Copyright ⓒ 2022 KSII

Application and Research of Monte Carlo Sampling Algorithm in Music Generation

Jun MIN[1], Lei WANG[1*], Junwei PANG[1], Huihui HAN[1], Dongyang Li[1], Maoqing ZHANG[1] and Yantai HUANG[2]
[1] College of Electronics and Information Engineering, TongJi university
Shanghai, 201804, China
[e-mail: newscenter@tongji.edu.cn]
[2] College of automation and electrical engineering, Zhejiang University of Science and Technology
Hangzhou, 310023, China
[e-mail: xcb@zust.edu.cn]
[*]Corresponding author: Lei Wang

Received October 16, 2021; revised May 14, 2022; revised June 13, 2022; revised August 5, 2022; accepted September 5, 2022; published October 31, 2022

Abstract

Composing music is an inspired yet challenging task, in that the process involves many considerations such as assigning pitches, determining rhythm, and arranging accompaniment. Algorithmic composition aims to develop algorithms for music composition. Recently, algorithmic composition using artificial intelligence technologies received considerable attention. In particular, computational intelligence is widely used and achieves promising results in the creation of music. This paper attempts to provide a survey on the music generation based on the Monte Carlo (MC) algorithm. First, transform the MIDI music format files to digital data. Among these data, use the logistic fitting method to fit the time series, obtain the time distribution regular pattern. Except for time series, the converted data also includes duration, pitch, and velocity. Second, using MC simulation to deal with them summed up their distribution law respectively. The two main control parameters are the value of discrete sampling and standard deviation. Processing the above parameters and converting the data to MIDI file, then compared with the output generated by LSTM neural network, evaluate the music comprehensively.

Keywords: Music generation; MIDI; Monte Carlo; Data convert.

This work was funded by Science and Technology Winter Olympic Project(Grant number 2018YFF0300505) and the Joint Fund of Zhejiang Provincial Natural Science Foundation(Grant number LHY20F030001). Jun Min is with the School of Electronic and Information Engineering, Tongji University, Shanghai, China(e-mail: minjun@ tongji.edu.cn). Lei Wang is now with the Department of Control Science and Engineering, Tongji University, Shanghai, China(e-mail: wanglei@ tongji.edu.cn).

http://doi.org/10.3837/tiis.2022.10.008 **ISSN : 1976-7277**

1. Introduction

COMPOSSING melodious music is a challenging task since many musical elements need to be considered, such as pitch, rhythm, chord, timbre, musical form, and accompaniment. In the past, music composition was usually accomplished by a few talented people[1-4]. The algorithmic composition, which formulates the creation of music as a formal problem, facilitates the development of algorithms for music composition.

The algorithmic composition enables automatic composition by using computers and mathematics. In 1959, Hiller and Isaacson first programmed the Illinois automatic computer (ILLIAC) to generate music algorithmically[5]. The music piece was composed by the computer and then transformed into a score for a string quartet to perform. At present, the company Google and Sony are trying and exploring actively this field. In Google's magenta project, based on Tensorflow[6], the latest achievement is to generate a 90-second melody. FlowMachines[7], an artificial intelligence system developed by Sony company, can learn various music styles in a huge music library, create different styles of music with unique style transformation, optimization, and interaction technology independently.

From a mathematical perspective, composing music can be viewed as a stochastic process and, therefore, mathematical models such as Markov chains are useful for composition[8]. The music composition using mathematical models has the advantages of low complexity and fast response, which are adequate for real-time application. In recent years, other technologies have appeared in papers published on automatic composition, such as hierarchical structure[9], cellular automata theory[10], experience-based composition[11], answer set programming[12], possibility construction spatial thinking model[13], probability logic[14], etc.

The advances of artificial intelligence (AI) promote its application to algorithmic composition. Spurred by the progress of deep neural networks, Support Vector Machine (SVM)[15], Deep Neural Networks (DNN)[16], Convolution Neural Network (CNN)[17][18], especially the Recurrent Neural Network (RNN)[19-23] have gathered renewed interests these days. In particular, recurrent neural network (RNN) has aroused people's interest again. Through the accumulation and certification of practice, most of the network models used to generate music are based on recurrent neural networks and their variants. These techniques have been applied to music composition. In particular, In RNN, the LSTM network is used to process and generate music files commonly[24]. Although the LSTM can model music and synthesize new note sequences, they only can keep the time unchanged. In RNN, LSTM is used to process and generate music files commonly. Although the LSTM can model music and synthesize new note sequences, they can just keep the time unchanged. For example, when the training data move time steps forward or backward, the model will not recognize them as the different data separately among the processing of music training. When the training data move several semitones up or down, the above models can not keep the transposition invariant, which will increase the time cost unnecessary.

Therefore, in the actual operation, it is not difficult to find that if only through the LSTM network to generate the music, both the system running time and the music timbre are all not satisfactory. To realize the transposition invariant and equip the system to recognize the interval relationship between chords, the relative position of the notes rather than the absolute position in the whole score must be considered when designing the music generation structure.

In light of the shortcomings of the above methods during the process of music generation, the main contributions of this paper are listed as follows.

1) Model the music sample structure as a probability distribution, that is, the system will count the probability of each note appearing in each specific bar and each tone.

2) Proposing that the system can realize the shift-invariance by learning the notes distribution probability in the music sample, and the system can recognize the interval relationship between chords.
3) Establish a complete music evaluation index, and evaluate the generated music.

The MC method can simulate the music generate process fantastic. The advantage of using the MC algorithm to create music is the system will consider the relative position between the notes rather than the absolute position in the whole score, and the running time will be shortened greatly. At the same time, the process of music generation can be regarded as a process that combines theoretical experience and accidental inspiration. For the theoretical experience of music generation, the MC algorithm can train the data set, then summarize the rules of notes, and learn the distribution of the note in the whole score. However, the influence of accidental inspiration in music generation needs to consider the breakthrough and innovation from different directions. In addition, the evaluation of compositions is a paramount issue that needs to be addressed in AI-based composition systems.

The paper is organized as follows. Section II reviews the studies on music generation by artificial intelligence algorithms. Section III introduces the method and theory of recommendation and recapitulates the process of the experiment. Section IV summarizes the result analysis and evaluation. Finally, Section V provides some suggestions for future research topics and presented a summary of this paper.

