IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 7 , 2022**

Automatic Music Generation Using Machine Learning

M Ravi Kumar[1], J Lakshmi Prasanna[2], Chella Santhosh[3]

[1,2,3] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur (Dist), Andhra Pradesh, India – 522302

DOI : 10.48047/IJFANS/V11/ISS7/311

Abstract. These days, music is a huge part of everyday life. It is now included in the entertainment. One way of creating new music that has received a lot of research is automatic composition. In this work, we presented a method for automatically creating random music utilizing deep learning approaches, such as recurrent neural networks (RNNs) in Python using the Keras package and also employing the idea of Long Short-Term Memory (LSTM). RNNSdraws a series of inputs and generates another series as output. We receive the item recommendations from neural networks. This framework is intended to run this method, with musical instrument digital interface (MIDI) files serving as the input data and the created musical sequence serving as the output. This model we employ should be able to recall previous knowledge of the musical sequences. We use RNN and LSTM to recall the musical sequences since we need a system that can remember previous sequences and anticipate the next sequence.

Keywords: Musical Instrument Digital Interface, recurrent neural networks, Music generation, Machine Learning.

1. Introduction

Music representation is a sequence of events. Music can be represented in many ways like sheet music, ABC notation, MIDI (Musical Instrument Digital Interface), and mp3 formats (which are known as audio files). Out of these representations abc notation and MIDI representation are widely used and common representations. ABC notation is the simplest way of representation. sheet-music is the visual form of representation. There is a Python library called music21 which can read the MIDI files and acquires the required information. MIDI representation is the most popular form of music representation.

The sound made by any musical instrument, or the human voice is known as a musical note. A musical note is a basic musical unit. Based on its performance and quality, music as well as its notes have certain characteristics [1]. Genetic algorithms are one way to create music using already existing musical sequences [2]. According to [2], a genetic algorithm can generate a strong rhythm in every segment and integrate them to create various musical compositions. Whereas it results in low efficiency due to the delay caused in the process of each iteration. Therefore, to solve this issue, we need a system that can forecast the following musical series as well as recall the prior musical note sequence. Recurrent Neural Networks (RNNs) [3],specifically for Long Short-term Memory (LSTM) a special RNN is used.

Researchers find it fascinating to test, from the rule-based and Markov techniques of the Illiac Suite [4] to further modern deep learning systems that allow interactive piano playing tools and score filling [5].Eck and Schmidhuber were the first to use LSTMs instead of normal RNNs to create blues music in 2002[6].The network was able to produce musical compositions that had the right structure. They developed their customized format to depict musical note sequences rather than mapping this structure to MIDI files.

MIDI is the most important tool for musicians. It is a protocol that permits musical instruments and additional hardware for communication. MIDI files contain information like a set of events and the time stamp between the notes and chords. It also contains information like which note to be played. The most commonly used type of piano chord or keyboard is a three-

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 7 , 2022**

note chord (triad). A triad consists of three notes: the root note, the second and third notes, and most frequently the notes that form the third and fifth intervals are above the fundamental note.

The businesses or individuals that created the software may occasionally sell the music or artwork it produces. They may be licensed for use in commercials by businesses, etc. An illustration of this is Aiva, an artificial intelligence music generator that has published all its copyrighted albums full of produced electronic music soundtracks [7].

This article talks about an RNNs algorithm that is employed to create music and tunes automatically that depends on LSTM networks. The primary purpose of this work is to create a model that can study a sequence of musical notes, learn from them, and afterward generate a new set of notes at random. The model needs to remember the previous musical notes and understand the original sequence notes adjacent to the previous one and translate them for the learning system.

The algorithm that is described in this article is for a piano harmony musical instrument. The final output will be the music. This project uses the RNN encoder-decoder model to predict the harmony sequence (left hand) given the melody sequence (right hand). The most widely used method to create music before adopting these machine-learning approaches was generative grammar. These earlier techniques resulted in numerous subpar outcomes, such as predictable repeated melodies, and simpler musical compositions with simple melodies. The latest used approach or algorithm for generating music is NNs (Neural Networks). To create creative musical compositions, NNs can indeed be trained in music. The network understands the sequence of the music that is given as input and then uses that to generate its musical sequence. Fig. 1 and Fig. 2 represent the structure of a piano keyboard.

