Investigation on Automatic Music Generation Using Gan and Lstm Networks

Suman Maria Tony (sumanmariatony@licet.ac.in)
Hindustan Institute of Technology and Science
S Sasikumar
Hindustan Institute of Technology and Science

Research Article

Keywords:

Posted Date: December 19th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2376478/v1

Additional Declarations: No competing interests reported.

INVESTIGATION ON AUTOMATIC MUSIC GENERATION USING GAN AND LSTM NETWORKS

Suman Maria Tony[1] and S. Sasikumar[2]

[1,2]*Department of Electronics and Communication Engineering, Hindustan Institute of Technology and Science, Chennai , india*
sumanmariatony@licet.ac.in

Abstract

In this article, the authors propose a few methodologies for composing music using deep learning algorithms and Long-short term memory (LSTM) neural network and Generative Adversarial Networks (GANs). The LSTM model is created by training with a set of input files from a music library. The trained model then synthesizes music when an arbitrary note is provided. The GAN and other variants are trained using a set of midi file accumulated from the piano dataset. The pre-trained GAN model is then used to generate music similar to piano roll. The quality of the music is calculated by comparing the harmony and few other parameters of the synthesized music with the trained files. The music library is made with a set of midi files and based on the chosen library, a unique model shall be created. For the model creation, the library files are converted into a suitable format and encoded in order to make it compatible with the LSTM network. Though the outcome of this experiment is a continuous music, the harmony and notes can still be improved to solve the discontinuity problem. The outcome of this experiment is evaluated using conventional evaluators and also the aesthetics by human observer.

1 Introduction

Music composing is one of the finest arts where the imagination of the music composer plays a major role in the outcome. Automating this music generation for a given scenario is a complex task where no scientific community had achieved results on par with the human composition.

The challenges are attributed to the following:

- Temporal dependency as time increments
- Multiple tracks and the complex interaction between such tracks or instruments

The idea is to generate a melody or a polyphony without any chords or patterns. The digitized song is converted to the musical instrument digital interface (MIDI) format for ease of processing and synthesizing music data. Automatic music generation can be done using algorithms that follow grammars and rules [1, 2], but the learning algorithms such as machine learning (ML) or deep learning (DL) can suit to generate music of varieties of genres. Various LSTM architectures for speech processing was explored which includes Deep Long Short-Term Memory (DLSTM), Long Short-Term Memory Projected (LSTMP) and Deep

Long Short-Term Memory Projected (DLSTMP) architectures [3]. An initial work on automatic music synthesis was carried out using GAN [4] and LSTM variants [5].

The inter-relationship between various tracks (Fig.1.) sometimes and independent tracks at times poses a challenge in generating music. Melody comprises musical notes in a temporal sequence which can be defined using pitch and duration. Pitch is explained as a variation of frequencies and it is played in various patterns which creates melodies [6]. Piano keys are numbered in MIDI representation from 21 to 108 denoting the pitch which can be played for a definite time representing the length and there are intervals in between[7, 8].

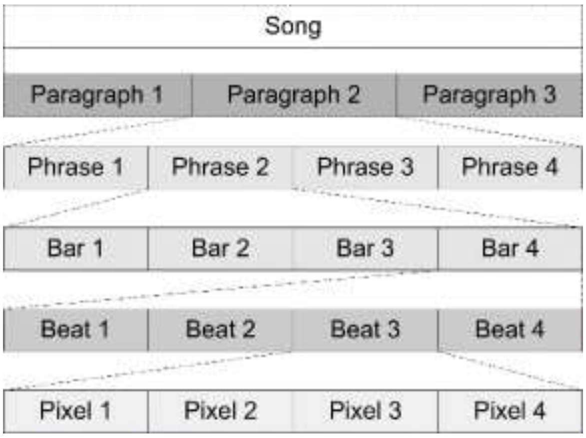


Fig. 1. Relationship between tracks

Section 1 introduces the challenges and temporal dependency in music synthesis and how AI has contributed significantly towards this. Section 2 highlights the related work in music synthesis particularly based on RNN, GAN and LSTM. Section 3 explains the architecture of LSTM and its variants. Section 4 explores several proven GAN architectures and its variants which successfully generated music files. Section 3 explains the proposed model and materials and methods used to conduct the experiment. The workflow of the conducted experiment is discussed in section 6. The simulation results and experiment outcomes are discussed in section 7. Section 8 explains the evaluation methods used to evaluate the outcome which is the synthesized music file. The concluding remarks are given in section 9.

2 Related Work in Music Generation

With the development of Artificial Neural Network (ANN) and several learning algorithms introduced using ANN, the music generating models created by such supervised learning algorithms plays a significant role in music synthesis [9] [10] [11] [12] [13] [14]. The usage of LSTM in music generation of Bach's music style was implemented using a trained neural network by considering arbitrary datasets [15]. Aiva IA, an AI-based automatic music synthesizer, has its own copyrighted audio tracks [16]. Jazz guitar music is synthesized by Sergio [17] using machine learning algorithms and also it is tuned using ornamentations to be similar to natural musical performances. Also, machine learning algorithms like the random forest was used to evaluate the parameters in music [18]. Manipulation of loudness to properly convey the emotion involved in the music was attempted using ML algorithms from the dynamic markings [19]. Also, generation of melody sequences was attempted using the genetic algorithm by modeling each sequence like a chromosome and having multi-objective optimization and fitness function [20]. Though a lot of research is on the anvil, the limitations of generating music using deep-learning algorithms still exist and the music composer has a prominent role in composing music [21].

