Understanding the application of Markov chain, LSTM, and GAN for music generation

Srinidhi K M, Nidhi Sindhwani
Amity Institute of Information Technology, Amity University
Noida, Uttar Pradesh, India

Abstract
Deep learning plays a crucial role in music generation by allowing machines to learn complex musical patterns and structures. It enables the creation of new music styles and genres and assists in the creative process of musicians and composers. With deep learning algorithms, we can generate high-quality music that closely resembles human-composed music. This paper discusses three commonly used deep learning approaches, namely Markov chains, Long Short-Term Memory (LSTM), and Generative Adversarial Networks (GAN), which are widely used in music generation. This paper examines the fundamental functioning, recent advancements, notable use cases, strengths, and limitations of these three techniques. The paper attempts to provide a comprehensive understanding of each method and compares them by highlighting the weaknesses of each technique and how they can be compensated for by other methods.

Keywords: *Deep Learning, Artificial Intelligence, Markov Chain, Long Short-Term Memory, Generative Adversarial Networks, Algorithmic music*

1. INTRODUCTION

Art is considered something that originates from human experience and original creativity. The nature of such creative processes is changing by the development of technologies particularly in the field of Artificial Intelligence (AI). Artificial Intelligence, through deep learning techniques has opened the possibilities to generate new musical ideas. To generate anything with Artificial Intelligence, the algorithm designed must learn from a large amount of data, learn the underlying patterns, and generate output until a desired output is generated. In the case of music, a set of melodies, chords and beats can be taken as input by an algorithm. The algorithm will learn the properties and patterns of that input and generate something new based on that input [1]. Artificial Intelligence can generate user specific music, for instance, it can generate music according to a particular mood and genre, it can mimic a particular artist's style of music composition, or it can even complete an incomplete piece of music by studying its previous patterns. One of the most fascinating moments of AI generated music was when a new Nirvana song titled 'Drowned in the Sun' was posted online after more than 25 years of Kurt Cobain's death. Google's Magenta, an AI framework, was used to produce this track. This track's composition was produced based on dozens of original Nirvana recordings, its lyrics were generated using a neural network which was then delivered by a Nirvana tribute band's singer [2].

This paper covers Markov chains, Long Short-Term Memory (LSTM) and Generative Adversarial Networks (GAN), three popular deep learning techniques used in the field of music generation. Specifically, these techniques are discussed as it has been observed that most of the generative models developed for music generation are based on one or more of these techniques. These machine learning techniques can learn and mimic the patterns and structure of music efficiently.

Markov chains are a simple probabilistic model that can be used to generate sequences of notes or chords by analyzing patterns in existing musical data. These models are based on the assumption that the probability of the next note or chord in a sequence depends only on the previous note or chord. Markov chains are widely used in music generation because they are relatively simple to implement and can generate coherent musical phrases.

Long Short-Term Memory (LSTMs) are a type of neural network that can learn long-term dependencies in data. They are particularly useful for music generation because they can learn complex patterns in the timing, pitch, and dynamics of musical notes. LSTMs are trained on a large corpus of musical data and can then generate new music that follows similar patterns and structures to the original data.

Generative Adversarial Networks (GANs) are a specific kind of generative model that utilizes two neural networks, namely the generator and discriminator. The role of the generator network is to learn how to generate novel musical sequences, while the discriminator network's job is to differentiate between the newly created sequences and the original data. With time, the generator improves its capability to produce more realistic music that can deceive the discriminator. The unique feature of GANs is their ability to generate music that is not only diverse but also creative, which makes them more advantageous than traditional generative models.

This paper explores these three techniques by understanding its basic working, work done using this technique in the past five years, some notable examples implementing these techniques for music generation and

understanding the limitations and advantages of each technique.
This paper tries to compare these three techniques by discussing the drawbacks of each and how they are overcome by the other techniques.