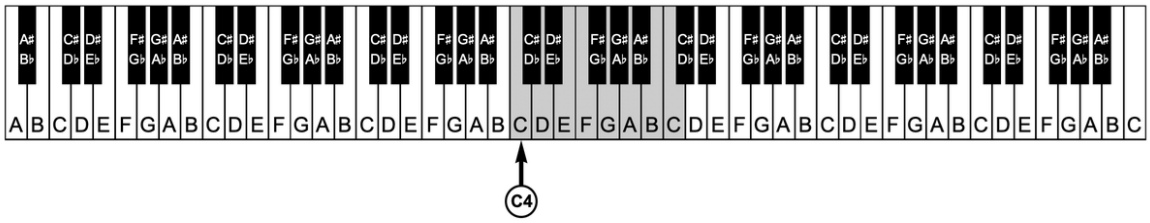


Figure 1 Piano keyboard

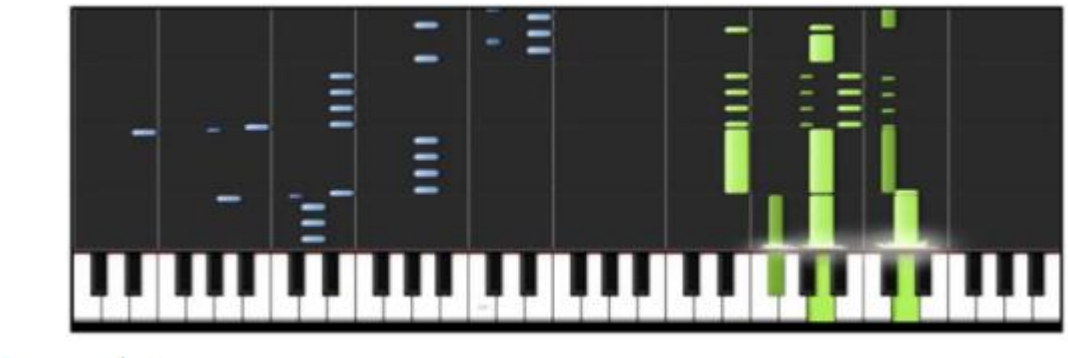


Figure 2 Piano keys

2. Literature Survey

ISSN PRINT 2319 1775 Online 2320 7876
 Volume 11, Iss 7 , 2022

The research community has spent the last few years studying music generation in detail. Numerous persons have attempted to produce music automatically by using distinct approaches. There are many open-source systems or algorithms which can generate music.

Garvit [8] presented a paper that compares the algorithms with two diverse hardware. A symphony of music can be created using a variety of methods, and by combining these methods, a new and effective model can be developed and predicted. We categorize these methods into two broad groups. One uses conventional algorithms that act on specified functions to create music, and the other uses an automated model that creates new music by learning from previous notes or sequence structures.

With the aid of a tree-based approach, Drewes [9] proposed a method for using mathematics to compose music linguistically. "Generation of Music with LSTM" [10] is a biaxial LSTM model that employs two LSTMs. Both the note to be played and the appropriate moment for the node to be played are predicted using separate LSTMs.

3. Proposed Methodology

The following steps are followed for this proposed methodology and the same was shown as architecture representation in Figure 3.

Step 1: Piano music as MIDI files is taken as input. Recurrent Neural Networks are used to predict which note or chord should be taken for the next musical sequence.

Step 2: These MIDI files taken as input are loaded. To read these MIDI files we need a package in Python i.e., music21. From this, all notes and chords can be loaded.

Step 3: Takes the path of a MIDI file and returns a list that contains 2 lists of strings. The first list is melody and the second list is harmony. These lists contain strings of pitch names.

Step 4: Takes the path string to the directory of MIDI files. Returns the input streams which are the melody list for each song and also returns the output streams which are the harmony lists for each song.

Step 5: We take the longest length of melody in the input stream and the longest length of harmony in the output stream as the output sequence.

Step 6: Get all the characters that shape our chord string.

Step 7: Prepare the data such that the network can one-hot encode the output and normalize the input.