A few networks with proven outcome are available for automatic generation of music and music accompaniment [22]. Synthesizing music tracks from each instrument has its own challenges based on the notes involved. Generating piano roll includes more active notes and a few authors have used RNN to generate piano rolls as RNN has significant learning capability [23]. Apart from the temporal learning which RNN possess, local learning is also essential and primary for the problem we have taken and hence GAN is suitable in this work on music synthesis [24]. The temporal aspects of music generation is addressed by a few researchers by implementing both RNN and GAN together [25, 26], whereas reinforcement learning along with GAN also fixes this problem [27, 28]. Since the feature extraction uses convolutional network, including the temporal features would be challenging [29, 30] and needs additional inputs for the music synthesizer to generate music with harmony.

3 Architecture of Long-Short Term Memory (LSTM) and its Variants

LSTM has been used to synthesize music and for works related to natural language processing (NLP). LSTM is a recurrent neural network that can process and remember time series data and can be used to predict the chain for a defined period.

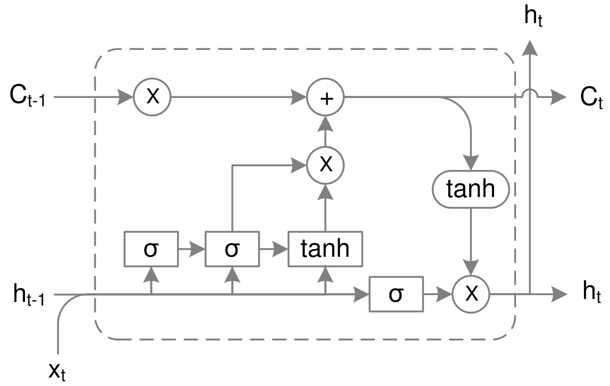


Fig. 2: LSTM Architecture

3.1 Conventional LSTM

Conventional LSTM in figure 2 consists of an input gate, a memory cell, multiplicative units, and an output gate. The input gate processes the input data and the memory cell stores the temporal data while the output gate controls the flow of output activations (equation 1). A forget gate is added to the architecture to let the unit discard the unwanted time-series data which need not be considered as part of the series.

3.2 Peephole LSTM

LSTM network with peephole connection is shown in figure 3. The mathematical equations (Equation 2) governing the Peephole LSTM and peephole convolutional LSTM is also highlighted to understand the dataflow and weight optimization. The Constant Error Carousel (CEC) avoids speedy decaying of error by adding a constant error to the units thereby addressing the vanishing gradient problem.

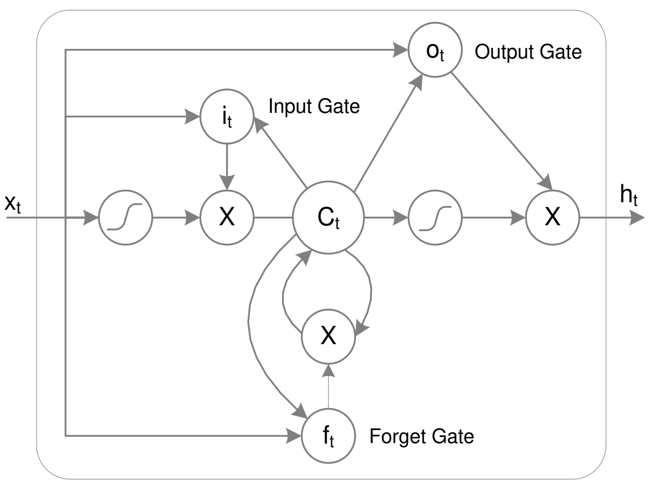


Fig. 3: Peephole LSTM Architecture

The LSTM network maps the input and output using various activation units inside the cell. The following equations are iterated for t=1 to T.

$$\begin{aligned}
f_t &= \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \\
i_t &= \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \\
o_t &= \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_t + b_o) \\
h_t &= o_t \circ \sigma_g(c_t)
\end{aligned} \tag{1}$$

(i input gate, o output gate, f forget gate, c memory cell)

Peephole convolutional LSTM is defined by the following expressions:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f C_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i C_{t-1} + b_i) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + b_c) \\
o_t &= \sigma_g(W_o x_t + U_o C_{t-1} + b_o) \\
h_t &= \sigma_h(o_t \circ c_t)
\end{aligned} \tag{2}$$

X is the element-wise multiplication and is the differentiable function

The other variants of LSTM are Deep LSTM where the architecture is deeper and it was used for some speech recognition tasks [31].

4 Generative Adversarial Networks

4.1 Overview of GAN.

The discriminator differentiates the real and the randomly synthesized signal by the generator (Fig.4.). The generator synthesizes signal close to the original music track which tries to dupe the discriminator [32.].

4.2 Existing GAN Architectures.

The three models GAN based architecture for synthesizing piano roll was proposed by Dong et al [33] and it is shown in Fig. 5. The jamming model as in Fig. **5**a has N generators and discriminators for generating N tracks. Learning is facilitated by the individual discriminators which back propagate its output. In the composer model, there in a single generator and also one discriminator to mimic a single composer creating several piano rolls. The random vector Z is shared by all the models as shown in Fig. 5b. The hybrid model uses M generator and N discriminator and the generator takes both intra-random and inter-random vectors (Fig. 5c).

