ‘att_LSTM’; to prevent model overfitting, each layer used the ‘DropOut’ mechanism with a pass rate of 0.5, and added L2 regularization with a weight of 0.0001 for all trainable parameters to the model loss. The batch gradient descent learning method was used in the training process, using the ‘AdamOptimizer’ gradient optimizer with a learning rate of 0.001, a batch size of 256 (i.e., 256 samples were randomly selected for each iteration), and a gradient cropping threshold of 5.

Since the difficulty level of the score is a multi-category problem, we used four metrics, Accuracy, Recall, Precision, and F-value (F1), for comparative analysis (Equation 5~6). Among them, Recall, Accuracy, and F-value, calculated separately by different categories, were used to find the average value. For each difficulty level, TP was the number of samples correctly identified as level I; TN was the number of other difficulty levels correctly identified; FP was the number of difficulty level i identified as other difficulty levels, and FN was the number of other levels identified as level i (i = the difficulty level):

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Number of Samples}} \tag{5}$$

$$\text{Recall} = \sum_{i=1}^{n} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} (\text{n is the total difficult level number}) \tag{6}$$

$$\text{Precision} = \sum_{i=1}^{n} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \quad (\text{n is the total difficult level number}) \tag{7}$$

$$F_1 = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

The experimental procedure consisted of two sets of independent experiments for model validation and data enhancement method validation. In the model comparison experiments, the recognition accuracy of different models was compared with the enhanced ‘Mikrokosmos’ dataset; in the data enhancement validation experiments, the ‘att_LSTM’ model was selected to test the effectiveness of the proposed data enhancement method by comparing its effect on the extended dataset and the original dataset.

Results and Discussion

The results of the model validation are shown in Table 2.

Table 2 *Enhanced Dataset classification results*

Model Name	Accuracy	Precision	Recall	F1
LSTM	0.85	0.85	0.85	0.85
‘att_LSTM’	0.90	0.88	0.89	0.88
‘bi_LSTM’	0.82	0.82	0.82	0.82

Precision, recall, accuracy, and F-value of the ‘att_LSTM’ model were the best among the three models (0.90, 0.88, 0.89, and 0.88, respectively), that of precision exceeding other methods in the literature on the same data set and indicating the potential of ‘att_LSTM’ for extracting the difficulty features of musical scores.

In order to compare the differentiation of the model on varying difficulty levels, we selected the recognition of ‘att_LSTM’ model in different categories, as shown in Tables 3 and 4.

Table 3 *Confusion matrix of LSTM model on ‘Mikrokosmos’ dataset*

	Beginner	Moderate	Professional
Beginner	0.85	0.15	0
Moderate	0.18	0.61	0.21
Professional	0	0.36	0.67

Table 4 *Confusion matrix of an ‘att_LSTM’ model on ‘80notes’ dataset*

	Beginner	Easy	Intermediate
Beginner	0.86	0.14	0
Easy	0.16	0.64	0.20
Intermediate	0	0.34	0.66

The results show that there is a higher probability of recognition error for adjacent difficulty levels, in which 0.15 of difficulty level 1 is incorrectly recognized as difficulty 2; 0.57 of difficulty level 3 is incorrectly recognized as difficulty 2. Also, since difficulty level 2 is an intermediate difficulty level, there is a possibility that the samples are incorrectly identified as difficulty level 1 and difficulty level 3, and that the incorrect identification rates are 0.18 and 0.21, respectively. At the same time, ‘att_LSTM’ model’s ability to distinguish adjacent difficulty levels still needs to be improved.

Considering that the enhanced dataset was used in the model validation experiment in order to verify the effectiveness of the proposed data enhancement method, we chose the ‘att_LSTM’ model with the highest recognition accuracy; these evaluation indexes on the original dataset and the enhanced dataset are shown in Table 5.