Step 8: One hot encoding technique for the encoder and decoder input data, and decoder target to feed the model.

Step 9: Compile the encoder-decoder model and train using the given parameters. Cont is a parameter. If it is true then it will read the output file name from the weight folder and continue training that weight file instead of overwriting.

Step 10: We set up a decoder model such that it returns the internal states as well as full output sequences.

Step 11: In the proposed model, four diverse layers are used:

(i) LSTM layers is an RNN layer that accepts a sequence as an input and returns either a matrix or a sequence.

(ii) To avoid overfitting, dropout layers are a regularisation approach that involves changing a portion of the input units to 0 at each step throughout the training. The parameter employed in the layer determines the fraction.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 7 , 2022

(iii) Each input node is coupled to each output node in a layer of a connected neural network called dense layers, also known as fully connected layers.

(iv) The activation function used by our neural network to calculate a node's output is determined by the activation layer.

Step 12: The network is trained for 60 epochs; each batch is transmitted across a network enclosing 91 samples.

Step 13: This function takes an encoded melody matrix and the original melody note list. Makes a left-hand piano stream and gives the output of playing both melody and harmony and the combined part. Finally, a music sequence is generated.

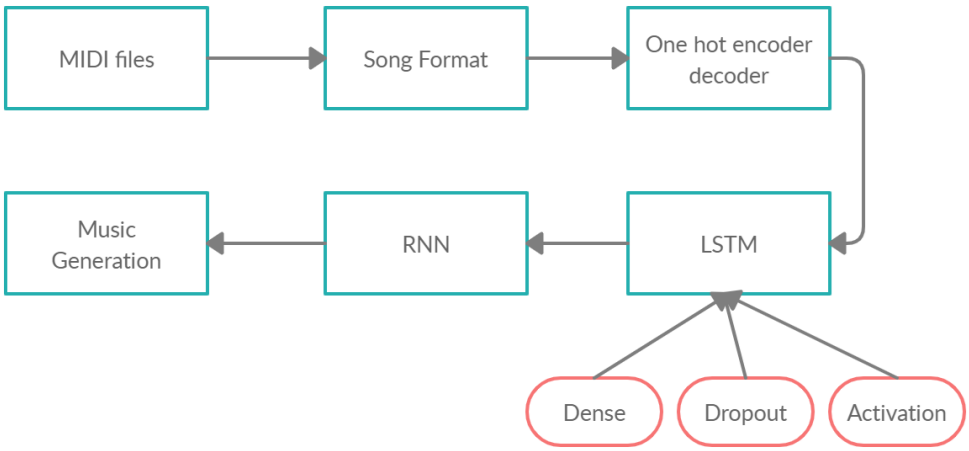


Figure 3 Proposed architecture

3.1 IMDI file format

The MIDI file stores the instructions that contain note pitches, their velocity and pressure/volume, the beginning and end of the phrases, etc. Instead of storing audio formats, it offers instructions on how to generate sound. These instructions are interpreted by the sound card that uses a wavetable to translate the information present in the MIDI file into actual sound.

It can be interpreted by MIDI-player studio software by taking some popular sequencers. Musical notation software like Muse Score translates MIDI files to editable sheet music. This eliminates the requirement for an instrument by enabling users to compose music using standard music composition on their devices and afterward listen to it using MIDI devices. Messages of MIDI are separated into the channel and system and messages.

As the notes are our primary concern, channel messages were focused initially. The number of MIDI channels is 16. 16 streams of channel messages with playable notes are included. The type of message, the note number, and the time are all included in every MIDI channel message. Other information may also be contained in the messages, but for this article, only the fundamental note messages are taken into account.

3.2 Recurrent NNs

Recurrent connections exist between previous and present states in the NN in RNNs' hidden layers. It functions as a sort of memory by storing data in the form of activations. Speech processing and music creation are made possible by this storage capacity. A regular RNN has the drawback of just storing the

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 7 , 2022**

data for the previous state only. This indicates that the context goes back just a generation. This is not particularly helpful while writing music because the beginning, middle, and end of the piece are all interconnected.