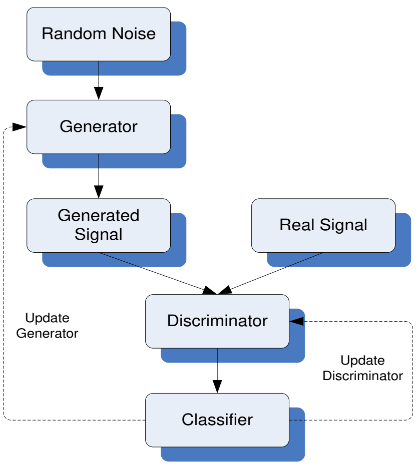
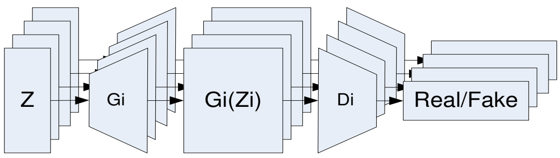
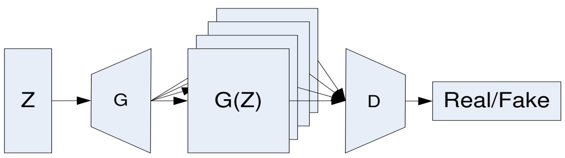


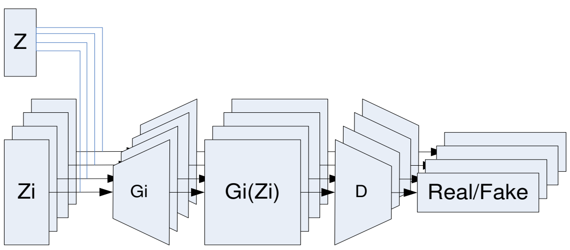
Fig. 4. General Generative Adversarial Networks



(a)



(b)



(c)

Fig. 5. Models for music synthesis (a) Jamming Model (b) Composer Model (c) Hybrid Model

5 Materials and Methods

5.1 Music Background and File Formats

MIDI format is widely used by electronic instruments and software drivers and it has Notes and Blanks. The note has a channel number, MIDI note number, velocity, etc. The main drawback of MIDI is in encoding multiple tracks and saving the intention of multiple notes which reduces the effectiveness in terms of usage. Another format that overcomes this difficulty is the piano roll which comes from the automatic piano that uses a perforated roll of papers.

5.2 LSTM Topologies

Several LSTM implementation topologies are shown in figure 6. These topologies include DLSTM, LSTMP, and DLSTMP.

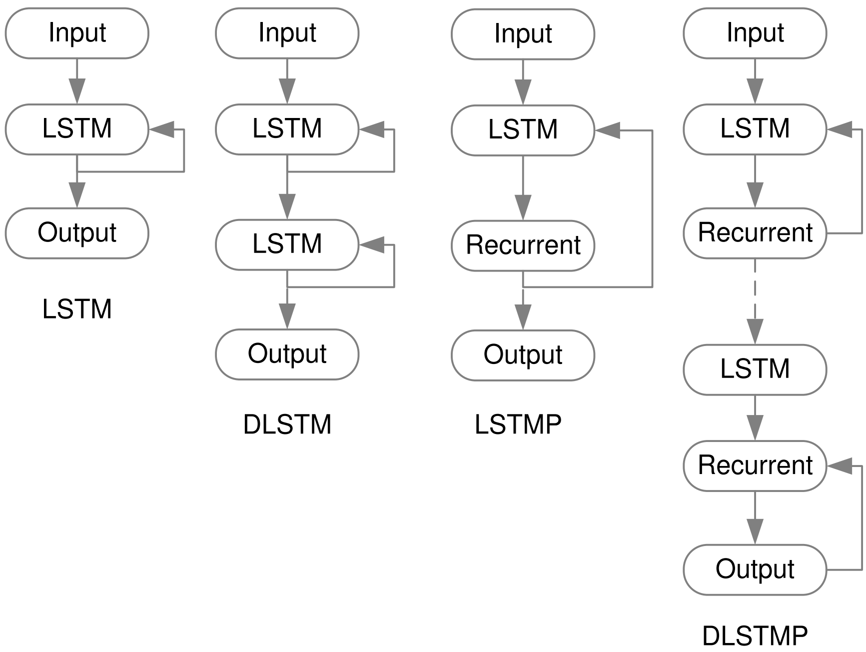
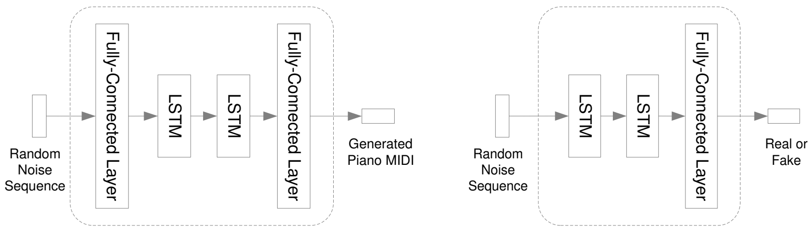


Fig. 6: LSTM architectures [2.3]

Various deep LSTM Recurrent Neural Networks (RNN) are shown in figure 6. These deep LSTM topologies which vary in depth has already been used for audio processing tasks in speech modeling [34]. These deep LSTMS are constructed by stacking LSTM RNNs and thus increasing the depth further. Thus the input travels through the stacked LSTM RNNs and facilitates learning at different times.

