Automatic Music Generation Using Deep Learning

Ratika Jadhav[1], Aarati Mohite[2], Debashish Chakravarty[2], and Sanjay Nalbalwar[1](✉)

[1] Electronics and Telecommunications, Dr. Babasaheb, Ambedkar Technological University, Lonere Raigad, India
nalbalwar_sanjayan@yahoo.com

[2] Electronics and Telecommunications, Indian Institute of Technology Kharagpur, Kharagpur, India

Abstract. This paper aims to build an automatic music generation model for generating musical sequences in ABC notation using a multi-layer Long Short-Term Memory (LSTM) neural network. The model is trained on polyphony such as piano folk and old Scottish flute, merged with various ABC notation tunes by five composers, viz., Nottingham, Jack Campin, Rachael Rae, Quin Abbey, and Rabbie Burns. This approach inputs an arbitrary note from each of the five merged datasets into the neural networks. Depending on the input note, the sequence can process and enlarge until a tune of descent music is generated. With the help of hyperparameter optimization, 95% accuracy is achieved. The model's output efficiency is evaluated using frequency, autocorrelation, PSD, noise filtering, and spectrum analysis. The results show that expressive elements like duration, pitch, and harmony are essential aspects of music composition, and progress has been made to improve these parameters. In addition, the generated note frequency of music is C# (sharp), which evokes sentiments of peace and happiness in mind.

Keywords: Music Generation · Neural Network · RNN · LSTM · Deep Learning · ABC Notation · Duration · Frequency

1 Introduction

Music is the strongest form of magic and universal language, as it relieves stress and produces a kind of pleasure. Music is the time-based harmonization of sounds to create a pleasing pattern, defined as a collection of tones of different frequencies. This paper focuses on generating music automatically using the multi-layer Long Short-Term Memory (LSTM) model. To generate music, one doesn't need to be an expert in music. Even a person who does not have any knowledge of music can generate decent quality music using LSTM. We decided to train the model using ABC music notation. ABC music notation is a type of musical notation in which musical notes are symbolized by the letters from A to G [1]. The model's output data obtained is in the form of ABC music notation.

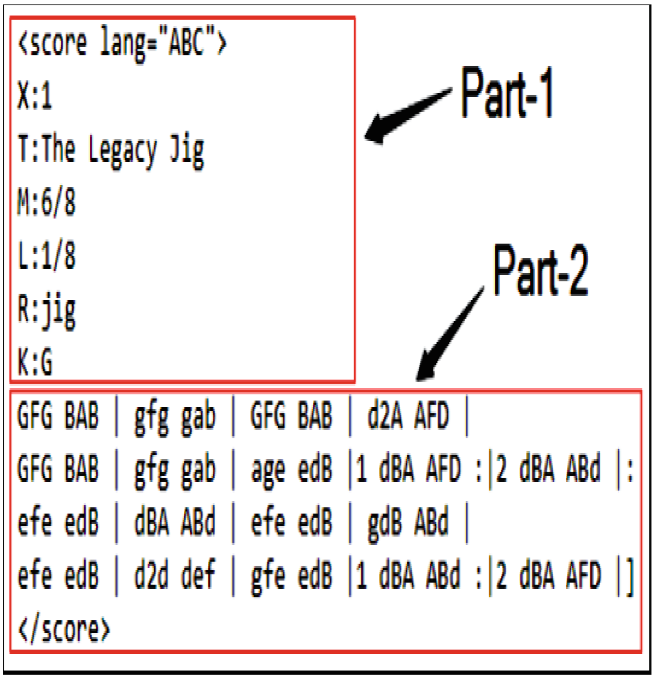
The model uses a multi-layer Long Short Term Memory Network to generate musical sequences in ABC notation. LSTM is a modified version of Recurrent Neural Networks

(RNNs) that makes it simpler to recall past information in memory. In machine learning models, the model's previous stages can't be stored. Therefore, previous stages can be stored with Recurrent Neural Networks (RNN). A repeating module in an RNN receives input from previous stages and feeds its output to the next stage. RNNs can only remember information from the most recent stage, therefore a model network that needs to learn long-term dependencies requires additional memory. It is where the LSTM Network comes into the picture [2]. A single LSTM hidden layer is followed by a typical feedforward output layer in the existing LSTM model. The multi-layer LSTM is an extension of this model that contains multiple LSTM hidden layers with numerous memory cells on each layer.