Table 5 *Classification results of an ‘att_LSTM’ model on original and enhanced dataset*

Dataset Name	Type	Accuracy	Precision	Recall	F1
‘Mikrokosmos’	original	0.63	0.65	0.67	0.67
	enhanced	0.83	0.82	0.84	0.82
‘80notes’	original	0.65	0.62	0.67	0.67
	enhanced	0.79	0.81	0.80	0.77

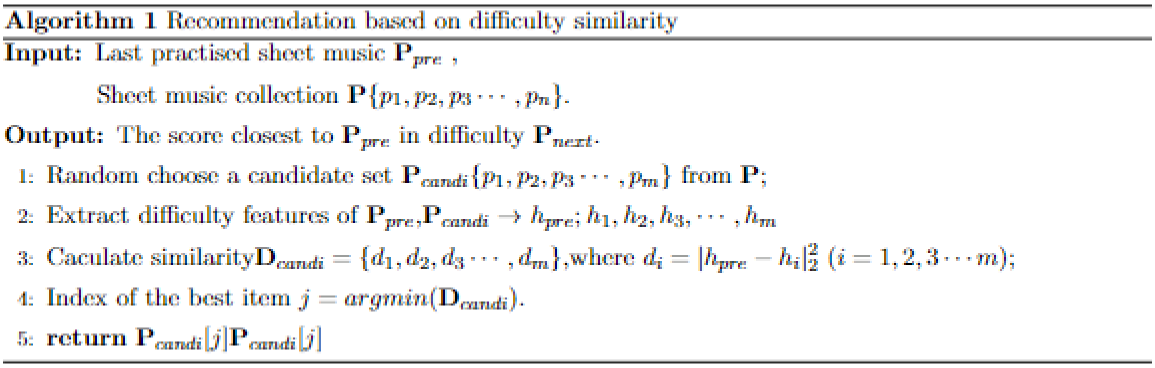
Recognition of the ‘att_LSTM’ model’s accuracy on the extended dataset was 0.90, which is much higher than that on the original dataset. Therefore, it can be concluded that the data enhancement method proposed in this paper can effectively improve the accuracy of the difficulty recognition model of the score, as well as confirming that the change of minimum and maximum pitches in the score does not affect the difficulty level of the score itself within the same pitch range.

The data in the tables shows that although the difficulty features extracted by the LSTM network lack interpretability compared with the manually defined ones, the resulting difficulty features have good representability and can more accurately distinguish between scores of different difficulty levels. Our proposed data augmentation method effectively solves the problem of small data sets within the larger issue of sheet music difficulty recognition, and provides a reference for other deep learning models that can be applied to such problems.

Educational applications

When learning the piano, the learning space is constrained by the learning material's rigidity content, which also erodes students' confidence and interest when they repeatedly play the same compositions with little variation. This study integrates musical difficulty evaluation into the teaching process and uses a suggestion system based on musical difficulty level to prevent thems from becoming frustrated or worn out. Therefore, when students begin to feel discouraged or exhausted, it is advised that they practice a score with a comparable level of difficulty.

The final output h_{last} of the LSTM network in the sheet music difficulty level recognition model can be considered as the feature value of the sheet music difficulty level. When retrieving teaching resources, making individualized recommendations, and conducting personalized assessments, the similarity of characteristics can be utilized to differentiate between the varying degrees of difficulty of different pieces of sheet music. Figure 5 displays the recommendation process based on the difficulty similarity of the score.



```
Algorithm 1 Recommendation based on difficulty similarity
Input: Last practised sheet music $\mathbf{P}_{pre}$ ,
    Sheet music collection $\mathbf{P}\{p_1, p_2, p_3 \cdots, p_n\}$.
Output: The score closest to $\mathbf{P}_{pre}$ in difficulty $\mathbf{P}_{next}$.
  Random choose a candidate set $\mathbf{P}_{candi}\{p_1, p_2, p_3 \cdots, p_m\}$ from $\mathbf{P}$;
  Extract difficulty features of $\mathbf{P}_{pre}$,$\mathbf{P}_{candi} \rightarrow h_{pre}; h_1, h_2, h_3, \cdots, h_m$
  Caculate similarity$\mathbf{D}_{candi} = \{d_1, d_2, d_3 \cdots, d_m\}$,where $d_i = |h_{pre} - h_i|_2^2$ $(i = 1, 2, 3 \cdots m)$;
  Index of the best item $j = argmin(\mathbf{D}_{candi})$.
  return $\mathbf{P}_{candi}[j]\mathbf{P}_{candi}[j]$
```