5.3 Conditional GAN Model

LSTM has been used to synthesize music and for works related to natural language processing (NLP). LSTM is a recurrent neural network that can process and remember time series data and can be used to predict the chain for a defined period. The generator has as many LSTM cells depending on the number of notes needed. The first layer of the generator as shown in the Fig. 7a is a rectified linear unit, ReLU. For example, if the sequence has 10 notes, the generator requires 10 LSTM cells for learning. The discriminator in Fig. 7b also has LSTM cells and it helps to differentiate the generated music sequence from the real one. The degree of closeness from the real music sequence is an important one in the learning process for which the discriminator plays a major role. The output of the hidden layers of the discriminator is given to a sigmoid function for estimating the degree of closeness to the real sequence. Thus the training happens in both the generator and the discriminator and the learning converges.



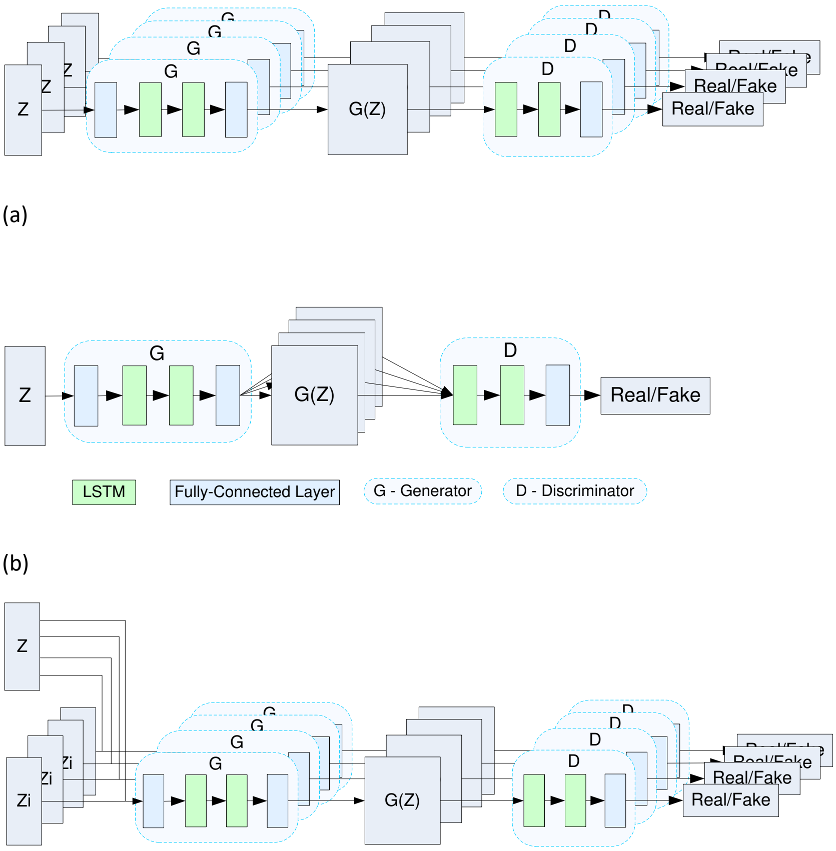
(a)

(b)

Fig. 7. GAN network with LSTM for music synthesis (a) Generator Network with LSTM (b) Discriminator Network with LSTM

5.4 Music synthesizer using Conditional GAN Models

Incorporating the conditional GAN network in the existing music synthesizer it's a good solution to preserve the temporal dependency. The Fig. 8 shows all the three models originally discussed in section 2.2.2 and it is being modified by adding the conditional GAN architecture shown in Fig. 8.



(a)

(b)

(c)

Fig. 8. Conditional GAN models for music synthesis (a) Jamming Model (b) Composer Model (c) Hybrid Model

6 Workflow

As shown in figure 9 the workflow is carried out. The MIDI file is encoded to the proper format and the LSTM network is trained with that. The model created is used to synthesize music. The midi file from the dataset is converted into song format for a more natural way of learning and synthesizing music based on the learned model. The music file is then encoded into a format suitable for the LSTM network to recognize the sequence. Here the music file is converted into a 2 dimensional matrix of samples generated by the music data. The output of LSTM after few epochs of training represents the synthesized music which is converted to a “wav” or encoded to “mp3” format to make the output compatible with a conventional audio player.

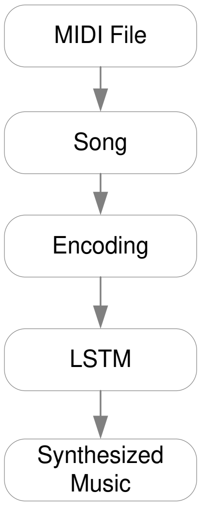


Fig. 9: Work flow of music synthesis

The MIDI file is encoded to the proper format and the conditional GAN-LSTM network is trained with that. The model created is used to synthesize piano roll particularly. The midi file from the dataset is converted into song format for a more natural way of learning and synthesizing music based on the learned model. The music file is then encoded into a format suitable for the conditional GAN-LSTM network to recognize the sequence. Here the music file is converted into a 2 dimensional matrix of samples generated by the music data. The output of music synthesizer after few epochs of training represents the synthesized music which is converted to a “wav” or encoded to “mp3” format to make the output compatible with a conventional audio player.