This paper consists of ABC sheet music data merged using two instruments with different ABC notation tunes of five composers and tested on abcjs quick editor. Finally, the merged data is given as input to the model for training. As a result, this model must learn the musical patterns that humans enjoy. The model should be able to generate new music for us once it has learned these patterns. It can't just be copied and pasted from the training data. It should be able to recognize musical patterns to create new music. It is not expected that the model will generate new music of professional quality. Instead, it should generate music of decent quality, melodic, longer duration, and pleasant to hear.

2 Literature Review

Researchers have been working on automatic music generation in artificial intelligence and audio synthesis domains for the past two decades. There are a variety of techniques for making music, and a combination of these approaches can be utilized to construct and design a novel but efficient model. The two primary categories in which these approaches are classified are traditional [3] and autonomous [4]. The traditional method of composing music employs algorithms that operate on predefined functions, but the autonomous model learns from preceding iterations of the notations and then creates new ones. After Artificial Intelligence, many additional models were presented [5], including Anticipation-RNN, probabilistic RNN models, and recursive artificial neural networks (RANN), an updated version of artificial neural networks for generating sequential notes and duration for rhythm composition [6]. "One technique currently being used to make musical sounds is the Generative Adversarial Networks (GANs), which consists of two neural networks: the discriminator network and the generator network. The generator and discriminator network operate together to compare the new data to the original dataset. According to the research, when it comes to fixating on specific sequences, LSTM outperforms GAN; that is, LSTM is better at discovering unique patterns and then reusing them throughout the output sequence" [7]. The LSTM-based models were capable of changing notes and breaking out specific note loops [8]. When it came to GAN-based models, they could only detect basic concepts and had shorter training times [7]. Deep Learning architectures have recently integrated two distinct approaches to automatic music generation. The first uses WaveNet architectures, while the second uses Long Short-Term Memory (LSTM) architectures. WaveNet is Google DeepMind's generative model for raw audio that is based on Deep Learning. WaveNet uses a piece of a raw audio wave as an input. A time-series Recurrent Neural Networks (RNNs)



```
<score lang="ABC">
X:1
T:The Legacy Jig
M:6/8
L:1/8
R:jig
K:G
GFG BAB | gfg gab | GFG BAB | d2A AFD |
GFG BAB | gfg gab | age edB |1 dBA AFD :|2 dBA ABd |:
efe edB | dBA ABd | efe edB | gdB ABd |
efe edB | d2d def | gfe edB |1 dBA ABd :|2 dBA AFD |]
</score>
```

Part-1
Part-2

Fig. 1. A Sample of ABC Notation of Music

variation capable of capturing long-term dependencies in the input sequence is referred to as a "raw audio wave" [9].

3 Methodology

3.1 Music Representation Used for the Model

Music can be represented in a variety of ways, including MIDI notation, ABC notation, and sheet music. In this paper, ABC notation is used to train the model for music generation because it is easy to understand and interpret using only a sequence of characters [10]. ABC notation is an ASCII musical notation format that is simple but more effective. It is divided into two parts, as shown in Fig. 1. The first part is the metadata, which includes information about a tune, such as an index, tune type, time signature, and default notes length. The second part represents the real tune, which is expressed as a string of characters. The developed algorithm learns polyphonic musical note sequences using a multi-layered LSTM network.

3.2 Music Dataset and Data Processing

In this paper, the model is trained on polyphony ABC notation, which means that the dataset consists of two instruments, such as piano folks and old Scottish flute, and the dataset is merged with various ABC notation tunes by five composers, namely Nottingham, Jack Campin, Rachael Rae, Quin Abbey, and Rabbie Burns. Tunes selected from this composer's database are Jigs, Reels, Hornpipes, Waltzes, and other dance tunes with bass, Jigs, and Slip Jigs. The model is designed to generate musical notes using integer encoding from a dictionary of 89 unique characters. The dataset is one hot-encoded that is used to categorize the data. The LSTM has been trained using batches of data. The batch specifications used here are as follows: The batch size is 16, and the sequence length is 64. The merged dataset contains a total of 133371 characters.

