2021 13th International Conference on Wireless Communications and Signal Processing (WCSP) | 978-1-6654-0785-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/WCSP52459.2021.9613311

Novel LSTM-GAN Based Music Generation

Guangwei Li
School of Artificial Intelligence
Guilin University of
Electronic Technology
Guilin, China, 541000

Shuxue Ding
School of Artificial Intelligence
Guilin University of
Electronic Technology
Guilin, China, 541000
Email: sding@guet.edu.cn

Yujie Li
School of Artificial Intelligence
Guilin University of
Electronic Technology
Guilin, China, 541000
Email: yujieli@guet.edu.cn

***Abstract*—With the rapid development of deep learning, many models for music generation have emerged. There are, however, many problems for methods based on the general neural network model, such as slow calculation speed, complex calculation, and long-term dependence. This study proposes a combined model method for music generation, in which the long short-term memory (LSTM) neural network and generative adversarial network (GAN) are combined to form an LSTM-GAN model. In this paper, a new data preprocessing conversion rule is proposed to process the musical instrument digital interface (MIDI) message data obtained by the performance coding method. Finally, the effectiveness of the proposed model by the maximum mean discrepancy assessment is verified. Experimental results demonstrate that the proposed model can produce novel music automatically and have good performance.**

***Index Terms*—Deep neural network, long short-term memory, generative adversarial network, audio conversion.**

I. Introduction

Music generation, also known as music creation, is a process of composing or creating an original piece of music, and is a creative activity originally practiced by humans. To learn these kinds of rules and concepts such as mathematical relationships between notes, timing, and melody, the earliest study of various music computational techniques related to artificial intelligence (AI) has emerged for music composition in the middle of the 1950s [1]. In recent years, with the development of deep learning, music generation problems have returned to our vision and various related algorithm models are also applied to research music generation issues.

The first thing that appears is the application of convolutional neural networks (CNN) in music generation [2]. CNN based methods can correctly predicts and generated note sequence, which give people a lot of confidence in the music generation. Because music sequences are highly correlated, the quality of generated music is gradually improving with the continuous development of deep learning. In particular, Hong et al. [3] previously proposed a hierarchical recurrent neural network (RNN) approach, in which hierarchical coding is conducted based on prior knowledge of pop music composition. The lower layer produces a melody, while the higher layer produces drumbeats and chords. However, the RNN based generated music lacks a global structure. Although a note sequence can be generated, the structural information of the song is insufficiently clear and the learned features are insufficiently comprehensive. The reason for this failure seems to be a lack of long-range structure between data, as the RNN approach can learn short-term structures only. Because of the shortcomings of RNNs, a variant known as the LSTM network [4] has been developed. LSTM overcomes the problems of RNN and is widely favored for various applications. In addition, Goodfellow et al. [5] proposed a new framework called the GAN, which estimates a generative model through the adversarial process. GAN is the most promising deep learning model for unsupervised learning of complex distributions developed in recent years. In particular, good results have been achieved for both image generation and data enhancement. Many other methods of music generation also exist, such as autoregressive discrete autoencoders (ADAs) [6], vector quantized variational autoencoding (VQ-VAE) [7], WaveNet [8].

Inspired by the considerable success of the LSTM and GAN approaches in computer vision, natural language processing, and other fields, in this study, we propose an LSTM-GAN model for music generation. In this model, the GAN generator is used to generate music and the GAN discriminator ensures that the generated melody has the same probability distribution as the real melody. Our proposed generation framework makes several significant contributions to the generative music, as detailed below:

1) We propose a new music generation LSTM-GAN model, which is formed by combination of a LSTM network and GAN. The proposed LSTM-GAN can capture contextual structure information through the long-term correlation determined by the LSTM network learning. The model uses the structure of GAN, which constantly generates sample approximation of real data to make music generation.

2) In terms of data processing, a new note-on conversion rule is proposed. We transform the note-on information of MIDI message to our specified data information for training models. This method has effected very good compared with the existing partial data processing mode, and can yield harmonious music.