3.3 LSTM NNs

Regular RNNs' that lack long-term memory are fixed by LSTM networks. Memory cells, a distinct subset of cells, are added by LSTM networks. There are three gates in these memory cells: input, output, and forget. The amount of data that is input into the memory is controlled by input gates, the data transmitted to the next layer is monitored by output gates, and memory loss or shredding is managed by forget gates. These gates are made up of sigmoid and hyperbolic tangent functions, and because of their low computational complexity, they won't slow down during training. A memory cell's internal structure is discernible. Each gate serves a specific purpose. The central circle is utilized for stored memory, and the S circles are used for sigmoid activation functions. These memory cells have the purpose of extending the memory of a typical RNN and enabling the network to read, write, and delete stored data according to importance.

This makes it ideal for the majority of music generators and enables the networks to preserve long-term memory, which is essential in voice recognition and music composition. It facilitates the understanding of melody structure and appropriate note sequencing. The fundamental three LSTM network layout is shown in Fig. 4, where each node corresponds to an LSTM memory cell and each column to a stage of the NN. The output of one stage is mapped to the input of the following stage in this network, and the arrow connecting the memory cells denotes the stored data that is present between stages. This structure is therefore perfect for music generators that take a random note as input and anticipate an output note, which is then utilized as input for the subsequent stage to produce a sequence of notes.

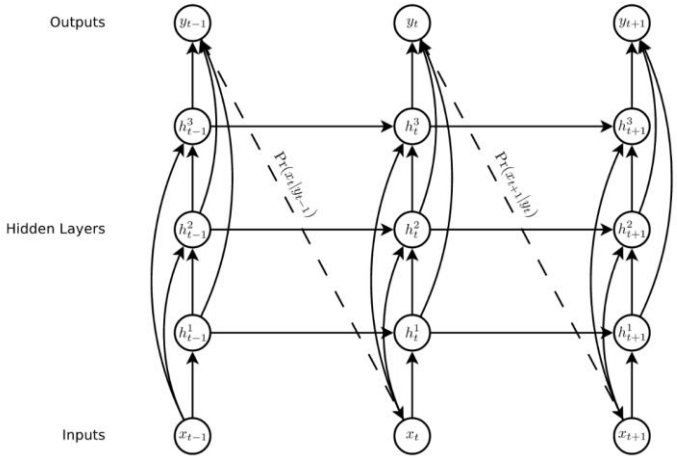


Figure 4 Three-layer LSTM network

This paper used various libraries in Python:

NumPy:
NumPy is a numerical Python library of open source. Data structures for multidimensional arrays and matrices are available in NumPy.Numerous mathematical operations, including algebraic, trigonometric, and statistical routines are carried out on arrays using this method.It is very useful in performing tasks that are more than one array. It is mainly used in statics data (mathematical operations). It is the most important library to perform statical data-related operations within time.

Pandas:

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 7 , 2022**

Pandas is a high-level data manipulation tool. It was created using the NumPy package. The Data Frame is its primary data structure. Data Frames enable the storage and manipulation of tabular data organized into columns of variables and rows of observations.

Keras:
It is made to be modular, quick, and simple to use. It is a Python-based open-source neural network library. It is a high-level API wrapper for the low-level API which can handle the way that we make our models, crucial layers, or set up multiple input-output models. At this level, our model is compiled with a loss function, optimizer functions, and a training process with a fit function. Because the "backend" engine has already taken care of low-level API tasks like creating tensors, computational graphs, and other variables, it is unable to handle them.

Matplotlib:
A plotting library for the Python programming language is called Matplotlib. NumPy is its numerical mathematics extension. It provides a general-purpose object-oriented API (Application Programming Interface) for embedding charts into applications.