2. Related work

With the application of deep learning techniques for music, generation has been rapidly advancing, there are some application examples of using algorithms to generate music. Smith et al. use several variants of multilayer neural networks to generate music, which can understand the structure and rhythm of music well. Different from the traditional model, this model takes the music data as a whole. At the same time, it divides the music data by time, and each part can be regarded as a music sample. These segmented parts are processed into more advanced music structures abstractly, then repeated process in the main structure of the system until the final result is generated[25]. Nayebi et al.[26] compare the practicability of unit structure in music generation between LSTM network with Gated Recurrent Network (GRU), these models use the WAV format music files as the input samples of the network, output the waveform of the music samples by giving a random number. However, after repeated experiments, it has taken too much time to process music files of the WAV format, this idea does not try to carry out a lot of testing data, lacks a certain persuasion. Bonn G et al.[27] put forward the answer set programming method, which uses the answer set to build an automatic composting system called Anton. Although this method can generate a short melody well, it can not generate complete music and there are also have some shortcomings in rhythm coding and melody processing. Aguilera G et al.[28] propose a probabilistic logic for generating counterpoint notes in the Polyphonic Logic method, which uses a probability algorithm to generate a fixed melody counterpoint music, but the output is unstable even input the same music sample in the system, and the algorithm without considering the genre characteristics of melody. Voss and Clark[29] observed and proposed the $1/f$ noise for composition. They have used independent 1/f noise sources in a simple algorithm to determine the duration and pitch of successive notes of a melody. The music obtained by this method was judged by most listeners to be much more pleasing than that obtained using either a white noise source or an $1/f^2$ noise source. Sertan and Chordia[30] utilized the variable-length Markov model that considers the pitch, rhythm, instrument, and key, to predict the subsequent

sequence of Turkish folk music. The thesis also introduces pitch-related viewpoints that are specifically aimed to model the unique melodic properties of making music. Prechtl et al.[31] generated music for games by using the Markov chains with musical features such as tempo, velocity, volume, and chords. They outline an implementation of the approach in an actual game, focusing primarily on how the music system traces the game's emotional narrative by periodically querying certain narrative parameters and adjusting the musical features of its output accordingly.

3. Proposed method

Through the introduction of the ways in the first two chapters, it can be concluded that the time cost and the final music quality are two important standards of the evaluation criteria in music generation. Although the process of using algorithmic to synthesize the music still need human to participate, this can be regarded as an effective auxiliary means to promote the final music generation. Also, the music generated by different models have their characteristics, but if the system only contains one method the final audio files will not be satisfactory. When designing a music generation system, it needs to combine a variety of means, the system will also develop in the direction of hybrid. From the perspective of the system's structure, the process of music generation is mainly divided into three parts, namely, music data preprocessing, feature extraction, and music generation. The first is the preprocessing of the music data, which needs to convert MIDI files into music matrix or others, convert music files into digital symbols, which can be processed by computer directly. Secondly, processing the music matrix with the MC method, summarize the distribution probability, and characteristics of these data. Finally, restore the music digital symbols to MIDI music files.

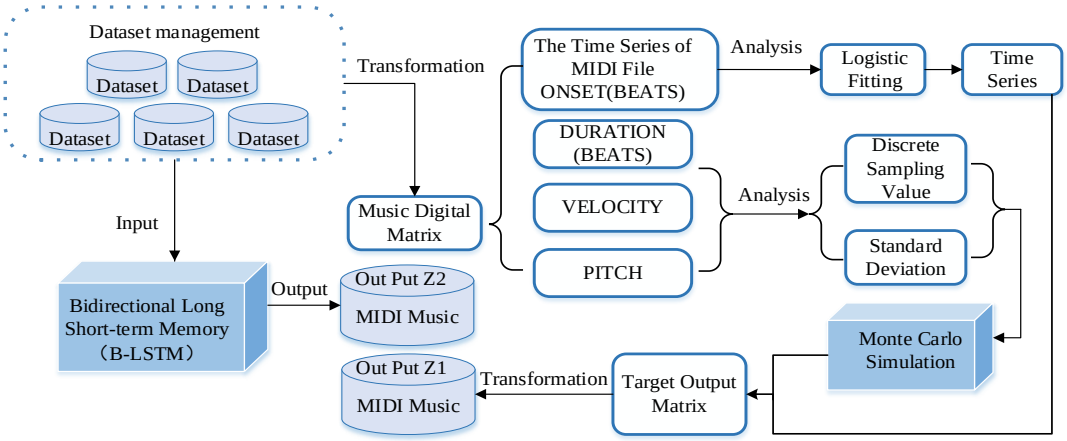


Fig. 1. The structure of the paper

From the point of view in data analysis, using the MC method can summarize the distribution law of duration, pitch and speed in music files, at the same time, by using the logistic fitting method to fit the time characteristics. After comparing the music files generated by the MC method with the music files generate by the LSTM system under the music evaluation mechanism of Euler, the final result shows the system which bases on the MC method will consume less more time and generate music have better quality. The structure of the paper is shown in **Fig. 1**.

3.1 Data Processing

By the usage of 215 world-famous piano MIDI pieces as the sample data. The MIDI music samples are processed through conversion tools. Express the duration, pitch, and velocity with the numbers at each time node, express the output information such as the notes, pitch, and beat in the form of numbers. Based on this point, it will be possible to employ the algorithm to process the music files. For instance, **Table 1** is the note matrix fragments for famous piano music <For Elise>, **Fig. 2**. is the corresponding musical notation.

Table 1. The note matrix for MIDI

ONSET	DURATION	CHANNEL	PITCH	VELOCITY	ONSET	DURATION
0	0.575	12	76	31	0.0000	0.2875
0.5166	0.500	12	75	47	0.2583	0.2500
0.9500	0.4541	12	76	58	0.4750	0.2270
1.3250	0.5375	12	75	66	0.6625	0.2687
1.7500	0.4250	12	76	57	0.8750	0.2125
2.1333	0.4791	12	71	59	1.0666	0.2395
2.9541	0.5383	12	74	55	1.2687	0.2395

In **Table 1**, the first column indicates the onset of the notes in beats (based on ticks per quarter note) and the second column indicates the duration of the notes in these same beat values. The third column indicates the MIDI channel (1-16), and the fourth indicates the MIDI pitch, where the middle C (C4) is 60. The fifth column is the velocity describing how fast the key of the note is pressed (0-127). The last two columns correspond to the first two (onset in beats, duration in beats) except that seconds are used instead of beats. The numbers of the column time (beats) and duration (beats) are twice as much as the time (sec) and duration (sec) respectively.

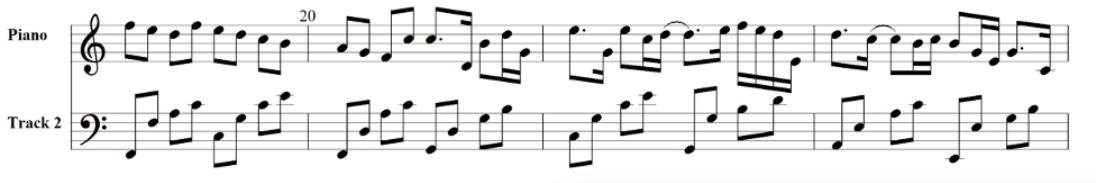


Fig. 2. The part musical notation for <For Elise>

3.2 MC Sampling Algorithm

Through sampling in random numbers, the MC sampling algorithm can predict complex trends. The basic idea is to find out the probability of the occurrence for some elements in the problem, grasp the geometric quantity and motion characteristics of data distribution, then use mathematical methods to simulate the process of generating data. Based on the probability model, the final result can be regarded as an approximate solution. Firstly, the MC sampling algorithm is used to simulate the distribution of the value of duration, pitch, and velocity in music. The rules of distribution are summarized to form a probability distribution model. Secondly, estimating the digital characteristics of the music generation model by statistical method, and calculate the optimal solution of the actual results quantitatively. Then need to calculate the discrete sampling value of the output by simulating the input distribution characteristics of the music sample model, and obtain the best estimation of the output from the discrete distribution value of the output directly. To achieve the goals stated above, a framework that can fulfill these two objectives is built:

1) Establish the measurement model of output **Y** and input *Xi* to analyze the main error sources, then establish the uncertainty evaluation model $y = f(Xi)$;
2) Determine the input (music samples) and the number of simulations (10000);
3) Generate several groups of pseudo-random numbers by computer and brought them into the music model, then obtain the output value of **N** groups of models *yr* (**r** = 1, 2, ..., **N**). Calculate the optimal output value and standard uncertainty with the **N** groups model values and the formula (1) and the formula (2);

$$\bar{y} = \frac{1}{N}\sum_{r=1}^{N} Y_r \tag{1}$$

$$u(y) = \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}\left(y_r - \bar{y}\right)^2} \tag{2}$$

4) Obtain the maximum and minimum values among the **N** groups of the model values. Divide the number between the maximum and minimum values into several intervals (equal to the numbers of the final output, 100), count the frequency of each value in each interval. At the same time, determine the inclusion interval by combining the inclusion probability **P**, so the expanded uncertainty of output **Y** also can be obtained.
The determination of sampling number **N** directly affects the reliability of evaluation results and the feasibility of simulation in the process of estimating the digital characteristics for generate the music model and calculating the optimal solution for the actual results. With the increase of sampling times **N**, the output of various index values are more accurate, but the calculation time also will be longer, until can not be achieved, to get stable statistical data, we need to increase or decrease the amount of energy artificially when implement the MC sampling algorithm. Describe this step with mathematical language is to make the standard deviation less than half of the standard uncertainty. The steps of uncertainty adjustment in the MC sampling process are as follows:
1)Assume the value n_{adj} is a suitable positive integer, the general value is 1 or 2;
2)Suppose **N** = max (**J**, 10^4), where **J** is the minimum integer which greater or equal to 100 / (1-**P**), in this paper **P** = 99 %, **N** = 10000;
3)If **h** = 1, which means this is the first time to conduct the simulation of MC;
4)When using the calculation tool to run the MC algorithm, **N** groups random numbers of the input quantity X_i in the measurement model will be generated;
5)The model values of **N** groups y_r (**r** = 1, 2, ..., **N**) were been obtained by measuring. According to formula (1) and formula (2), calculated the estimated value *y(h)* and the standard uncertainty $u(y_{(H)})$, calculate the endpoint values of the inclusion interval when the inclusion probability is *P* in the sequence;
6)Let $h = h + 1$ and restart MC simulation;
7)According to formula (3) and formula (4), calculate the average value **y** and standard deviation s_y of the estimated value of **Y**;

$$y = \frac{1}{h}\sum_{j=1}^{h} y^j \tag{3}$$

$$s_y = \sqrt{\frac{1}{h(h-1)}\sum_{j=1}^{h}\left(y^{(j)} - y\right)^2} \tag{4}$$

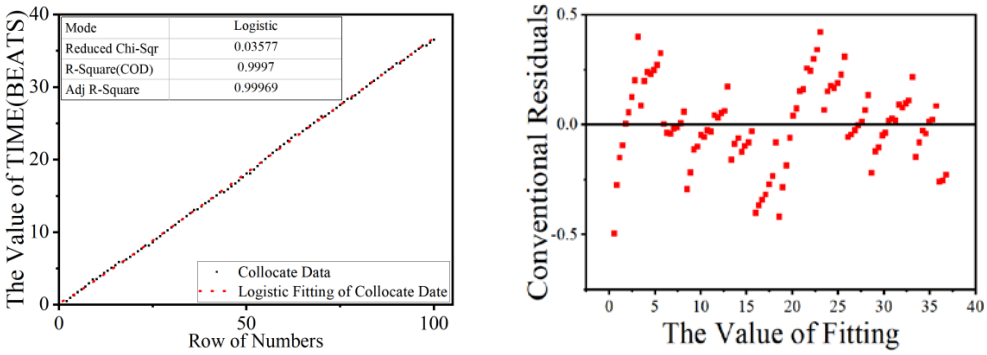
8)In the same way, calculate the standard deviations $s_{u(y)}$, s_{ylow}, s_{yhigh} corresponding to the, $u_{(y)}$, y_{low}, y_{high};
9)Calculate $u_{(y)}$ according to all models, the numbers of the models are $h \times N$;
10)Calculate the numerical tolerance δ of $u_{(y)}$: if $u(y) = c * 10^4$, where c is an n-digit decimal positive number of n_{adj}, so c=1, then $\delta = \frac{c * 10^4}{2} = \frac{10^4}{2}$;
11)If one of either s_y, $s_{u(y)}$, s_{ylow}, s_{yhigh} is greater than half of δ, let $h = h + 1$ and restart the MC simulation. Otherwise, consider all calculation conditions are stable. According to all model values, calculate the optimal value of **y**, standard uncertainty $u_{(y)}$, and inclusion interval under inclusion probability **P**.

ALGORITHM: MC SAMPLING ALGORITHM

Input: The number of rounds **h**=1, the number of MC executed per round is **M**, Significant number n_{dig}, inclusion probability **ρ**.
Output: Output estimate $y(h)$, standard uncertainty $u(y)^{(h)}$, include interval endpoints $y_{low}^{(h)}$, $y_{high}^{(h)}$
The distributed iteration at each node, for **h**=1, 2, ...
1: Calculate $y^{(r)}$, $u(y)^{(r)}$, $y_{low}^{(r)}$ and $y_{high}^{(r)}$, obtain the mean standard deviation s_y, $s_u^{(y)}$, s_{ylow} and s_{yhigh}.
2: Calculate the standard uncertainty $u(y)$.
3: Calculate the numerical tolerance δ of $u(y)$.
4: Judge s_y, $s_u^{(y)}$, s_{ylow} and $s_{yhigh} < \delta$, if not, step to 1.
End for