II. Related Works and Preliminaries

Automatic music generation has experienced a significant development in computational techniques related to artificial intelligence and music [9]. Most traditional music generation methods were based on music knowledge representation, which is a natural approach focused on composition rules [10]. A large number of deep learning networks now can solve many problems in this area. In addition, music data are now expressed in various forms, which are based on music knowledge as well as structures.

A. Music data representation

With the in-depth study of deep learning for music generation, the representation of training data has become more complete and, hence, the obtained feature information is increasingly rich. Earlier forms of data representation utilized a one-hot vector, in which each note of the song was assigned a value of 0-1 in accordance with the piano key value. The full song was then transformed into matrix form with easy processing. Another approach to processing of MIDI datasets is known as "MIDI-like" [11], and textualizes the audio file to obtain a representation of each song. A further approach is the WaveNet method, in which the original audio signal is used for data processing and a representation of each time-step is obtained through direct reading of the audio file. A dictionary is then constructed for data processing.

The main data processing method employed in this study is similar to MIDI-like. The key difference is that a note-on conversion rule is formulated to classify identical data types, which alleviates the memory usage problem during data processing and reduces the time required for training. However, some detailed information may be lost.

B. LSTM and GAN

An LSTM cell contains three gates: input, forget, and output [12], which determine whether new input is permitted entry, whether old information is forgotten, and the output at the current time-step. At time-step t, the states of the three gates are given by the relations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{3}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ denote the input, forget, and output gate states, respectively; $\mathbf{h}_{t-1}$ is the output of the LSTM cell at the previous time-step; $\mathbf{W}$ and $\mathbf{b}$ are weights and biases, respectively; $\mathbf{x}_t$ is the LSTM cell input; and $\sigma(\cdot)$ is the sigmoid function.

Then, the current output of the cell is computed by:

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \tag{4}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \tag{5}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \tag{6}$$

where $\circ$ denotes the element-wise multiplication between vectors.

The GAN is divided into a generator and discriminator. This network is theoretically competent for any generative network task; however, it is very difficult to implement for some tasks. The GAN implementation process can be summarized as training of a generator to generate realistic samples from random noise or latent vectors, along with training of a discriminator to identify real data and the generated data. All above are trained simultaneously until the nash equilibrium is reached; i.e., there is no difference between the generated sample and the real data, and the discriminator cannot correctly distinguish the generated sample from the real data. It takes an uniform noise vector z as an input and outputs a vector $\tilde{\mathbf{x}} = G(\mathbf{z})$.

$$\min_{\mathbf{G}}, \max_{\mathbf{D}} \boldsymbol{V}(D, G) = E_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{7}$$

C. Existing music generation methods

1) Existing LSTM methods: LSTM mainly processes time series, and the sample time sequence is very important for applications such as natural language processing [13], speech recognition [14], speech processing [15], and video detection [16]. Moreover, to overcome the problems of RNNs, Eck et al. [17] used LSTM as an alternative approach to music generation. Their experiment results showed that LSTM learns data characteristics successfully and can compose novel melodies in the same style. Further, Wang et al. [18] utilized music grammar data with an LSTM network to generate jazz music. In addition, bidirectional LSTM (BiLSTM) has been used to generate polyphonic music, with the "Thought Of Lookback" as being introduced to the architecture to improve the long-term structure and to design a new emotional music generation system [19].

2) Existing GAN methods: GAN also has many music generation applications. In 2016, Mogren et al. [20] proposed a generative adversarial model that operates on continuous sequential data; the model was trained on a collection of classical music and the associated experiments confirmed the efficacy of GAN for music generation. Subsequently, many variant GAN structures of GAN appeared in the field of music generation were developed. Dong et al. [21] proposed the MuseGAN structure for generation of multi-track music, which combined GAN models were used to generate different instruments. The different tracks were combined based on the correlation between them. The VAE-GAN model was developed, in which the variational autoencoder concept was added to a GAN then music was generated through encoding [22].