Figure 5 *Algorithm flow of score recommendation based on difficulty similarity*

Since there are only a finite number of scores in the resource library, the actual application recommends multiple scores of varying degrees of difficulty as candidate sets for learners to choose from on their own after setting a threshold in order to avoid making duplicate recommendations. As seen in Figure 6, when students practice a piece in the textbook using our self-created piano teaching APP, multiple scores of comparable difficulty are suggested for them so that they can practice on various scores depending on their learning circumstances.

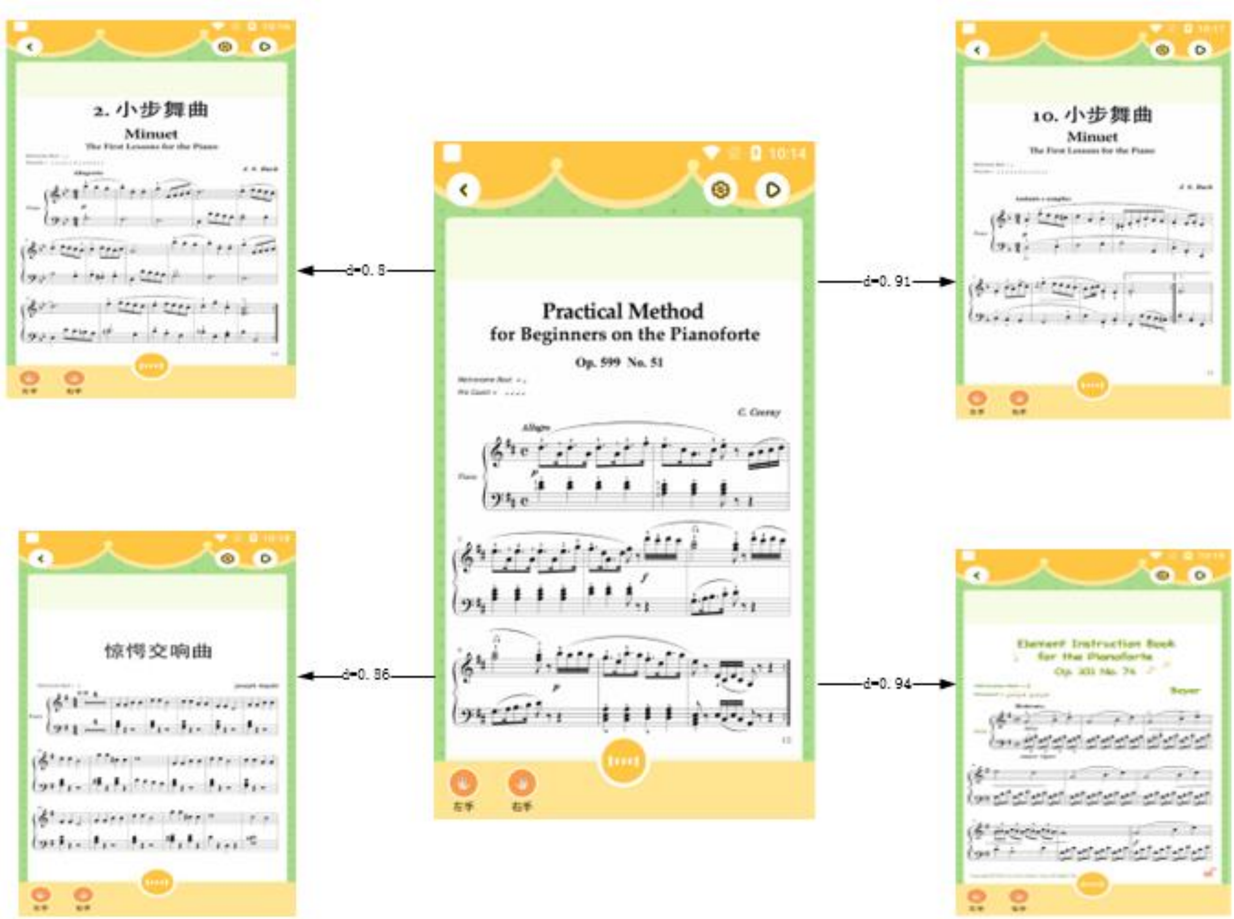


Figure 6 *Recommended algorithm example*

Conclusions and Future Work

This study examined the utilization of a classifier based on an LSTM neural network in order to categorize the degree of difficulty of musical scores and validated it using real data. The experimental findings indicate that given enough experimental data, the classifier proposed in this paper can effectively extract difficulty-related information from score sequences, and its classification accuracy of 88% is superior to that of the best study to date (78.67%). Furthermore, the comparison of various coding techniques in terms of classification effect establishes from one side that the level of difficulty of a score should be assessed by factoring both the complexity of the performance and the score structure, both of which are essential requirements for students' score recognition and performance ability in piano learning. Finally, we explored how score difficulty could be applied practically in piano instruction and designed and implemented a system for recommending valuable functions based on the score difficulty recognition model.

We will continue to refine the classification model for sheet music and increase the accuracy with which difficulty level is recognized in the follow-up study. Further research will be done on the application of sheet music difficulty in individualized learning and instruction sessions, such as adaptive evaluation and automated testing based on our piano teaching platform. This will enhance the platform's individualized educational experience and help the system become more intelligent and personalized.

References

8notes.com. (2020). Retrieved December 2, 2022, from Free sheet music on 8notes.com website: https://www.8notes.com/