3.3 Experiments

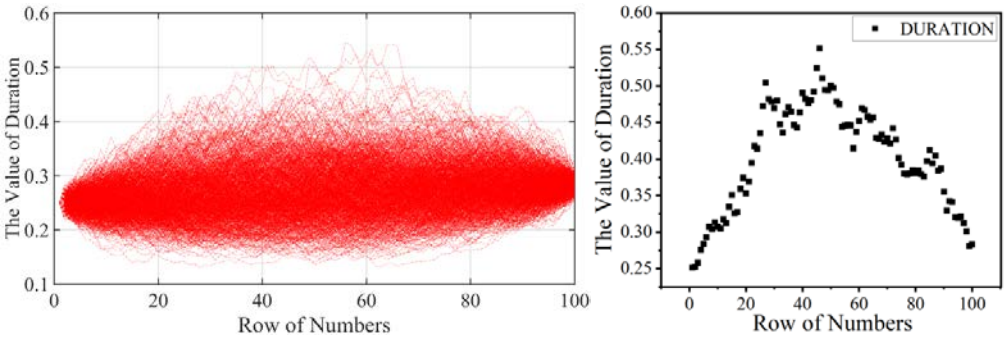
The time distribution of music samples is independent of each other, which is consistent with the characteristics for the application object of the logistic regression model. The essence of logistic regression is to divide the probability of occurrence by the probability of non-occurrence and then take logarithm, this transformation changes the value range contradiction and the curve relationship between dependent variables and independent variables. The reason is that the probability of occurrence and non-occurrence becomes a ratio, which is a buffer to expand the range of values, then the logarithmic transformation is carried out to change the dependent variable. According to an enormous number of experiments, this kind of transformation often makes the variable of dependent and independent formation a linear relationship. Therefore, logistic fitting not only solves the problem that the dependent variable is not continuous but also solves the problem that the dependent variable is continuous. **Fig. 3** shows the time data nonlinear curve fitting (logistic).
As seen from **Fig. 3 (a)** logistic regression analysis can show the distribution law of time series fantastic. Use logistic fitting the music samples, the Reduced Chi-Sqr=0.03577, square of the error be equal to COD (R^2) =0.9997, the formula for Chi-Sqr COD (R^2) are see in **Table 4** and the value of the conventional residuals fluctuate in a reasonable range. Using logistic regression analysis to fit the time series can summary the distribution law of time series data well.



(a) (b)

Fig. 3. Nonlinear curve fitting of time

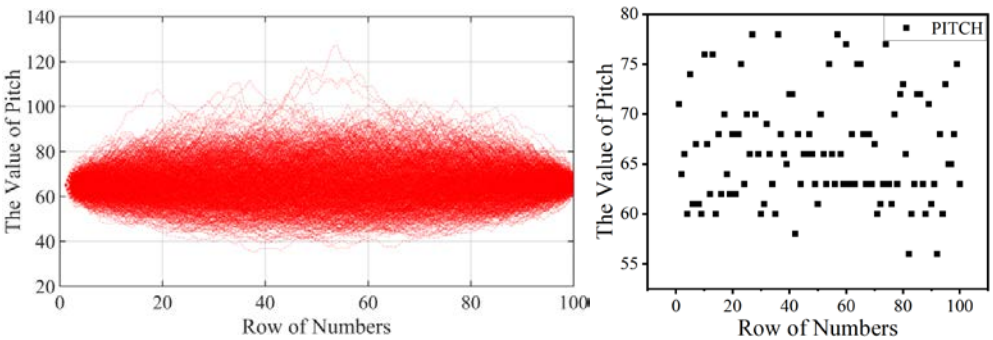
Fig. 3 (b) is the value obtained by using a logistic fitting method to fit the time series of music samples, the results show that the error between the fitting value and the original data is between -0.5 ~ 0.5. Fitting data can represent the original data well.



(a) (b)

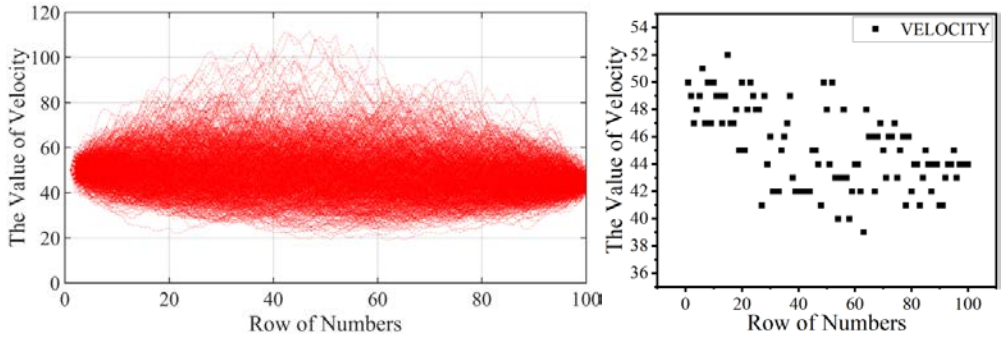
Fig. 4. Simulation value for the duration

Fig. 4 shows the value of the duration simulated by the MC method.
Fig. 4 (a) is the process of data analysis for duration applying the method of MC. During the MC simulation, the final duration value conforms to the basic law in music creation, curve changes in **Fig. 4 (b)** also correspond to the prelude, climax, and ending in the music. The duration value which fitted out has certain regularity with the advance of the time.



(a) (b)

Fig. 5. Simulation value for the pitch



(a) (b)

Fig. 6. Simulation value for the velocity

Using the same way to deal with the data of pitch and velocity, then obtain **Fig. 5** and **Fig. 6**. **Fig. 5 (a)** is the process of data analysis for pitch applying the method of MC. **Fig. 5** (b) is the final value generated by the MC algorithm, it can be seen that most of the values are concentrated around 60 (the average value).

For the value of the pitch and velocity, from the point of view of mathematical distribution, the distribution has its focus but also has a certain degree of increasing or decreasing. Finally, restore the processed data in the form of **Table 2** (select the first 30 groups data and reserve 4 decimal).

Table 2. The note matrix for the final result

R	T(SEC)	D(SEC)	P	V	T(BEATS)	D(BEATS)
1	0	0.1256	71	50	0	0.2512
2	0.2584	0.1262	64	49	0.5167	0.2523
3	0.4750	0.1289	66	47	0.9500	0.2578
4	0.6625	0.1378	60	48	1.3250	0.2755
5	0.8750	0.1417	74	49	1.750	0.2834
6	1.0667	0.1466	61	51	2.1333	0.2932
7	1.2688	0.1535	67	47	2.5375	0.3069
8	1.4771	0.1519	61	50	2.9542	0.3038
9	1.7479	0.1566	60	47	3.4958	0.3132
10	1.7647	0.1535	76	50	3.5293	0.3070
11	1.9959	0.1524	67	49	3.9917	0.3048
12	2.1917	0.1590	62	49	4.3833	0.3179
13	2.3625	0.1561	76	47	4.7250	0.3121
14	2.5500	0.1673	60	49	5.1000	0.3345
15	2.7396	0.1753	68	52	5.4792	0.3505
16	2.9459	0.1630	62	47	5.8917	0.3259
17	2.9646	0.1636	70	47	5.9292	0.3271
18	3.1250	0.1796	64	48	6.2500	0.3592
19	3.3042	0.1870	62	45	6.6083	0.3740
20	3.4979	0.1765	68	50	6.9958	0.3530
21	3.6834	0.1847	62	45	7.3667	0.3693
22	3.8750	0.1972	68	48	7.7500	0.3944
23	4.0834	0.2088	75	50	8.1667	0.4175
24	4.0917	0.2072	63	49	8.1833	0.4143
25	4.3125	0.2176	70	48	8.6250	0.4352
26	4.5500	0.2362	66	48	9.1000	0.4723

27	4.7417	0.2522	78	41	9.4833	0.5044
28	4.9521	0.2410	70	49	9.9042	0.4820
29	5.1334	0.2394	66	44	10.2667	0.4787
30	5.3334	0.2348	60	52	10.6667	0.4696

In **Table 2**, R represents a row of numbers, T (sec) represents onset (sec), D (sec) represent duration (sec), P represents pitch, V represents velocity, T (beats) represent onset (beats) and D (beats) represent duration (beats).