III. Model and Formulation

The overall architecture of the LSTM-GAN model system is shown in Fig.1. A combined LSTM and GAN network is used, in which the GAN architecture is constructed by the LSTM network. As audio data have long-term relevance, we use this method to learn the feature information of the data, to better capture the context information of the audio sequence. This approach allows generation of music with greater fluidity and harmony. Below, the method and model adjustment strategy are described in detail.

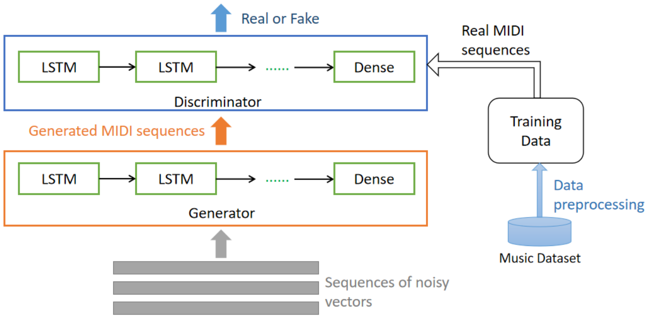


Fig. 1. Overall system framework. The LSTM network is incorporated in the GAN framework to yield a combined model. The noise vector $\boldsymbol{z}$ is the input and the output $G(\mathbf{z})$ is obtained through the generator. The discriminator then discriminates between $G(\mathbf{z})$ and real data.

A. Data Preprocessing

In this study, we used a dataset of 500 piano songs. The data set contains a large number of songs from jazz, classical and other genres, from which 400 are used for training and 100 are used for test. Hence, the feature information of the different music styles were learned and different types of music were generated.

For processing of the piano music data, we revisited some basic music theory and gained understanding of the underlying meaning and structure of various piano music elements. This knowledge played a key role in our subsequent processing of the datasets. For the data processing performed in this study, we formulated rules conforming to music theory based on our obtained knowledge. The MIDI dataset were processed via the performance coding method and, hence, the dataset were transformed into the required type [23]. The data had the following form:

128 NOTE-ON events: one for each of the 128 MIDI pitches, with each starting a new note;

128 NOTE-OFF events: one for each of the 128 MIDI pitches, with releasing a note;

VELOCITY events: each event changed the velocity applied to all subsequent notes;

TIME events: the duration of each note event.

Through the performance encoding, we could obtain the MIDI message corresponding to each note. Hence, a song performance could be expressed in text-like form, as shown in Fig.2.

The note-on conversion rules formulated according to music theory are listed in TABLE I. The white piano keys values are C, D, E, F, G, A, B, and are called natural keys. The black piano keys are called accidentals, and can have sharp or flat signs, they are represented by C#, D#, F#, G# and A#.

TABLE I
NOTE-ON CONVERSION RULES TABLE: THE NOTE-ON VALUE OF THE CORRESPONDING NOTE IS OBTAINED FROM THE TABLE AND CONVERTED TO THE GIVEN VALUE.

Note	Note-on	Note-on conversion
C	0, 12, ..., 120	0
C#	1, 13, ..., 121	1
D	2, 14, ..., 122	2
D#	3, 15, ..., 123	3
E	4, 16, ..., 124	4
F	5, 17, ..., 125	5
F#	6, 18, ..., 126	6
G	7, 19, ..., 127	7
G#	8, 20, ..., 116	8
A	9, 21, ..., 117	9
A#	10, 22, ..., 118	10
B	11, 23, ..., 119	11

In accordance with the performance coding and the effective note-on conversion rules formulated in the table, we obtained the training data following processing. The processing flow is shown in Fig.3. We converted the MIDI files in the dataset to piano-roll form and obtained the MIDI messages of the dataset through performance encoding. The required information were the note-on and time data, where the former was used to train the network model, and the latter was used for combination after the model generated a new note-on. We obtained the final training dataset by transforming the note-on information using our above note-on conversion rules.