2. LITERATURE SURVEY

The authors in [3] did a comprehensive overview of the recent uses of GANs for visual art generation, music composition and literary text generation. The authors have also presented a detailed description and comparison of performance of various GAN based architectures. Additionally, the challenges faced in the generation of art using GANs are also addressed.
In [4], the authors reviewed various AI methods in the field of music generation. Three categories based on abstraction were presented for the discussed AI methods which were categorized as nonadaptive, probabilistic and evolutionary. They analyzed conceptual blending, evolutionary computation and deep learning based methods on their performance in creating creative results.
In [5], the authors analyzed the trends and range of research on music generation using artificial intelligence. They reviewed available publications systematically to understand the work done till now in this field and its scope in the future. Their results concluded that there is an increasing global interest in the area of creative art generation using artificial intelligence. They found out that GAN-based models and transformers are gaining popularity for the design of generative models for music generation.
In [6], the authors presented limitations of deep learning for music generation. They analyzed the reason behind the issues encountered and presented some approaches that could possibly address the issues. Different examples of various systems were cited and analyzed. They introduced their strategies of tackling the issues by presenting actual architectures as examples.
The authors in [7] discussed algorithm composition and computational creativity. They presented a detailed description of techniques used for automatic music generation while highlighting its advantages and limitations. Furthermore, they review the debate on whether novel creative work can be generated by machines efficiently or not.
In [8], the authors presented a functional taxonomy for systems that generate music by referencing existing solutions. They organized solutions according to their purposes and analyzed the relations amongst them. Furthermore, they presented challenges encountered and their solutions to overcome them.
The authors in [9] built a system based on first-order Markov model to generate traditional bagana music. Their model was able to analyze structure and repetition encountered in their music. Their model handles long-term coherence efficiently. They also discussed improvements that could be done to enhance their model.
The authors in [10] presented a web application named FlowComposer that can generate musical lead sheets containing melodies and chords. Their system can generate lead sheets from scratch in the style mentioned by the user and can also from partial lead sheets provided by the user. They implemented their system using Markov chains.
The authors in [11] proposed a generative model for raw audio called WaveNet. WaveNet is a deep neural network that models the conditional probability distribution of raw audio samples given the previous samples. The network is based on a dilated causal convolutional architecture that allows it to generate long sequences of audio data. Furthermore, the authors compare WaveNet to other state-of-the-art models for audio generation and demonstrate its superior performance in generating high-quality audio.
In [12] the authors presented MidiNet, a convolutional generative adversarial network (GAN) that learns to generate symbolic-domain music in MIDI format. The paper concluded that MidiNet is a promising approach for symbolic-domain music generation. The authors suggested that MidiNet can be further improved by incorporating additional musical knowledge, such as chords and harmonies, into the model.
In [13], the authors discussed the problem that arises in music generation using Markov models. The issue is that the generated music can often sound too similar to the training data, resulting in a lack of creativity and originality. The paper presented a method for avoiding plagiarism in Markov sequence generation by introducing a measure of similarity between generated sequences and the training data.
The authors in [14] presented a novel approach for structured music generation based on sampling variations of sequences using Midi data. The authors introduce the concept of "variation modes" which was inspired by the idea that multiple interpretations can be given to the same sequence of events in music.

3. MARKOV CHAIN

The Markov chain is a mathematical tool used to describe how systems change from one state to another based on concepts from probability and matrix algebra. The Markov chain is a stochastic process and its basic idea was developed by the Russian mathematician Andrei Andreevich Markov. It describes a sequence of possible events which needs to satisfy the assumption that the probability of the occurrence of the next state depends only on its previous state rather than on all previous states in the sequence and thus is also known as a memoryless process as it does not depend on the memory of previous states.

Predictions made using the Markov chain can be made easily and accurately in many fields, including weather forecasting, natural language processing, finance, and sales. Language and speech are the most common applications of the Markov Chain, for example, predicting the next word. Music can be considered as a sequence of notes just like natural languages [15]. The sequence of next notes can be determined depending on its previous notes. Using Markov chains for music composition dates back to 1957, when Hiller and Isaacson composed the Illiac Suite on the ILLIAC I computer. Since then, Markov chains have been used to automatically generate music [16].