Next, process the note matrix into a MIDI file and displayed it in the form of musical notation. Here selected a fragment of the music file and the result is shown in **Fig. 7**.

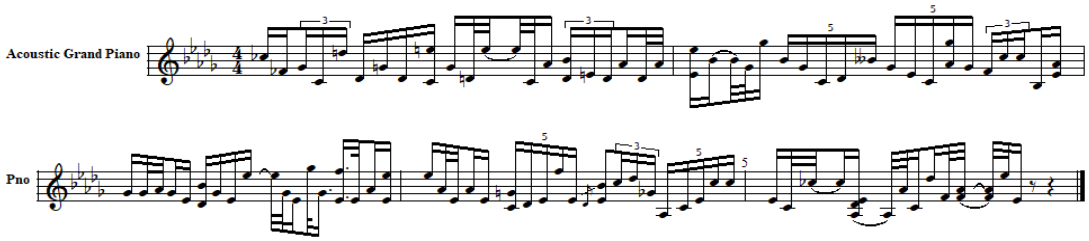


Fig. 7. The part musical notation for the final result

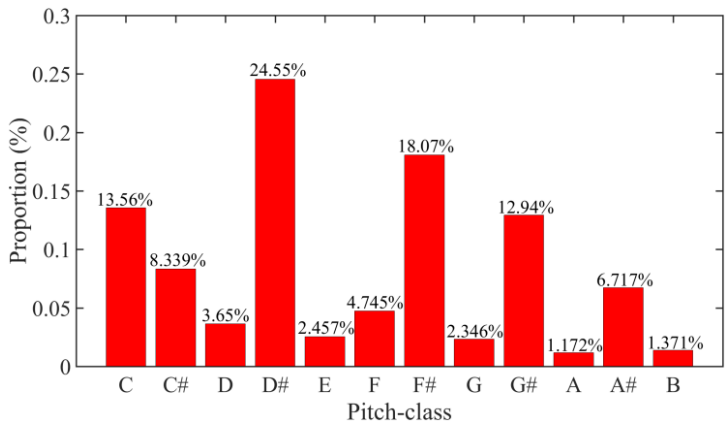


Fig. 8. Proportion of notes

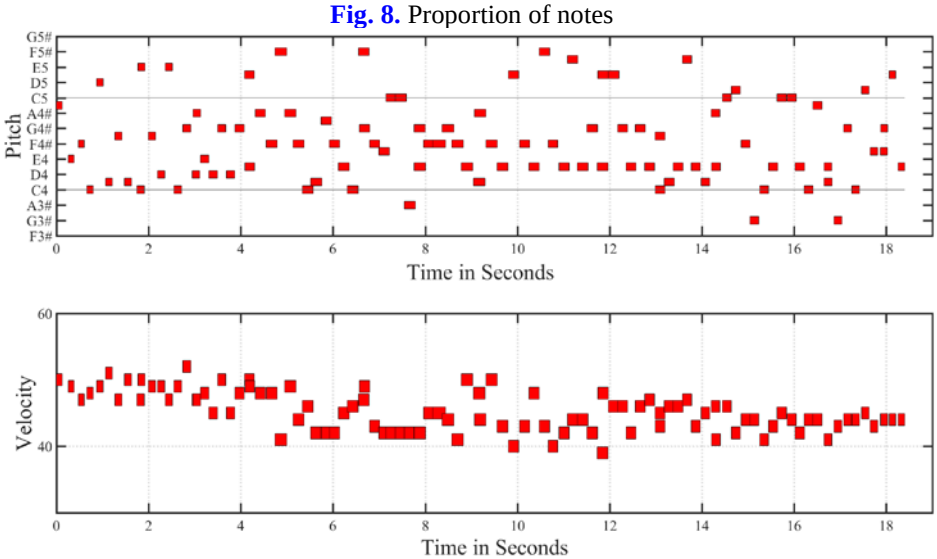


Fig. 9. The output of beat time and speed time

Fig. 7 shows the part musical notation for the final result.
Seeing from **Fig. 7** that the music generated by the MC method is tidy in note distribution. At the same time, there has a strong connection between each scale, and the generated music sound smooth.
Fig. 8 shows the proportion of each note in the generated sample. The proportion of each notes accord with the law of music generation. The final music samples not only have their main melody, but also the diverse characteristics in the distribution of notes.
Through the observation of **Fig. 9**, it can be discovered that the generated pieces capture basic harmonic relationships between the melody and accompaniment and contain consistent rhythmic patterns and easy to observe that the shift is accompanied by the desired changes in rhythm density and note density, as mentioned in **Fig. 4** to **Fig. 6**.

4. Evaluation

Using the subjective index of melody measure to evaluate the generated music samples, to obtain the objective melody evaluation index table of music, bring the scale data into the continuous chaotic model for simulation. **Table 3** is the objective melody evaluation index table of music.

Table 3. Melody evaluation index table

Evaluating Indicator	Meaning	Definition	Describe	Evaluating Indicator
V_t	Pitch change	$V_t = \frac{\eta_d}{n}$	η_d : Total number of different treble n : Total pitch	0.11
C_t	Tonality tendency	$C_t = \frac{\eta_p}{r_q}$	η_p : The number of tonic or dominant r_q : Sum of rhythm segments	0.64
I_d	Dissonance interval	$I_d = \frac{i_d}{n-1}$	i_d : The number of dissonant intervals	0.08
E_c	Contour stability of contour	$E_c = \frac{i_{id}}{n-1}$	i_{id} : Number of notes moving in the same direction	0.34
M_p	The proportion of the degrees progression	$M_p = \frac{i_{pd}}{n-1}$	i_{pd} : The number of degrees progression	1.00
S_r	The proportion of leap	$S_r = \frac{i_{dgr}}{i_{dg}}$	i_{dgr} : The number of parts that do not move backward after leap i_{dg} : Sum of leap	0.25
I_c	Melody extreme point	$I_c = \frac{1}{\eta_c}$	η_c : The number of extreme points	0.33

According to Euler's theory and the knowledge of music cognitive psychology, the complexity of melody has a certain relationship with auditory comfort. If the audiences have little mental calculation when listening to the melody, it means the melody is comfortable to enjoy. In other words, the more complex information that needs to be accepted and understood, the more terrible the audiences feel. The calculation process of melody measure are as follows:

1) Calculate the interval between every two notes in the melody;
2) For each interval i(i, is the number of intervals), define $a_i = n(i) \times n(d)$, where $n(i)/n(d)$ is the frequency ratio of the current interval, decompose the prime factor a_i.

$$a_i = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n} \tag{5}$$

Formula 5, p_j represents the j-th prime number in the sequence, k_j is the appear times for the prime number and **n** is the biggest number among the a_i.