3.3 Design and Architecture

LSTM is a type of RNN that overcomes the problem or difficulties of training RNNs on sequential data [11]. LSTM learns long-term dependencies as our music has long-term structural patterns. The LSTM unit is composed of a cell and three gates: an input gate, a forget gate and an output gate. The cell retains values for an arbitrary period of time, and these gates control how information in a data sequence enters, is stored in, and exits the network. The forget gate in an LSTM network cell determines whether we should keep or discard information from the previous timestamp. The input gate decides which input value will be used to update the memory, while the output gate sends updated information from the current timestamp to the next timestamp, with the input and memory blocks deciding the output. In this way, LSTM operates sequentially (Fig. 2).

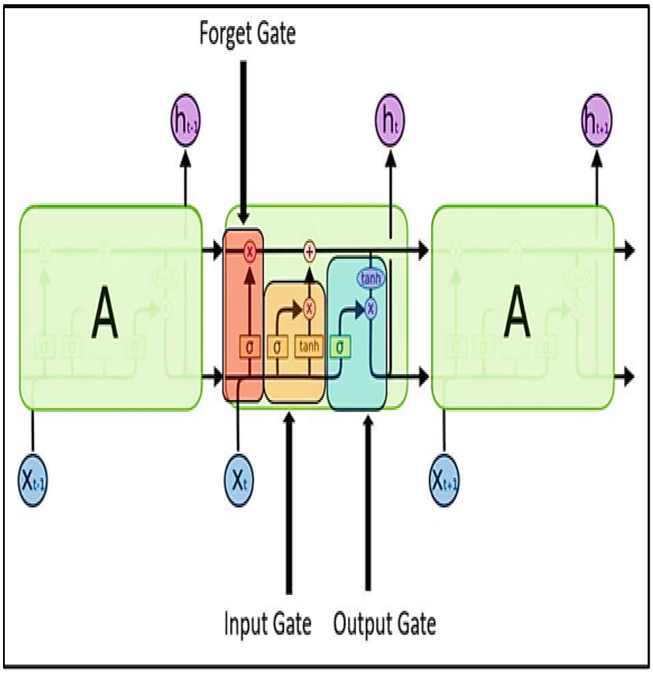


Fig. 2. Architecture of Long Short-Term Memory

The model is built on multi-layer LSTM acting as a core part, along with the dropout layer and time-distributed dense layer. To reduce overfitting, a dropout layer is used. A time-distributed dense layer is used to process the outputs at each timestep. The SoftMax classifier is also used due to the problem statement's multi-class classifying nature. To find predictions for multi-label outputs, the SoftMax classifier is used. SoftMax classifiers use cross-entropy loss to squeeze raw class scores into normalized positive numbers that add up to one to apply cross-entropy loss. To optimize the model, an Adaptive Moment Estimation optimizer is used, also called Adam [12]. It's an excellent choice for LSTM since the model deals with LSTM as input data is fed in sequence (Figs. 3 and 4).

3.4 Music Generation

After training the LSTM network, the model is ready to generate a new pattern of musical notes. The input of the model consists of 89 unique characters, which will generate 89 probability values using the SoftMax classifier. The model then probabilistically selects the next character from the returned 89 probability values and gives back the selected character's input to the model. The model generates music by combining the output characters. In this way, music will be generated.

4 Results and Discussion

The model is implemented over 150 epochs, which results in less training loss per epoch as the number of epochs increases, resulting in a training accuracy of 95%. Figure 5 depicts plotly outputs (visualizations) in the form of graphs representing epochs vs. accuracy. It describes the model's training accuracy values with corresponding epochs. The training accuracy level is observed to be 73% at 20 epochs and then increases linearly from 40 epochs on, that is, from 83% to 95%.

The second graph in Fig. 6 shows the variable epoch and its corresponding training loss. The gradual reduction in training loss is clearly visible with a higher number of epochs. The reduction can be seen in Fig. 6, from a 0.7822 training loss in the first 20 epochs to a final 0.1701 loss at 150 epochs.

4.1 Music Analysis

The model's generated music is analyzed using autocorrelation, PSD, frequency, noise filtering, and spectrum analysis, as illustrated below.