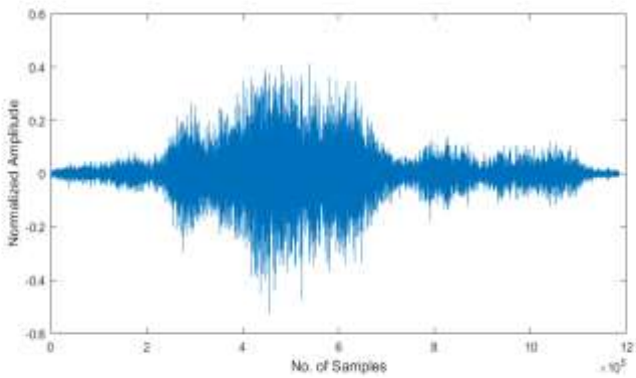
7 Experiments and Results

7.1 Data Set and Tools

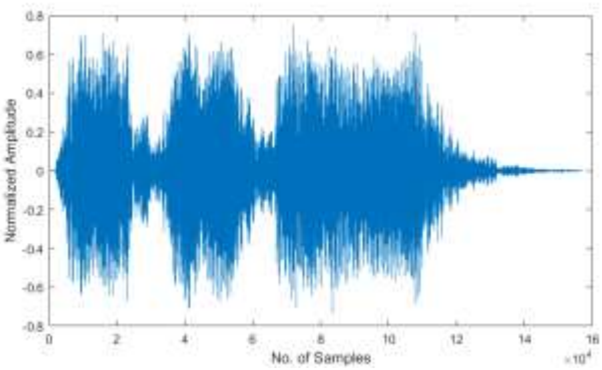
A lot of open-source music files of different genres and instruments are available on the internet. These files of different formats are converted to a single format (midi) for creating a database called “training library”. The midi music files in a particular library belong to a particular instrument like piano, drum, violin, etc. Before training the LSTM network, we have chosen one such training library as we wanted to create unique LSTM networks for each instrument.

7.2 Simulation Results

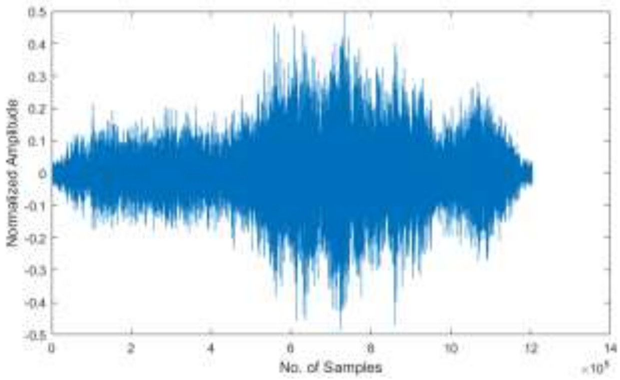
The LSTM network is trained with a set of music files collected and compiled as a separate training library discussed in the previous section. The next step in this experiment is to train the LSTM network with the music files available in the library and following the training of the network with a few epochs, the model is stored. This model is then used to synthesize music files with random initial seed and the output files obtained are compared with the music files in the training library.



(a) Synthesized music sample-1



(b) Synthesized music sample-2



(c) Synthesized music sample-3

Fig. 10: Music files synthesized by LSTM Network

At this moment quantitative analysis of the synthesized music file is not a major concern as we focused on an acceptable music file which has good harmony that is in sync with the training library. We have generated as many as music files available in the training library. Figure 10 shows some of the best music file synthesized which is on par with the music available in the library.

7.3 Generated piano rolls

Fig. 11. shows the generated piano composition automatically by the proposed algorithm using conditional GAN.