3) The melodic metrics of the interval a_i can be defined as:

$$G(a_i) = \sum_{j=1}^{n} (k_j p_j - k_j) + 1 \tag{6}$$

4) Finally, the melodic metrics of the whole melody can be defined as:

$$G = \frac{1}{m} \sum_{i=1}^{m} G(a_i) \tag{7}$$

Using formulas (5), (6), (7) to calculate the melodic metrics of the system. **Fig. 10** is the musical notation of the output with the LSTM, in which the sample was unprocessed.

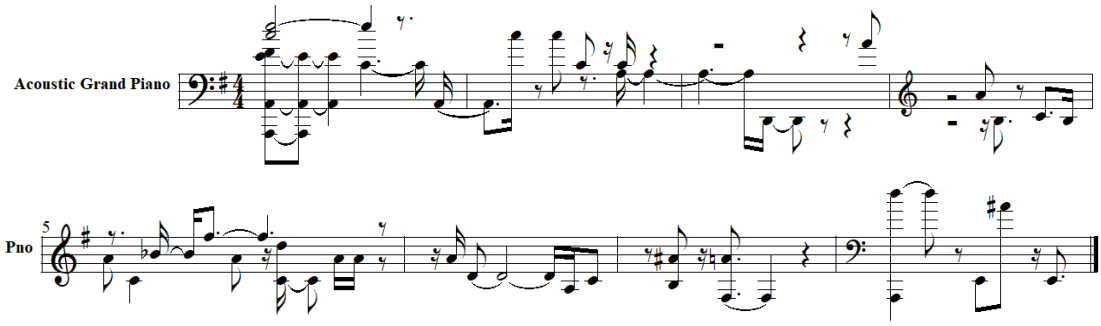


Fig. 10. The musical notation of the output with the LSTM

Seeing from **Fig. 10** that the music generated by the LSTM method is chaotic in note distribution. At the same time, there is a big gap between each scale, and the generated music sound is not very comfortable.

S. E. of the region reflects the average difference between the dependent variable and the actual value. The value is smaller, the difference between the estimated value and the actual value is smaller, and the representativeness of the estimated value is stronger. Sum squared resid is used to measure the effect of variables and random errors. Log-likelihood reflects the fitting state of the model when there is only an intercept. Generally, the value is negative. The smaller the absolute value is, the better the fitting state is. F-statistic indicates the significance of the whole fitting process. The larger the value is, the better the fitting effect is. S. D. dependent var is the arithmetic square root of the variance, which reflects the degree of dispersion of a data set. The smaller the general standard deviation is, the more stable these values are. Akaike info criterion (AIC) is an indicator to measure the goodness of the statistical model's fitting effect. The smaller the value is, the better the fitting effect is. Schwarz criterion,

Hannah Quinn criterion are similar to AIC. The smaller the value is, the better the fitting effect is. When the Durbin Watson (DW) stat is closer to 0, the correlation between the fitting data and the original data higher. The calculation process of the elements submitted above is shown in **Table 4**. It can be seen from **Table 5** and **Table 6** that the DW value of the data generated by MC theory is closer to 0.
T.

Table 4. The Elements of Regression Analysis

Variable	Calculation Formula	Describe
Chi-Sqr	$(M-x_1)^2+(M-x_2)^2+\cdots+(M-x_n)^2$	**n**: The number of free variables **k**: The number of independent variables **SSR**: Sum squared resid **RSS**: Regression sum of squares **TSS**: Total sum of squares **L**: The Likelihood function **ESS**: Explained Sum of Squares **e_t**: The residual figure **t**: The number of observations of the experiment.
S.E. of regression(**S.E**)	$\sqrt{SSR/(n-k)}$	
Sum squared resid(**R^2**)	$RSS/TSS=1-ESS/TSS$	
Log-likelihood(**L**)	$loglik=ln\,(L)$	
F-statistic(**F**)	$(ESS/k)/(RSS/(n-k-1))$	
Mean dependent var(**M**)	SUM/n	
S.D. dependent var(**S.D**)	$\sqrt{TSS/(n-1)}$	
Akaike info criterion(**AIC**)	$-2\,ln(L)+2k$	
Schwarz criterion(**BIC**)	$-2\,ln(L)+ln(n)*k$	
Hannan-Quinn criter(**HQ**)	$-2\,ln(L)+ln(ln(n))*k$	
Durbin-Watson stat(**DW**)	$\frac{\sum_{t=2}^{T}(e_t-e_{t-1})^2}{\sum_{t=1}^{T}e_t^2}$	

Table 5. Output Regression Analysis Based On MC

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.454898	3.932819	0.369938	0.7124
TIME(SEC)	4.551052	0.03431	132.6150	0.0000
DURATION(SEC)	24.17424	4.701805	5.141481	0.0000
PITCH	-0.004360	0.028233	-0.154413	0.8777
VELOCITY	-0.050238	0.068423	-0.734231	0.4649
R-squared	0.997020	Mean dependent var		44.50000
Adjusted R-squared	0.996876	S.D. dependent var		25.54734
S.E. of regression	1.427919	Akaike info criterion		3.605453
Sum squared resid	169.2330	Schwarz criterion		3.746211
Log-likelihood	-153.6399	Hannan-Quinn criter		3.662161
F-statistic	6941.406	Durbin-Watson stat		0.183682

Table 6. Output Regression Analysis Based On LSTM

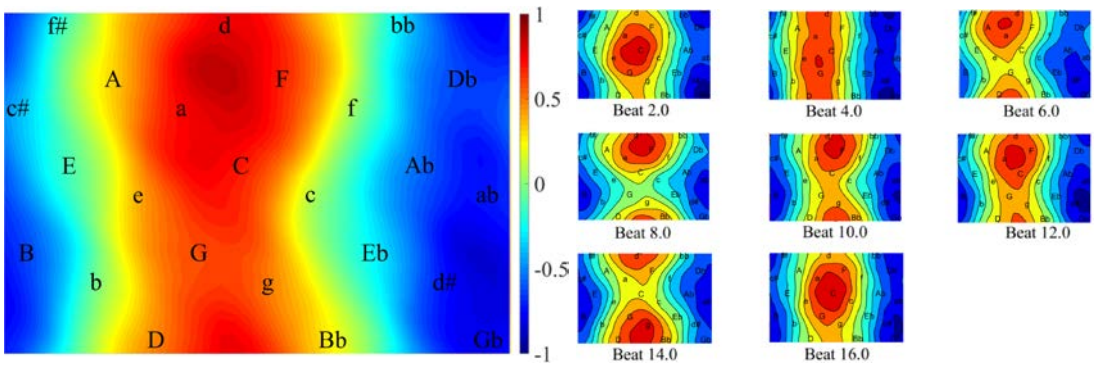
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.831544	1.536287	0.541269	0.5916
TIME(SEC)	2.454992	0.057348	42.80882	0.0000
DURATION(SEC)	0.669121	0.431023	1.552402	0.1291
PITCH	0.025576	0.022405	1.141561	0.2610
R-squared	0.982015	Mean dependent var		21.00000
Adjusted R-squared	0.980556	S.D. dependent var		11.97915
S.E. of regression	1.670375	Akaike info criterion		3.956441
Sum squared resid	103.2356	Schwarz criterion		4.123619
Log-likelihood	-77.10704	Hannan-Quinn criter		4.017318
F-statistic	673.4119	Durbin-Watson stat		0.552752