Cheng, J., Dong, L., & Lapata, M. (2016, September 20). *Long Short-Term Memory-Networks for Machine Reading*. arXiv. Retrieved from http://arxiv.org/abs/1601.06733

Chiu, S.-C., & Chen, M.-S. (2012). A Study on Difficulty Level Recognition of Piano Sheet Music. *2012 IEEE International Symposium on Multimedia*, 17-23. Irvine, CA, USA: IEEE. https://doi.org/10.1109/ISM.2012.11

David, B. (2019). Standard MIDI file format, updated. Retrieved December 2, 2022, from http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html

Dervakos, E., Kotsani, N., & Stamou, G. (2021). Genre Recognition from Symbolic Music with CNNs. *Artificial Intelligence in Music, Sound, Art and Design*, 98-114. Springer, Cham. https://doi.org/10.1007/978-3-030-72914-1_7

Elliot, W. (2016, July 15). Generating Long-Term Structure in Songs and Stories. Retrieved December 2, 2022, from Magenta website: https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn

Ghatas, Y., Fayek, M., & Hadhoud, M. (2022). A hybrid deep learning approach for musical difficulty estimation of piano symbolic music. *Alexandria Engineering Journal*, *61*(12), 10183-10196. https://doi.org/10.1016/j.aej.2022.03.060

Nakamura, E., Ono, N., & Sagayama, S. (2014). *MERGED-OUTPUT HMM FOR PIANO FINGERING OF BOTH HANDS*. 6.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In S. Dasgupta & D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning* (Vol. 28, pp. 1310-1318). Proceedings of Machine Learning Research: PMLR. Retrieved from https://proceedings.mlr.press/v28/pascanu13.html

Pedro. (2022). *Mikrokosmos-difficulty dataset*. Retrieved from https://github.com/PRamoneda/Mikrokosmos-difficulty (Original work published 2021)

Ramoneda, P., Tamer, N. C., Eremenko, V., Serra, X., & Miron, M. (2022). Score Difficulty Analysis for Piano Performance Education based on Fingering. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 201–205. https://doi.org/10.1109/ICASSP43922.2022.9747223

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning*, *53*(1), 23-69. https://doi.org/10.1023/A:1025667309714

Sébastien, V., Ralambondrainy, H., Sébastien, O., & Conruyt, N. (2012). *SCORE ANALYZER: AUTOMATICALLY DETERMINING SCORES DIFFICULTY LEVEL FOR INSTRUMENTAL E-LEARNING*. 6.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199-222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. *Thirteenth Annual Conference of the International Speech Communication Association*.