4. Results and Discussion

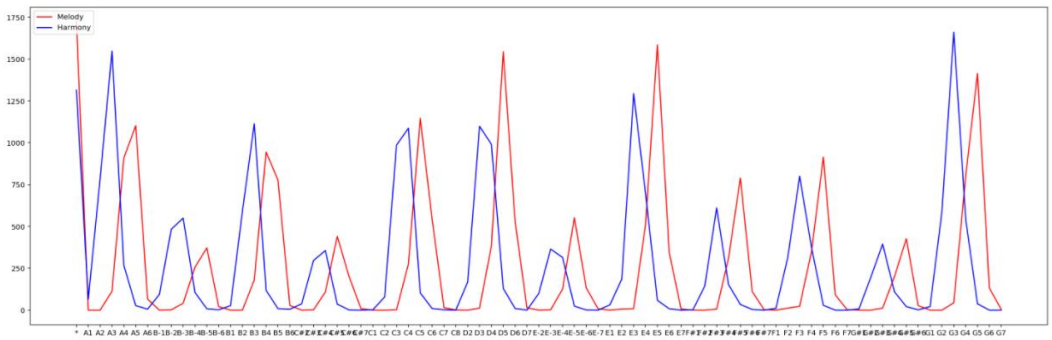
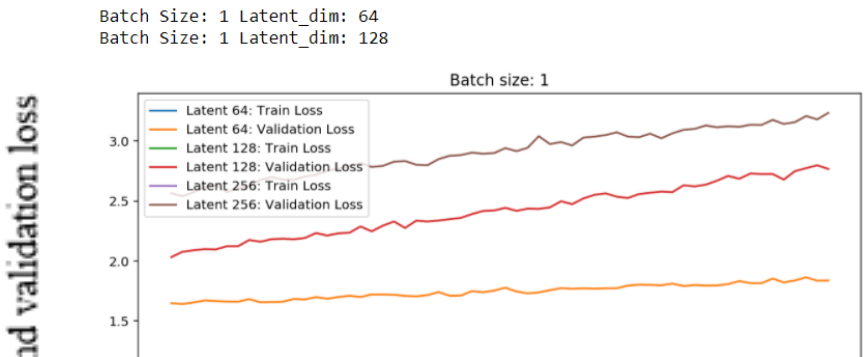


Figure 5 Keys used in all MIDI files

Figure 5 represents the melody and harmony of the MIDI files. The lines which are present in red color represent the melody sequence and the blue color lines represent the harmony sequence.



ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 7 , 2022**

Figure 6. Relation between training loss and validation loss for batch 1

Figure 6 represents the training loss and validation loss when the model is trained with batch size 1 and number of units 64,128,256 on epoch 60.

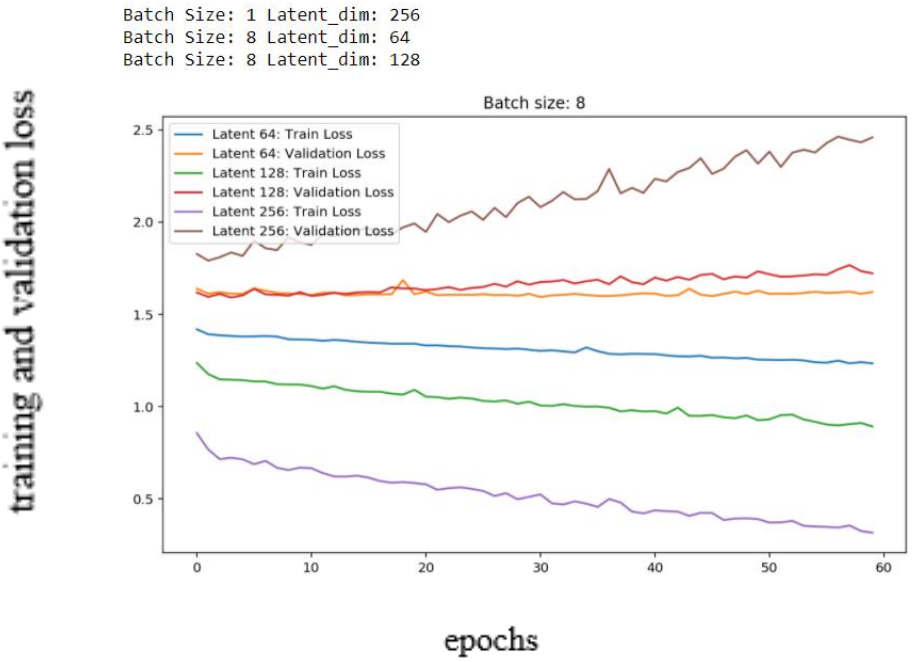
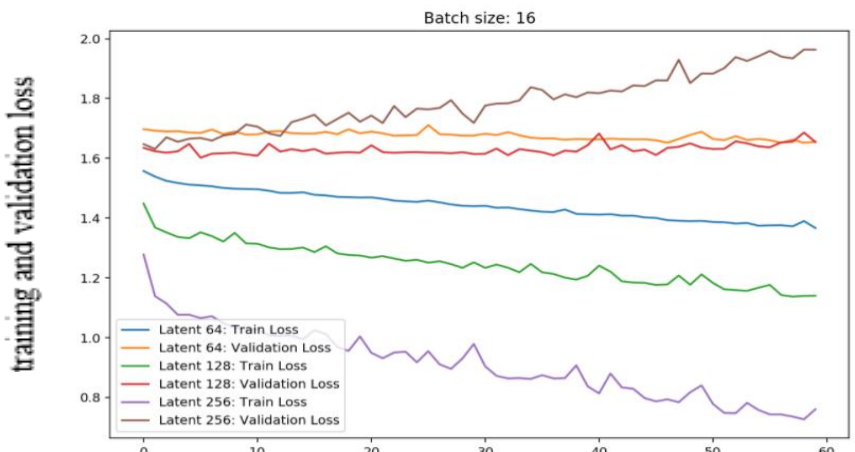