B. LSTM-GAN model

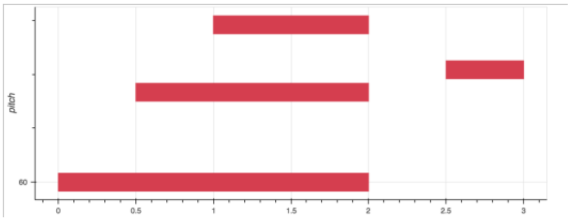
As noted above, this study proposes a combination of the LSTM network and GAN architecture to construct a combined network. That is, the LSTM network is used to construct the architecture of the GAN model to produce a new network model. As regards the LSTM, this network is beneficial for audio file processing because, compared with a CNN, it can better obtain long-term correlations between data. Music data have long-term structure and long-term correlation for a long period of time. Hence, use of the LSTM network can better train the characteristic information of the data. As regards the GAN, the samples generated by the generator can be compared with real data.

1) Generator Network: The generator network structure is detailed in TABLE II. The generator learns the distributions of real samples and is trained to reduce the discriminator error rate. The LSTM learns the notes of each melody sequence, and the LeakyReLU function acts as the network activation function. For a random input noise vector $\mathbf{z}$, music is generated through the network structure presented in the table.

TABLE II
GENERATOR NETWORK: TWO LSTM + DROPOUT LAYERS AND ONE FULLY CONNECTED LAYER. THUS, GENERATION IS PERFORMED THROUGH A THREE-LAYER NETWORK.

Generator	Networks
Layer1	LSTM+Dropout+LeakyReLU
Layer2	LSTM+Dropout+LeakyReLU
Layer3	Fully-connected+LeakyReLU

2) Discriminator Network: The discriminator network distinguishes generated samples from real data, and is trained



```
SET_VELOCITY<80>, NOTE_ON<60>
TIME_SHIFT<500>, NOTE_ON<64>
TIME_SHIFT<500>, NOTE_ON<67>
TIME_SHIFT<1000>, NOTE_OFF<60>, NOTE_OFF<64>,
NOTE_OFF<67>
TIME_SHIFT<500>, SET_VELOCITY<100>, NOTE_ON<65>
TIME_SHIFT<500>, NOTE_OFF<65>
```

Fig. 2. Example of song fragment transformation through performance coding. [24]

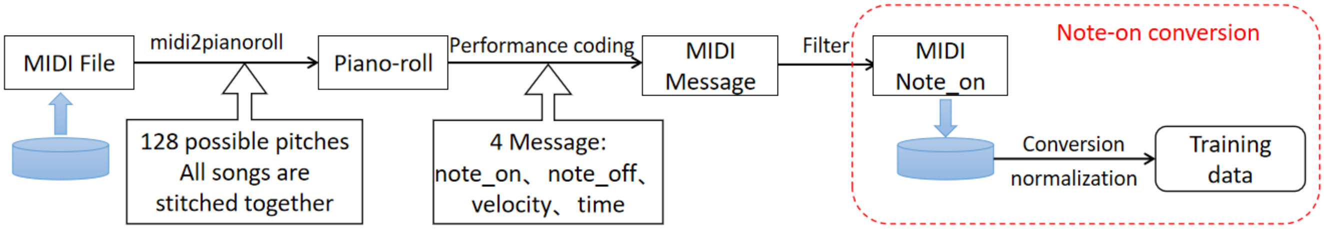


Fig. 3. Overall data processing flow chart.

by estimating the probability that the sample comes from the real training dataset instead of the generator. The generated samples are input to the discriminator for training along with the given dataset. The discriminator network structure is detailed in TABLE III.

TABLE III
DISCRIMINATOR NETWORK: TWO LSTM + DROPOUT LAYERS AND TWO FULLY CONNECTED LAYERS. NOTE THAT THE LAST LAYER OUTPUTS THE REAL OR FAKE RESULT OF THE COMPARISON BETWEEN THE GENERATED SAMPLE AND REAL DATA.

Discriminator	Networks
Layer1	LSTM+Dropout+LeakyReLU
Layer2	LSTM+Dropout+LeakyReLU
Layer3	Fully-connected+LeakyReLU
Layer4	Fully-connected

IV. EXPERIMENT AND EVALUATION

This section presents the verification method and experimental results used to verify the feasibility of the proposed LSTM-GAN model. The processed dataset was input into the network for training, which was performed with the gradient descent training method for a total of 500 epochs. The learning rate began at 0.2 and gradually decreased.