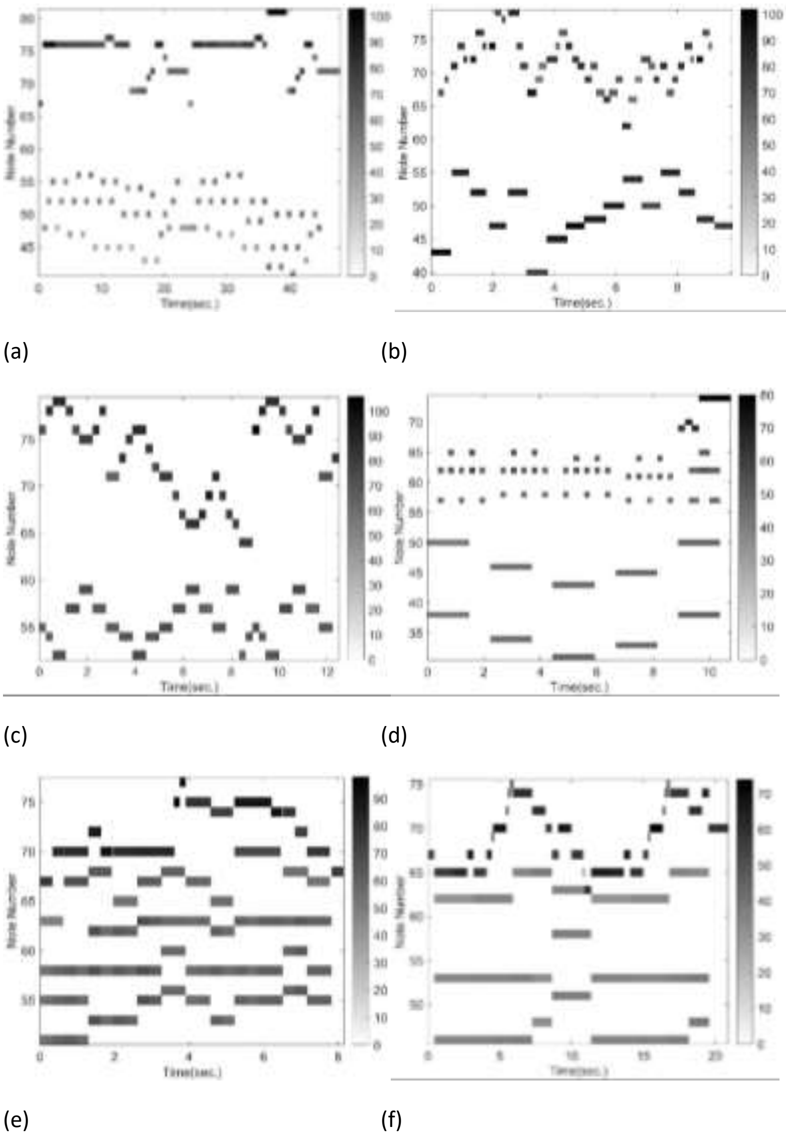


Fig. 11. Generated piano sequences

8 Evaluation of Models

Music generation models can be evaluated both subjectively and objectively (Fig.12.). The subjective evaluation methods needs the expertise of humans to assess the musical outcome of the models. In this case, the musical Turing test [35] is commonly used to compare the music composed my human and music synthesized by GAN. Whereas the objective evaluation methods use probabilistic methods, parameter metrics both for models and music generation [36, 37]. The metrics used for music generation includes the analysis of pitch, tone and rhythm. The evaluation method proposed by yang et al [38] is robust as it involves statistical parameters extracted.

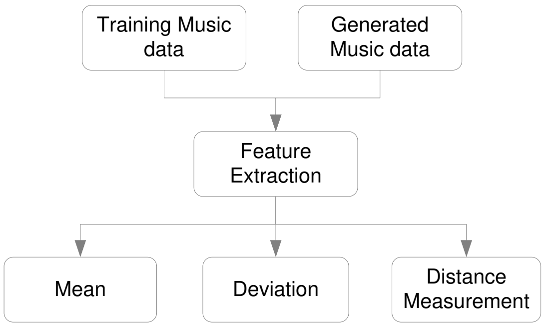


Fig. 12. Features for Evaluation of Music output

8.1 Subjective Evaluation Results.

The subjective evaluation which is used evaluate most of the automatic music synthesizer [39, 40] involves the scoring of music file generated with the help of evaluators. In this experiment, 5 male and 5 female evaluators are chosen and they scored the music files randomly chosen from the synthesized dataset. The subjects are asked to score on a 5 point scale considering the harmony of the music file. Fig.13. and Fig.14. plots the average subjective evaluation of melody score and rhythm score evaluated by various users.

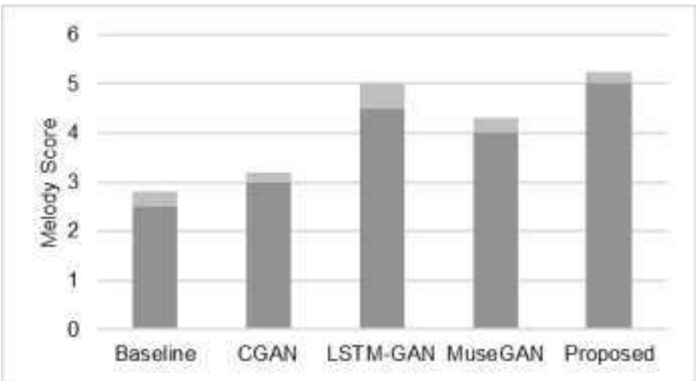


Fig. 13. Subjective evaluation: Melody Score

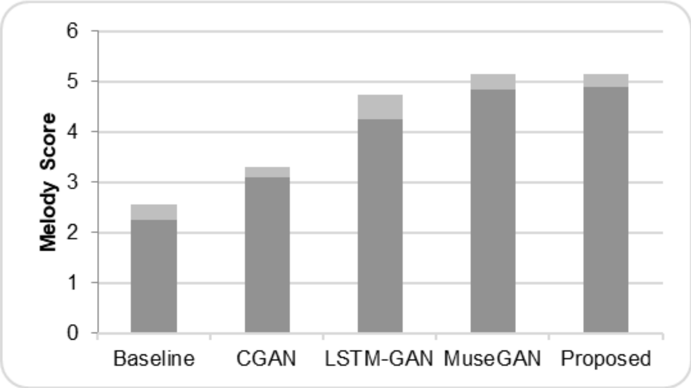


Fig. 24. Subjective evaluation: Rhythm Score

8.2 Objective Evaluation Results

The models are evaluated using pitch-based features and rhythm-based features as listed below.

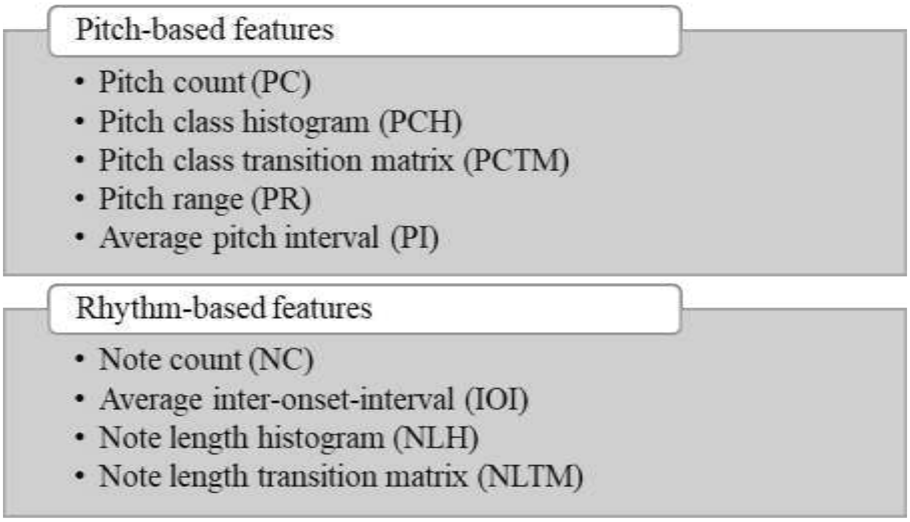


Fig. 35. Features for Objective evaluation

Based on these two sets of features mentioned in figure 15, the best two synthesized models are evaluated and the objective evaluation results are shown in figure 16. From the figure, it is evident that the pitch count, pitch count/bar, Note count, Note count/bar are better for model-2 than model-1. The Kullback–Leibler divergence (KLD) and Overlapped area (OA) of the probability distribution function (PDF) of the model-1 and model-2 are considered in this plot.