Autocorrelation detects repeated patterns in a signal and describes a signal's similarity to a time-shifted version of itself. The upper waveform in Fig. 7 is a time-domain plot of our model-generated output music file. The duration of the music is 30.348 s, and there are 1338368 samples in total. The music file's downward waveform traces the decreasing and increasing waveforms to the peak point and identifies the first peak position at 78, representing the autocorrelation function. This implies that the pattern will repeat itself after 78 sampled signals.

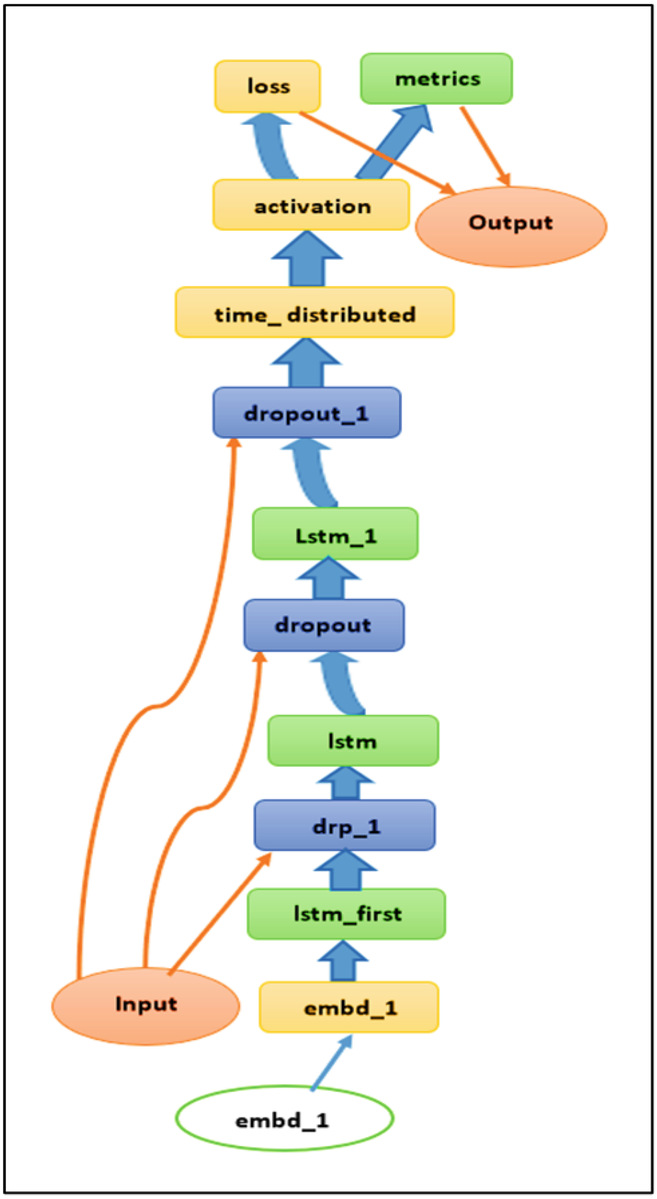
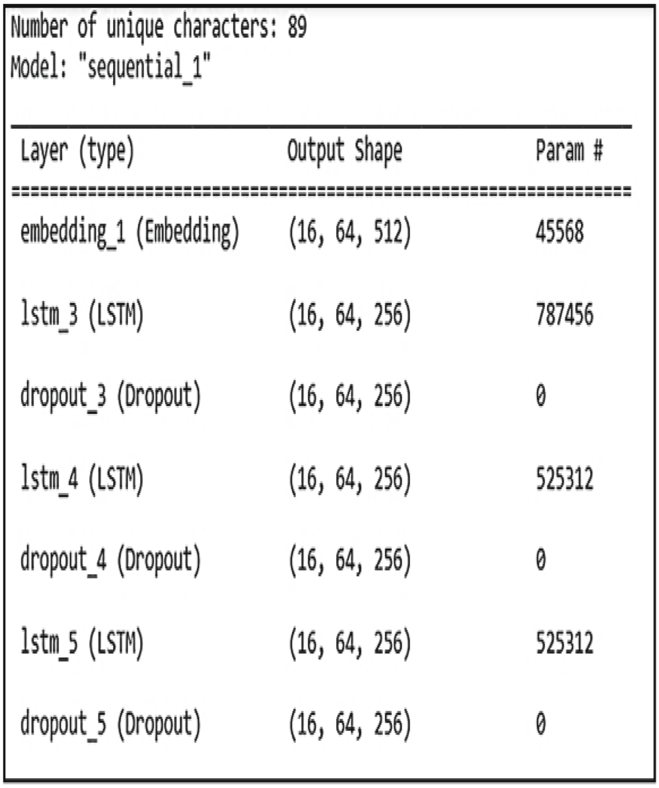


Fig. 3. The figure shows the data flow through the model's layers.