A. Maximum Mean Discrepancy

We introduced a statistical test of relative similarity, which was based on the maximum mean discrepancy (MMD) [25] between the generated sample and real data. Hence, we determined whether our generation sample gradually approached our real sample in the sense of MMD. The basic concept of MMD is that, if any order of two variables is the same, the distribution of these two variables is consistent. When the two distributions differ, the moment that maximizes the difference between the two distributions is the criterion for measuring the two distributions. After each training epoch, we used MMD to calculate the maximum mean discrepancy between the sample generated by the generator network and the corresponding real data.

Gretton et al. [26] proposed an unbiased empirical estimate and asymptotic distribution of $MMD_u^2(\mathbf{F}, \mathbf{p}_m, \mathbf{q}_n)$. Assuming two probability distributions of P_p and P_q, two sets of independent and identically distributed samples $\boldsymbol{p_m} : \{p_1, ..., p_m\}$ and $\boldsymbol{q_n} : \{q_1, ..., q_n\}$ are obtained. Then, an unbiased empirical estimate of MMD^2 is given by:

$$\begin{aligned} \mathrm{MMD}_u^2\left(\mathbf{F}, \mathbf{p}_m, \mathbf{q}_n\right) &= \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k\left(p_i, p_j\right) \\ + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k\left(q_i, q_j\right) &- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k\left(p_i, q_j\right) \end{aligned} \tag{8}$$

where $\mathbf{F}$ is a reproducing kernel hilbert space(RKHS) with the kernel function $k(p, p') = \langle \phi(p), \phi(p') \rangle$ and continuous feature mapping $\phi(p) \in \mathbf{F}$ for each p. We used $k(p, p') = \exp(-\|p - p'\|^2/(2\sigma^2))$ as the kernel function, with the kernel bandwidth σ set such that $\|p - q\|/(2\sigma^2)$ equals 1 when the distance between p and q is the mean distance between points from datasets $\mathbf{p}_m$ and $\mathbf{q}_n$ [27].

In accordance with the MMD method, we conducted verification tests on the training model. The generator output of each epoch was saved as $\tilde{\mathbf{x}} = G(\mathbf{z})$, and the real data corresponding to $\tilde{\mathbf{x}}$ in the discriminator was saved as $\mathbf{x}_{real}$. Each epoch verified the network generation effect through the MMD calculation of $\tilde{\mathbf{x}}$ and $\mathbf{x}_{real}$. The result is shown in Fig.4. In the early training stage, because the noise vector and initial value of the model parameters were inappropriate, and

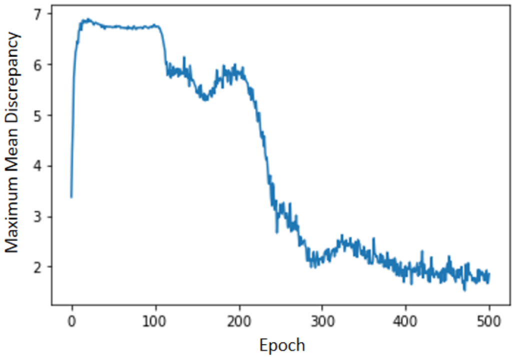


Fig. 4. Generated MMD curve between sample and real data.

as the real data of each epoch could vary, an upward trend was obtained. However, after training was performed for a sufficient volume of data, most of the data characteristics were learned. This allowed the curve to gradually decrease and to obtain a flat tendency.

B. Coherence, Musicality and Novelty

As individuals experience music differently, some evaluation methods tend to have limited efficacy for music generation studies, such as average opinion scores or similar FID-like indicators. In this study, we used surveys to manually evaluate the coherence, musicality and novelty of the generated music. Using these three subjective evaluation indicators, different music samples were compared; i.e., LSTM-GAN model generated samples, samples generated by WaveNet and LSTM, and real data. Fifty participants were randomly selected for the evaluation tests.