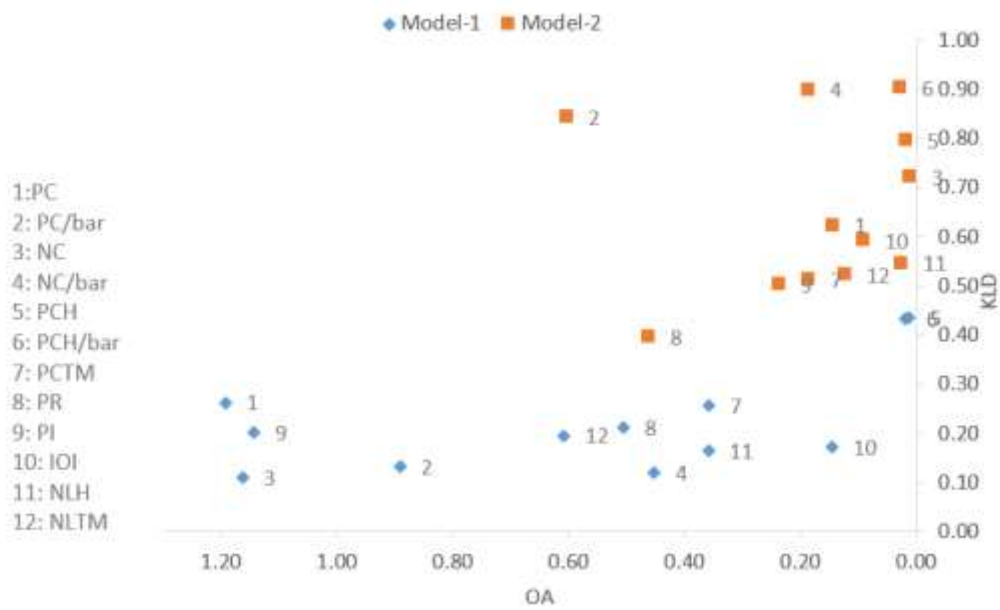


Fig. 46. Objective evaluation: Model 1 & 2

9. Conclusion

Music synthesis using AI and associated algorithms are in exploration and with the advent of GPUs, high computational complexity can he easily handled. To validate the proposed method, we have tested the algorithm with multiple tracks piano roll datasets. After ample training, the trained network synthesized piano roll that is continuous most of the times and it should be noted that the generated sequence has fragmented notes very occasionally. The evaluation of the generated music sequence shows good result which proves the effectiveness of the proposed method. The loss in temporal dependency happens in a few generated tracks and it needs to be fixed by improving the training accuracy and with a large dataset. By using the deep LSTM and GAN network, we could generate music files in a short time, but could not predict its efficiency from the conducted experiment as we focused on generating music on par with the training library. The synthesized music file is similar in harmony when compared with the training library created. Further, we are looking ahead to assess the synthesized music quantitatively by extracting statistical features and to improve the discontinuity in some music files. Also, the deep LSTM and GAN architecture can be optimized and implemented for this application thereby improving the efficiency of the training.

Acknowledgements

Not Applicable

Funding

This research work is not funded by any of the agencies

Availability of data and materials

Not applicable.

Declarations

Conflict of interest

The authors have no relevant financial or nonfinancial interest to disclose. The authors have no competing interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interest in any material discussed in this article.

Ethical Approval

This research work and the manuscript does not contain any studies with human participants or animals performed by any of the authors.

Research data policy and data availability statements

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Consent for Publication

Yes.