The Power Spectral Density (PSD) plot in Fig. 8 is a frequency-domain plot of power per Hz against frequency. The PSD graph shows that variations between 0 and 5 kHz are strong, while variations between 15 and 20 kHz are weak.

In music, frequency is the speed of vibration, which determines the pitch of the sound. A sound's pitch indicates how high or low a note is. The obtained music signal



```
Number of unique characters: 89
Model: "sequential_1"
_________________________________________________________________
 Layer (type)              Output Shape              Param #
=================================================================
 embedding_1 (Embedding)   (16, 64, 512)             45568

 lstm_3 (LSTM)             (16, 64, 256)             787456

 dropout_3 (Dropout)       (16, 64, 256)             0

 lstm_4 (LSTM)             (16, 64, 256)             525312

 dropout_4 (Dropout)       (16, 64, 256)             0

 lstm_5 (LSTM)             (16, 64, 256)             525312

 dropout_5 (Dropout)       (16, 64, 256)             0
```

Fig. 4. The figure shows the input-output shapes, various layers, and architecture.

has a frequency of 565.38 Hz. According to early research, this frequency is relaxing for the body and mind, as well as more harmonic and pleasant.

As shown in Fig. 9, the most efficient audio noise reduction tool is a Butterworth low pass filter, which is used to remove noise from an audio signal. A Butterworth filter has a frequency response in the passband that is as flat as possible. It is used to remove high-frequency noise with very minimal signal loss. A low-pass filter accepts low-frequency signals while rejecting high-frequency signals. The graph below shows a music signal's filtered and unfiltered time and frequency domain representation. The noise content in the model-generated music is extremely low, resulting in higher-quality music.

Figure 10 shows a graph depicting all of the frequencies present in music at a given time. So, the resulting graph is referred to as a spectrogram. The darker shades in the graph show frequencies with extremely low intensities, whereas the orange and yellow areas, represent frequencies in the music with high intensities.

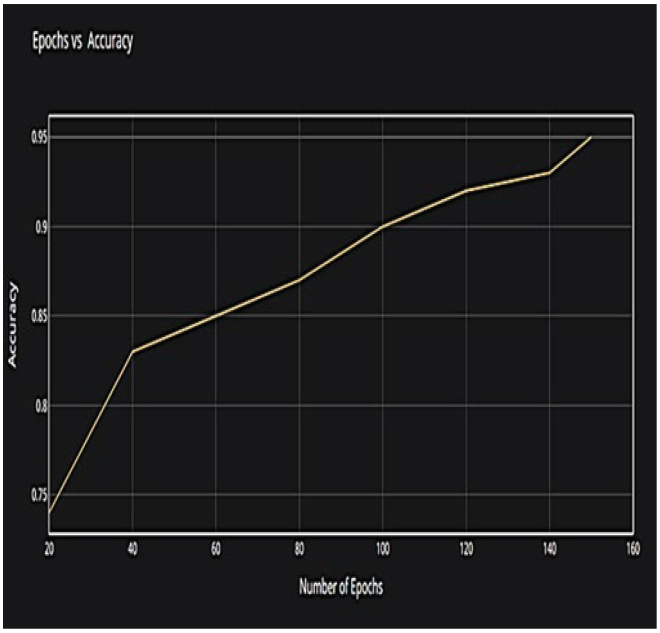


Fig. 5. The graph shows the epochs vs. accuracy of the model for 150 epochs.