In **Table 5** and **Table 6**, S. E. of the regression reflects the average difference between the dependent variable and the actual value. Sum squared resid is used to measure the effect of variables and random errors. Log-likelihood: reflects the fitting state of the model when there is only an intercept. Generally, the value is negative. F-statistic indicates the significance of the whole fitting process. S. D. dependent var is the arithmetic square root of the variance, which reflects the degree of dispersion of a data set. Akaike info criterion (AIC) is an indicator to measure the goodness of the statistical model's fitting effect. Schwarz criterion, Hannah Quinn criter are similar to AIC. During the experiment, deleting the parameter VELOCITY in **Table 6**, due to the number of velocities generated by the LSTM system is stationary, equal to 40, and it's meaningless for regression analysis. The final fitting number of R-square for the music which is used by the way of MC is 0.997, and the corresponding number for LSTM is 0.982. Both of these two numbers largely coincide with the raw data, no matter how the output numbers are generated. After the data comparison, excepting the Log-likelihood and S.D. dependent var, that the results generated by the MC method are always better than LSTM. At the same time, excepting parameter F-statistic, for all metrics, lower the values, better the results.

Using the LSTM, under the same number of samples, the generated music sounds terrible and unstable. This is because the distribution of notes lacks too much regularity, the music sounds abrupt, and the melody is excessive without a buffer. After many attempts, the LSTM system will consume a lot of time and take up a lot of running resources. Most importantly, the output results are especially different from each other each time and do not have a unified style. Although LSTM can deal with the problem of time series well, for melodious music, the influence of melody has a greater influence than the time series.

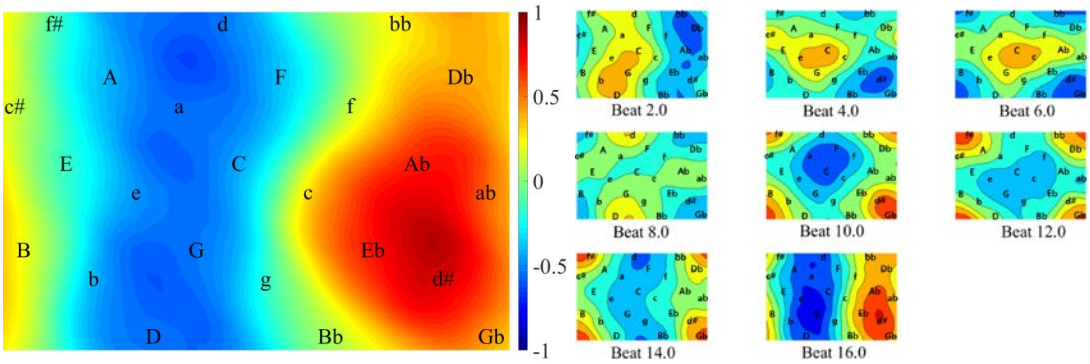
It can be seen that the melody is more natural than the result with LSTM, and the music doesn't sound too abrupt. Through the above simulation results and statistical index data, it shows that the music generated by the MC method has a more stable melody line contour and less dissonance interval. However, the melody generated by the LSTM system has more jumping intervals and larger local fluctuation of melody.

Therefore, compared with the system of the LSTM, the MC sampling algorithm can balance perfectly in the music field, which can not only deal with the time series of music samples but also can summarize the distribution of duration, pitch, and velocity.



(a) (b)

Fig. 11. Melody change of LSTM



(a) (b)

Fig. 12. Melody change of MC

Fig. 11 and **Fig.12** show the melody change diagram of the sample generated by LSTM and MC. Replaced the dynamics of the note by the depth of the color, if the dynamics are light, the color is close to blue, otherwise close to red. **Fig. 11(a)** and **Fig. 12(a)** show the melody change diagram of the sample generated by LSTM and MC. Replaced the dynamics of the note by the depth of the color, if the dynamics are light, the color is close to blue, otherwise close to red. **Fig. 11 (b)** and **Fig. 12 (b)** show the process of note change from the second beat to the 16th beat of the generated music with two beats as the interval. Here, using images to observe the melody change is an idea and suggestion, that is also an exciting way to describe the melody for the output.

5. Conclusions And Future Work

We have presented a deep generative approach to music structure analysis based on the MC algorithm. The model represents the essential characteristics of sections, homogeneity, repetitiveness, and regularity, with the MC algorithm. The experimental results show that the proposed method is effective for musical structure analysis.

The proposed method considers homogeneity, repetitiveness, and regularity, but not a novelty, which has been emphasized in conventional research. Exploiting this aspect remains an avenue for future work. It is also important to deal with further hierarchies, as music has a hierarchical structure moving from motives and phrases to sections and section groups.

References

[1] G. Hu, H. Wang, “Most Likely Optimal Subsampled Markov Chain Monte Carlo,” *Journal of Systems Science &amp*, vol. 34, no. 3, pp. 1121-1134, 2021. Article (CrossRef Link)

[2] S. Hu, X. Wang, “Exponential Convergence Rates of Markov Chains under a Weaken Minorization Condition,” *Acta Mathematica Sinica*, vol. 34, no. 12, pp. 1829-1836, 2018. Article (CrossRef Link)

[3] B. McFee, J. Kim, M. Cartwright, et al, “Open-source practices for music signal processing research: Recommendations for transparent, sustainable, and reproducible audio research," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 128-137, 2018. Article (CrossRef Link)

[4] X. Wang, D. Ding, H. Dong, X. Zhang, “Neural-Network-Based Control for Discrete-Time Nonlinear Systems with Input Saturation Under Stochastic Communication Protocol," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 12, pp. 766-778, 2021. Article (CrossRef Link)