References

1. Kemal Ebcio˘glu.: An expert system for harmonizing four-part chorales: Computer Music Journal (CMJ), 2(3):43–51, Autumn 1988.
2. Mark Steedman.: A generative grammar for Jazz chord sequences: Music Perception, 2(1):52–77, 1984.
3. Has¸im Sak, Andrew Senior, Franc¸oise Beaufays.: Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling: Google, USA
4. Tony, S.M., Sasikumar, S. (2022). Generative Adversarial Network for Music Generation. In: Satyanarayana, C., Samanta, D., Gao, XZ., Kapoor, R.K. (eds) High Performance Computing and Networking. Lecture Notes in Electrical Engineering, vol 853. Springer, Singapore. https://doi.org/10.1007/978-981-16-9885-9_9
5. Suman Maria tony, Sasikumar, S. (2021) Music Generation Using Supervised Learning and LSTM. In: Bansal R.C., Agarwal A., Jadoun V.K. (eds) Advances in Energy Technology. Lecture Notes in Electrical Engineering, vol 766. Springer, Singapore. https://doi.org/10.1007/978-981-16-1476-7_43
6. http://www.musiccrashcourses.com/lessons/pitch.html/
7. https://en.wikipedia.org/wiki/Duration (music)/
8. https://en.wikipedia.org/wiki/Rest (music)/
9. Laden, B., Keefe, D.H.: The Representation of Pitch in a Neural Net Model for Chord Classification: Computer Music Journal, Vol. 13, No.4, 1989.
10. Chen, C. C. J., Miikkulainen, R.: Creating Melodies with Evolving Recurrent Neural Network: In Proceedings of the International Joint Conference on Neural Networks, IJCNN'01, p.2241–2246, 2001.
11. Eck, D., Schmidhuber, J.: A First Look at Music Composition using LSTM Recurrent Neural Networks: Technical Report: IDSIA-07-02, 2002.
12. Franklin, J. A.: Recurrent Neural Networks for Music Computation: Informs Journal on Computing, Vol.18, No.3, pp.321-338, 2006.
13. Mozer, M. C.: Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing: Connection Science, 6(2-3), p. 247 – 280, 1994.
14. Carpinteiro, O. A. S.: A neural model to segment musical pieces: In Proceedings of the Second Brazilian Symposium on Computer Music: Fifteenth Congress of the Brazilian Computer Society, p. 114 – 120, 2001.
15. Hewahi, N., AlSaigal, S., AlJanahi, S.: Generation of music pieces using machine learning: long short-term memory neural networks approach: Arab Journal of Basic and Applied Sciences 26(1): 397–413, 2019.
16. Travers, J.: Can I write music well? j Technology. [online]. LabRoots. Retrieved from https://www.labroots.com/trending/technology/8810/listen-ai-writing-music-decide,2018.
17. Sergio Giraldo., Rafael Ramírez, : A machine learning approach to ornamentation modeling and synthesis in jazz guitar, Journal of Mathematics and Music, 10:2, 107-126, DOI: 10.1080/17459737.2016.1207814 , 2016.
18. Phillip, B., Kirlin & Jason Yust.: Analysis of analysis: Using machine learning to evaluate the importance of music parameters for Schenkerian analysis: Journal of Mathematics and Music, 10:2, 127-148, DOI:10.1080/17459737.2016.1209588, 2016.
19. Katerina Kosta, Rafael Ramírez, Oscar, F., Bandtlow & Elaine Chew.: Mapping between dynamic markings and performed loudness: a machine learning approach: Journal of Mathematics and Music, 10:2, 149- 172, DOI: 10.1080/17459737.2016.1193237, 2016.
20. José, M., Iñesta, Darrell Conklin & Rafael Ramírez.: Machine learning and music generation: Journal of Mathematics and Music, 10:2, 87-91, DOI: 10.1080/17459737.2016.1216369, 2016.
21. L. Casini, Marfia, G., Roccetti, M.: Some Reflections on the Potential and Limitations of Deep Learning for Automated Music Generation: *IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Bologna, 2018, pp. 27-31, doi: 10.1109/PIMRC.2018.8581038, 2018.
22. Jean-Pierre Briot, Ga¨etanHadjeres, and FrancoisPachet,"Deep learning techniques for music generation: A survey", arXiv:1709.01620, 2017.
23. Hawthorne, and Douglas Eck, "A hierarchical latent vector model for learning long-term structure in music", In Proc. ICML, 2018.

24 Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck,“Counterpoint by convolution”, Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017.
25. Ian J. Goodfellow et al, “Generative adversarial nets” In Proc. NIPS, 2014.
26. Mogren, Olof, “C-RNN-GAN: Continuous recurrent neural networks with adversarial training,” Advances in Neural Information Processing Systems, 2016.
27 Yu, L., Zhang, W., Wang, J., & Yu, Y., “SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient,” The Association for the Advance of Artificial Intelligence, pp. 2852-2858, 2017.
28. Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C., & Aspuru-Guzik, A., "Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models,” in arXiv preprint arXiv:1705.10843, 2017.
29. Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H., “MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment,” The Association for the Advance of Artificial Intelligence, 2018.
30. Dong, Hao-Wen, and Yi-Hsuan Yang, “Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation,” International Society for Music Information Retrieval Conference, pp. 190-196, 2018.
31. Eyben,F., Wollmer, M., Schuller, B., Graves, A.: From speech to letters using a novel neural network architecture for grapheme based ASR: in Automatic Speech Recognition & Understanding: ASRU, IEEE Workshop on. IEEE, 2009, pp. 376–380, 2009.
32. Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. “Generative adversarial nets”, Advances in Neural Information Processing Systems 27 (NIPS 2014).
33. Dong, H., Hsiao, W., Yang, L., & Yang, Y. (2018) “MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment” AAAI.
34. F. Eyben, M. Wollmer, B. Schuller, and A. Graves, “From speech to letters using a novel neural network architecture for grapheme based ASR,” in Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on. IEEE, 2009, pp. 376–380.
35. Turing AM (1950) “Computing machinery and intelligence” Mind 59(236):433–460.
36. Sturm BL, Ben-Tal O (2017), “Taking the models back to music practice: evaluating generative transcription models built using deep learning” J Creat Music Syst.
37. Huang CZA, Cooijmans T, Roberts A, Courville A, Eck D (2017),“Counterpoint by convolution” In: International society of music information retrieval (ISMIR). Suzhou, China.
38 Li-Chia Yang, Alexander Lerch (2018), “On the evaluation of generative models in music”, Neural Computing and Applications, https://doi.org/10.1007/s00521-018-3849.
39. Agarwala N, Inoue Y, Sly A (2017),“Music composition using recurrent neural networks”, Stanford University, Technical Report in CS224.
40. Hadjeres G, Pachet F (2016),“Deepbach: a steerable model for bach chorales generation”, In: International conference on machine learning (ICML). New York City, NY, USA.