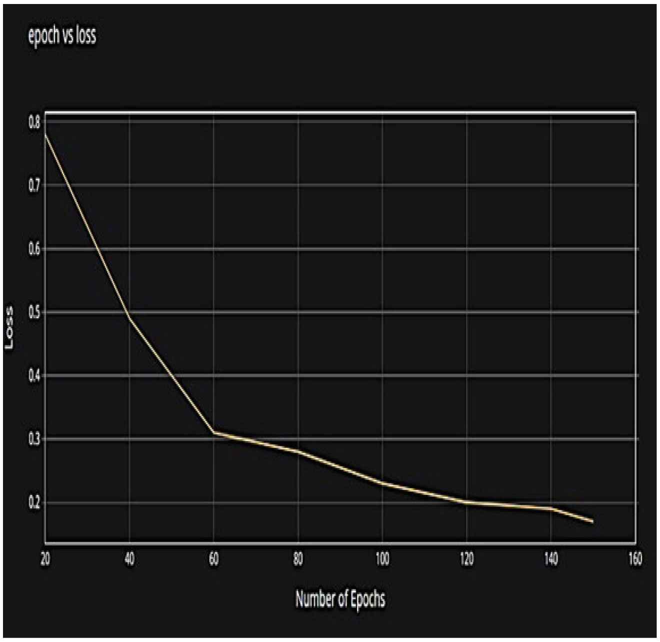


Fig. 6. The graph shows the epochs vs. loss of the model for 150 epochs.