Figure 7. Relation between training loss and validation loss for batch 8

Figure 7 represents the training loss and validation loss when the model is trained with batch size 8 and several units 64,128,256 on epoch 60.



ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 7 , 2022**

Figure. 8 Relation between training loss and validation loss for batch 16

Figure 8 represents the training loss and validation loss when the model is trained with batch size 16 and several units 64,128,256 on epoch 60.

Figure. 9 Generated musical sequences

Figure 9 represents the final output where the first three audio files give the musical sequence of harmony and the next three audio files give the musical sequence of melody.

5. Conclusions

In this article, we tried to investigate the behavior of LSTM to produce music by designing an approach made up of the following steps: converting MIDI format to song format, encoding, learning with LSTM, and music production. To better understand how learning can be improved using different strategies, numerous experiments have been carried out with diverse parameters. In this work, we have shown how to produce music by using LSTM Network. Despite not being perfect the results are pretty impressive and show that NNs can generate music and can be potentially used to aid or generate more complex musical pieces.

References

1. *F. Furukawa, "Fundamentals of Music & Properties of Sound – Music Theory Lesson," 3 August 2015. [Online]. Available: https://popliteral.com/fundamentals-of-music-properties-of-sound-music-theory-lesson/. [Accessed 7 November 2018].*

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 7 , 2022**

2. *M. C. A. O. Manuel Alfonseca, "A simple genetic algorithm for music generation using algorithmic information theory," IEEE Congress on Evolutionary Computation, Singapore, pp. 3035-3042, 2007*
3. *Z. C. a. B. J. a. E. C. Lipton, "A Critical Review of Recurrent Neural Networks," arXiv preprint arXiv:1506.00019, 2015.*
4. *Hiller, L.A., Isaacson, L.M., 1979. Experimental Music;"Composition with an Electronic Computer". Greenwood Publishing Group Inc.*
5. *Donahue, C., Simon, I., Dieleman, S., 2018. "Piano genie." Preprint. arXiv:1810.05246.*
6. *D. Eck and J. Schmidhuber, "A first look at music composition using lstm recurrent neural networks," 2002. [Online]. Available: http://people.idsia.ch/~juergen/blues/IDSIA-07-02.pdf*
7. *Travers, J. (2018). Can AI write music well? j Technology. [online]. LabRoots. Retrieved from https://www.labroots.com/trending/technology/8810/listen-ai-writing-music-decide.*
8. *G. a. N. V. a. S. J. a. K. A. Joshi, "A Comparative Analysis of Algorithmic Music Generation on GPUs and FPGAs," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 229-232, 2018.*
9. *F. a. H. J. Drewes, "An algebra for tree-based music generation," Springer, pp. 172-188, 2007.*
10. *P. Y. Nikhil Kotecha, "Generating Music using an LSTM Network," arXiv.org, vol. arXiv:1804.07300, 2018.*