[5] L. Hiller, L. Isaacson, *Experimental Music: Composition With an Electronic Computer*, New York, NY, USA: McGraw-Hill, 1959. Article (CrossRef Link)
[6] C. Walder, "Modelling Symbolic Music: Beyond the Piano Roll," in *Proc. of JMLR: Workshop and Conference Proceedings*, 63, 174–189, Jun. 2016. Article (CrossRef Link)
[7] F. Ghedini, P. Francois, R. Pierre, "Creating music and texts with flow machines," *Multidisciplinary Contributions to the Science of Creative Thinking*, pp. 325-343, July. 2016. Article (CrossRef Link)
[8] V. Nasiri, A.A. Darvishsefat, R. Rafiee, A. Shirvany, M. A. Hemat, "Land use change modeling through an integrated Multi-Layer Perceptron Neural Network and Markov Chain analysis(case study: Arasbaran region, Iran)," *Journal of Forestry Research*, vol. 30, no. 3, pp. 943-957. April. 2019. Article (CrossRef Link)
[9] H. Takamori, T. Nakatsuka, S. Fukayama, M. Goto, S. Morishima, "Audio-based automatic generation of a piano reduction score by considering the musical structure," in *Proc. of International Conference on Multimedia Modeling (ICMM)*, pp. 169-181, Dec. 2019. Article (CrossRef Link)
[10] M. Müller, A. Arzt, S. Balke, M. Dorfer, G. Widmer, "Cross-modal music retrieval and applications: An overview of key methodologies," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 52-62, 2019. Article (CrossRef Link)
[11] M. Xu, Z. Wang, G. Xia, "Transferring piano performance control across environments," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 221-225, 2019. Article (CrossRef Link)
[12] AllMusic, "Allmusic-record reviews, streaming songs, genres and bands," Aug, 2020. Article (CrossRef Link)
[13] M. Müller, "An educational guide through the FMP notebooks for teaching and learning fundamentals of music processing," *Signals*, vol. 2, no. 2, pp. 245-285, 2021. Article (CrossRef Link)
[14] B. D. Smith, G. E. Garnett, "Reinforcement Learning and the Creative, Automated Music Improviser," in *Proc. of EvoMUSART 2012: Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pp. 223-234, 2012. Article (CrossRef Link)
[15] Z. Wang, X. Gus, "A framework for automated popsong melody generation with piano accompaniment arrangement," *arXiv preprint*, Dec, 2018. Article (CrossRef Link)
[16] J. Gillick, A. Roberts, J. Engel, D. Eck, D. Bamman, "Learning to groove with inverse sequence transformations," in *Proc. of the 36th International Conference on Machine Learning, ICML*, pp. 2269-2279, 2019. Article (CrossRef Link)
[17] M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, G. Widmer, "End-to-end cross-modality retrieval with cca projections and pairwise ranking loss," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 117-128, 2018. Article (CrossRef Link)
[18] M. Niu, Y. Lin, Q. Zou, "sgRNACNN: identifying sgRNA on-target activity in four crops using ensembles of convolutional neural networks," *Plant molecular biology*, vol. 105, no. 4, pp. 483-495, 2021. Article (CrossRef Link)
[19] H. Dong, W. Hsiao, L. Yang, Y. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. of Thirty-Second AAAI Conference on Artificial Intelligence, AAAI Press*, vol. 32, no. 1, pp. 34-41, 2018. Article (CrossRef Link)
[20] P. Sangkloy, J. Lu, C. Fang, F. Yu, J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. Honolulu*, HI, USA, pp. 5400-5409, 2017. Article (CrossRef Link)
[21] G. Hadjeres, F. Nielsen, "Anticipation-rnn: enforcing unary constraints in sequence generation, with application to interactive music generation," *Neural Computing and Applications*, vol. 32, pp. 995-1005, 2020. Article (CrossRef Link)
[22] K. Ganguli, S. Gulati, X. Serra, P. Rao, "Data-driven exploration of melodic structures in hindustani music," in *Proc. of The 17th International Society for Music Information Retrieval Conference*, 2016. Article (CrossRef Link)

[23] K. Lu, C. S. Foo, K. K. Teh, H. D. Tran, V. R. Chandrasekhar, “Semi-supervised audio classification with consistency-based regularization,” in *Proc. of Interspeech 2019*, pp. 3654-3658, 2019. Article (CrossRef Link)
[24] S. Hochreiter, J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, Issue 8, pp. 1735-1780, Nov, 1997. Article (CrossRef Link)
[25] L. Yang, A. Lerch, “On the evaluation of generative models in music,” *Neural Computing and Applications*, vol. 32, pp. 4773-4784, 2020. Article (CrossRef Link)
[26] A. A. S. Gunawan, A. P. Iman, D. Suhartono, “Algorithmic Music Generation using Recurrent Neural Networks,” *Atlantis Press*, vol. 13, no. 1, pp. 645-654, 2020.
[27] G. Boenn, "The Farey Sequence as a Model for Musical Rhythm and Meter", *Computational Models of Rhythm and Meter*, pp. 83-112, 2018. Article (CrossRef Link)
[28] G. Aguilera, L. Riggi, K. Miller, T. Roslin, R. Bommarco, “Organic fertilisation enhances generalist predators and suppresses aphid growth in the absence of specialist predators,” *Journal of Applied Ecology*, vol. 58, no. 7, pp. 1455-1465, 2021. Article (CrossRef Link)
[29] R. F. Voss, J. Clarke, “1/*f* noise in music and speech,” *Nature*, vol. 258, pp. 317-318, 1975. Article (CrossRef Link)
[30] P. Chordia, “Modeling melodic improvisation in Turkish folk music using variable-length Markov models,” in *Proc. of Int. Soc. Music Inf. Retrieval Conf*, pp. 269-274, 2011. Article (CrossRef Link)
[31] P. Anthony, L. Robin, W. Alistair, S. Robert, “Algorithmic music as intelligent game music,” in *Proc. of AISB Anniversary Conv*, pp. 1-4, Apr, 2014. Article (CrossRef Link)

Jun Min, born in Tai Zhou, Jiang Su Province, Chian in 1993. He received the M.S. degrees in engineering from the Shanghai Institute of Technology, Shanghai, China. He is currently pursuing the Eng.D. degree in control engineering at TongJi University, Shanghai, China. His research interests include audio source separation, signal processing, and machine learning.

Lei Wang, born 1970. He received the M.S. degrees in engineering from the Tongji University, Shanghai, China. He received the Ph.D. degree in engineering from Tongji University, Shanghai, China, in 1998. This author is a Member of IEEE. His research interest includes intelligent computing, audio source separation, signal processing, and machine learning.

Junwei Pang, received the B.S. degree in mechanical engineering from Donghua University, Shanghai, China, in 2018. He is currently pursuing the M.S. degree in control engineering with Tongji University, Shanghai, China. His research interests include evolutionary computation, multi objective optimization, and its application in real word.

Huihui HAN, received the M.S degree in control theory and control engineering with the Hefei University of Technology, Hefei. in 2020. Her research interests were deep learning, pattern recognition, computer vision, and semantic segmentation. She is currently pursuing the Ph.D. degree in control science and engineering (the direction is systems engineering) at Tongji University, Shanghai, China. From 2020 until now, she is working with the team of CIMS Research Center in Tongji University, and doing the research of ternary data fusion and knowledge graph.

Dongyang Li, received the M.S. degree in school of electronics and information engineering, Tongji University, Shanghai, China, in 2017. He is currently pursuing the Ph.D. degree in school of electronics and information, Tongji University, Shanghai, China; His research interest includes intelligent computation and its application.

Maoqing Zhang, Ph. D. candidate at Tongji University. His research interest covers evolutionary computation, deep learning and their applications. Currently, He has published 8 papers indexed by SCI and 5 conference papers.

Yantai Huang, he is current research interests include computational intelligence, intelligent automation, approximate dynamic programming, and optimization and scheduling of home energy management system.