1) Coherence: For a song created through a conventional process, the attractive melody is coherent and there are no intermittent situations. Hence, the song coherence is an important indicator of song quality. Its mathematical expression is as follows:

$$C_n = (p_n/p_s) \times 100\% \tag{9}$$

where p_n represents the number of people who recognize the consistency of music generated by different methods, and p_s represents the total number of people who participated in the survey.

The survey results are shown in Fig.5. Most of the music generated through the network is coherent. As the LSTM network uses one-hot vector form to learn the relationship between keys for piano music training, the LSTM-generated samples had a very blunt sound. Both our proposed model and the WaveNet model generated coherent samples, such that the two exhibited roughly identical performance for this measure. However, the same rhythm was retained throughout the song in the case of the WaveNet samples. In contrast, the songs generated via the proposed approach had varying rhythms because time events could be employed.

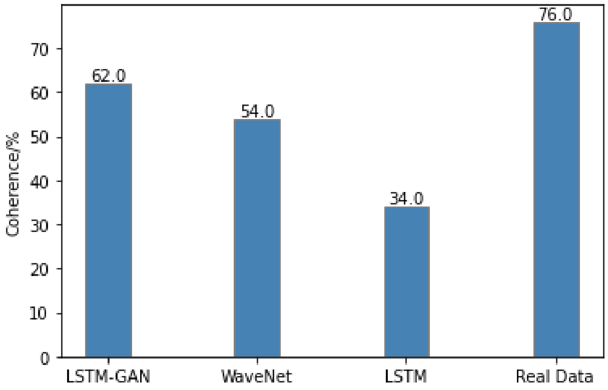


Fig. 5. Coherence analysis of four songs evaluated separately. The acceptance level of each song was also obtained.

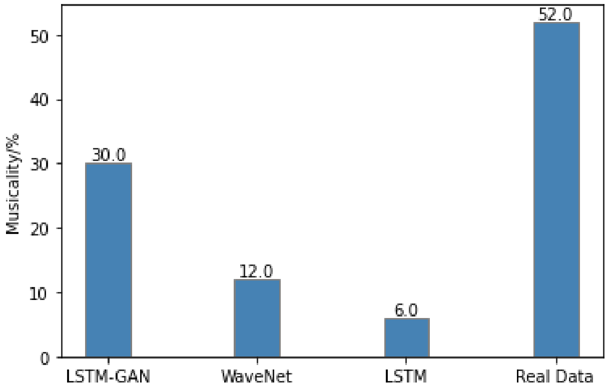


Fig. 6. Musicality analysis of four songs. Participants selected the song with the highest musicality based on their subjective impression.

2) Musicality: We must ensure the quality and effect of the samples produced by the generation network. This is not random generation, but generation and realization based on learned features. In other words, we must ensure that the generated samples have musical characteristics. Musicality is defined by the following formula:

$$M_i = (p_i/p_s) \times 100\% \tag{10}$$

where p_i represents the approved number of people who have generated music musicality by different generation methods.

As shown in Fig.6, a musical comparison of the four songs was conducted. The results indicate that all generated samples had inferior musicality to the real data. However, the music generated by our model had far better musicality than those samples generated by the WaveNet and LSTM networks.

3) Novelty: In music generation, we aim to generate music samples having a novel melody rather than a blind repetition of an existing melody. Thus, novelty is another important indicator. The mathematical expression of music novelty is as follows:

$$N_j = (p_j/p_s) \times 100\% \tag{11}$$

where p_j represents the number of people recognized for music novelty obtained by different music generation methods.

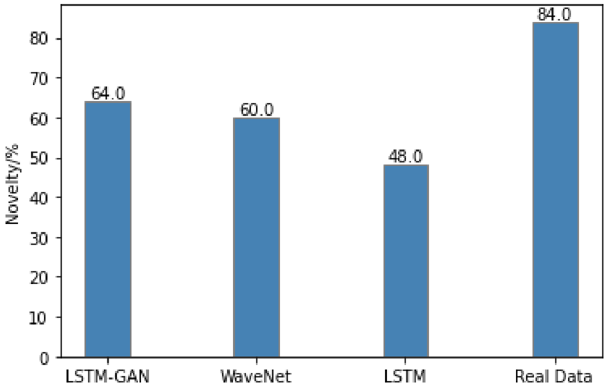


Fig. 7. Novelty analysis of four songs, evaluated separately.