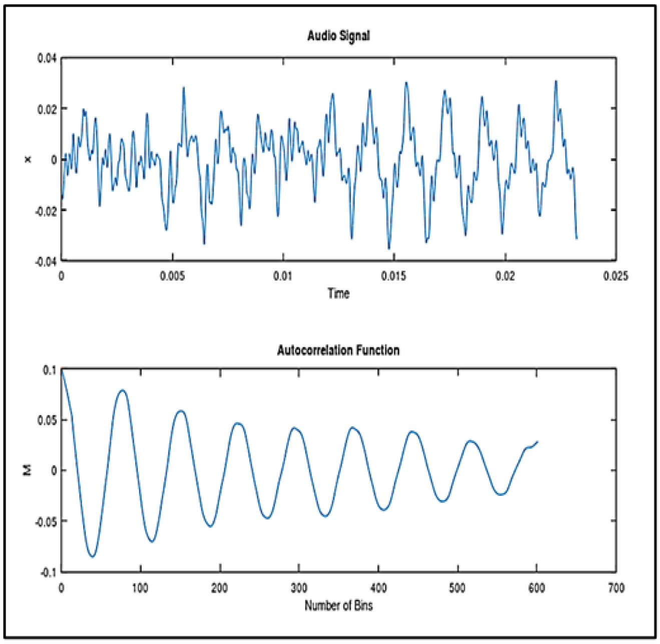


Fig. 7. Autocorrelation function of the model generated audio signal

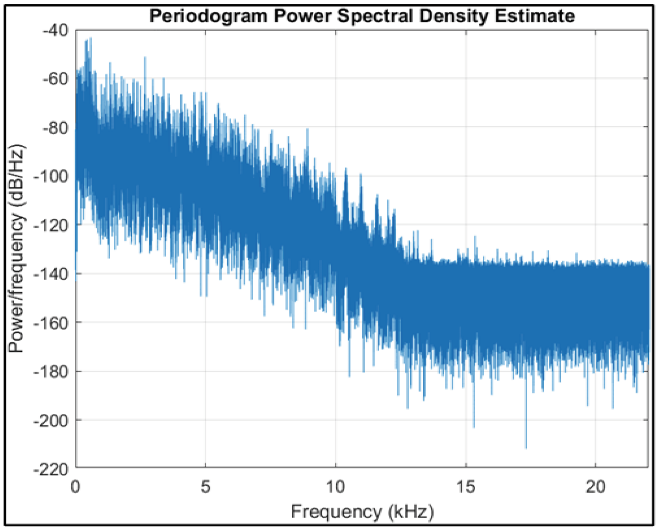


Fig. 8. PSD graph of power vs. frequency

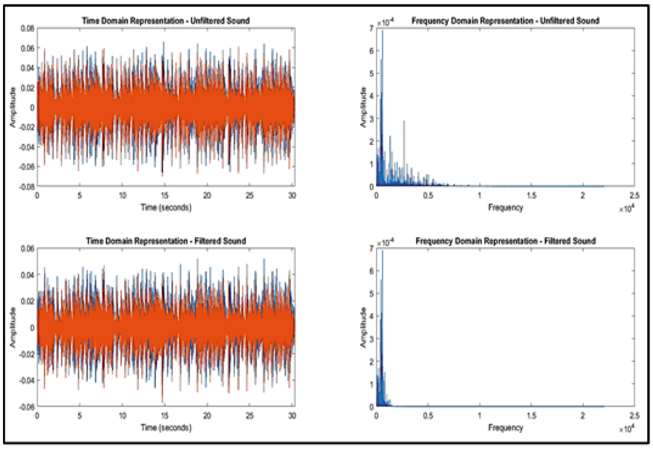


Fig. 9. Noise Filtering using Butterworth low pass filter

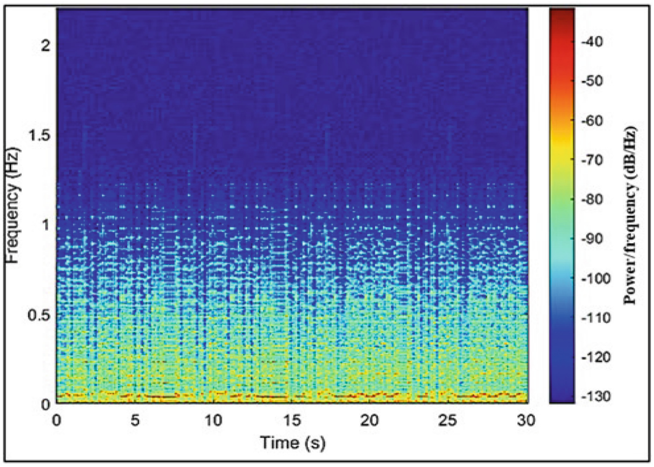


Fig. 10. Spectrogram of Music signal

5 Conclusion

This paper achieves the goal of building a model that is capable of automatically composing melodious music without any human interference. Using a multi-layer LSTM network, the model can recall previous dataset details and generate polyphonic music. It can also learn harmonic and melodic note sequences from the ABC notation dataset. The LSTM is better at analyzing a song structure and creating a more pleasant-sounding composition. After thoroughly analyzing the model, it achieved stellar results in composing new melodies with hyperparameter optimization, yielding an accuracy rate of 95%. The musical note generated from the trained model is C# (sharp), which has a good melody, longer duration, harmony, and decent quality.

In this paper, the overall research agenda is studied and analyzed, which will hopefully help to understand the issues and feasible solutions for music generation using deep learning.

6 Future Scope

As music impacts humans, it is useful to know which genres of music have positive or negative effects on an individual. In the future, we may apply new techniques to enhance learning and produce more enjoyable music, such as predicting emotions in music through audio pattern analysis and generating a model that can conclude the emotional rating of audio patterns on the human brain and act as a buffer for people to decide whether to listen to music or not based on the rating. The purpose of music generation through deep learning is to enhance AI and human interaction, and we can ultimately serve humanity by implementing automatic music-generating technologies in daily life.

References

Sanidhya Mangal, “LSTM Based Music Generation System,”2019.

Ramya Vidiyala, “Music Generation Through Deep Neural Networks,” 2020.

A. Joshi,” A Comparative Analysis of Algorithmic Music Generation on GPUs and FPGAs” Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.

F. a. H. J. Drewes,” An algebra for tree-based music generation,” 2007.

Boulanger-Lewandowski,” Modeling temporal dependencies in high dimensional sequences Application to polyphonic music generation and transcription,” 2012.

N. Hadjeres,” Interactive Music Generation with Positional Constraints using Anticipation-RNNs,” 2017.

Olof Mogren,” C-RNN-GAN: Continuous recurrent neural networks with adversarial training,” 2016

Nikhil Kotecha,” Generating Music using an LSTM Network”, 2018.

Aravind Pai, “Automatic Music Generation.” 2020.

Dr.P. SHOBHA RANI, S.V. PRANEETH, “Music Generation Using Deep Learning,” 2020.

Nikhil Kotecha,” Generating Music using an LSTM Network,” 2018.

Vaishali Ingale, “Music Generation Using Three-layered LSTM,” 2021.