As shown in Fig.7, compared with the other networks, our network also exhibited good novelty performance. Because our model was developed based on the GAN and using real data during the training process, it could learn some network features and, hence, generate novel music.

V. Conclusion

In this study, the characteristics of the LSTM network and GAN were combined to facilitate training of correlation characteristics between datasets. Hence, a combined model for music generation was obtained, and music with continuity, musicality and novelty was generated. In addition, a MIDI data processing note-on conversion rule was proposed, which yields processed data that are very useful for model training, so that the model generates music with continuity, musicality and novelty. Throughout the training process of the music generation model, the cross-entropy error was used as the loss function. We also used the MMD method to verify the effectiveness of the training process. Our proposed method can generate coherent audio samples.

The present study has allowed us to identify several directions for future work. Compared with traditional music, the structure of our generated music is not sufficiently clear because it lacks important emotional structure. Thus, it is of lower quality than many popular and celebrated songs. How to generate the combination of melody and lyrics, and how to align the lyrics and melody strictly during generation, further research is needed.

VI. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61903090 and 62076077.

References

[1] S. Schwanauer and D. Levitt, "Musical composition with a high-speed digital computer," in *Machine Models of Music*, 1993.

[2] H. Li, S. Xue, and J. Zhang, "Combining cnn and classical algorithms for music genre classification," 2018.

[3] H. Chu, R. Urtasun, and S. Fidler, "Song from pi: A musically plausible network for pop music generation," *International Conference on Learning Representations*, 2016.

[4] A. Graves, "Long short-term memory," *Springer Berlin Heidelberg*, 2012.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[6] S. Dieleman, A. Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[7] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, and I. Sutskever, "Jukebox: A generative model for music," *OpenAI*, 2020.

[8] A. Oord, S. Dieleman, H. Zen, K. Simonyan, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[9] J. D. Fernandez and F. Vico, "Ai methods in algorithmic composition: A comprehensive survey," *Journal Of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[10] T. Anders and E. R. Miranda, "Constraint programming systems for modeling music theories and composition," *ACM Comput. Surv.*, vol. 43, no. 4, 2011. [Online]. Available: https://doi.org/10.1145/1978802.1978809

[11] Y. S. Huang and Y. H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky, "Strategies for training large scale neural network language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2012.

[14] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[15] T. N. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013.

[16] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *Computer ence*, 2014.

[17] D. Eck and J. Schmidhuber, "A first look at music composition using lstm recurrent neural networks," *Idsia Usi Supsi Instituto Dalle Molle*, 2002.

[18] J. Wang, X. Wang, and J. Cai, "Jazz music generation based on grammar and lstm," in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2019.

[19] T. Jiang, Q. Xiao, and X. Yin, "Music generation using bidirectional recurrent network," in *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, 2019.

[20] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *Constructive Machine Learning Workshop at NIPS 2016*, 2016.

[21] H. W. Dong, W. Y. Hsiao, L. C. Yang, and Y. H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *32nd AAAI Conference on Artificial Intelligence*, 2017.

[22] M. Akbari and L. Jie, "Semi-recurrent cnn-based vae-gan for sequential data generation," *IEEE*, 2018.

[23] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: learning expressive musical performance," *Neural Computing & Applications*, 2018.

[24] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, "Music transformer: Generating music with long-term structure," *arXiv preprint arXiv:1809.04281*, 2018.

[25] A. Smola, A. Gretton, S. Le, and B. Schlkopf, "A hilbert space embedding for distributions," in *International Conference on Algorithmic Learning Theory*, 2007.

[26] E. T. Anders, "A survey of constraint programming systems for modelling music theories," *ACM Computing Surveys*, 2012.

[27] Y. Yu and S. Canales, "Conditional lstm-gan for melody generation from lyrics," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, pp. 1–20, 2019.