In the Markov chain, each node represents a state, while each edge is associated with a probability that indicates how likely it is that the source node will transition to the destination node. The sum of all the probabilities associated with the edges should be one. Markov chains can also be represented by a transition matrix, where the probability of moving from state x to state y is entered in the (x,y)th position.

When working with music, Markov chain's nodes represent sound objects. These sound objects contain information about a particular note, chord, duration, octave, velocity, and other dynamics. While the edges represent the probability of the next note from its previous one. A piece of music can be used as training data to estimate the probabilities in the Markov chain. Which will allow the Markov chain to generate music similar to the one it is trained with.

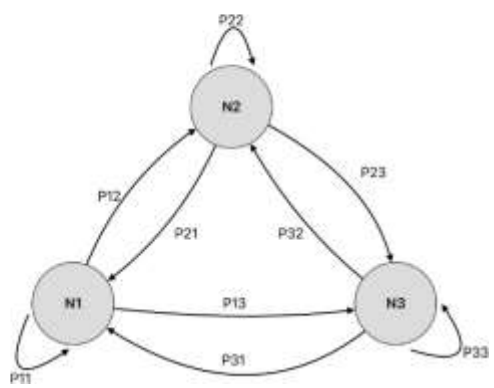

Figure 1. Topology for note movement in Markov Chain

One of the examples of implementing this theory practically in the past three years was when the authors in [17] used Python to experiment with the Markov chain's capabilities to generate music where the training data (musical piece) was given as a MusicXML file. Python's ElementTree library was used to parse the training data and NumPy library to manipulate matrices. The generated music had similarities to the trained data. The rhythmic properties and harmonic structure were intact. Some interesting note changes from the original piece were picked up by the model and encountered in the generated piece. They noticed that a simple Markov chain could generate music similar to the music piece it is trained with successfully. But has some limitations to it such as, it does not handle music pieces with multiple instruments or multiple voices when considered simultaneously.

Additionally, here are some of the notable examples where the markov chain has been used for generating music:

1. MidiMarkov [18] - It is an open-source software tool that generates music using Markov chains. It allows users to input a MIDI file and generates a new piece of music by analyzing the patterns and transitions in the original file.
2. Jukedeck [19] - It is a music composition platform that uses machine learning algorithms, including Markov chains, to generate royalty-free music for video and other media. The platform analyzes patterns in different music genres and generates new music that matches the user's input criteria.
3. EminemBot [20] - It is a Twitter bot that uses Markov chain to generate lyrics in the style of rapper Eminem. The bot was trained on a corpus of Eminem's lyrics and uses Markov chain to generate new lyrics based on the patterns and transitions in the original text.
4. MelodyRNN [21] - It is a neural network-based music generation system that uses both LSTM and Markov chain models to generate melodies. The system was trained on a large dataset of MIDI files and uses a combination of LSTM and Markov chain models to generate new melodies based on the patterns in the training data.

It can be concluded that the sequential nature of Markov chains can be very beneficial when working with a sequence of notes that make up a melody [22]. The assumption that the next note is only dependent on its previous note makes sense but at the same time limits the musical result and leaves it less interesting. Also the Markov chain can end reusing the original piece in a non-creative and repetitive manner.

4. LONG SHORT-TERM MEMORY

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that allows a neural network to remember the data it needs to hold on to. LSTM was invented in 1997 by Jürgen Schmidhuber and Sepp Hochreiter [23]. However, it also allows it to forget the data that is no longer relevant. Recurrent neural networks have a node that receives input. That input is then processed, resulting in an output. As it is a recurrent neural network, the output of a given step is provided alongside the input in the next step. It is because of this property that recurrent neural networks are capable of remembering previous steps in a sequence which would

be helpful in generating more interesting music compared to Markov chains.
LSTM comes into play as RNN suffers from what is known as the long-term dependency problem, which is to say that over time, as more and more information accumulates, RNNs become less effective at learning new things. LSTM provides a solution to this long-term dependency problem, adding an internal state to the RNN node. Now, when the RNN receives input, it receives the state information as well. This state is a LSTM cell that consists of three parts. Each part is a gate. There is a forget gate, an input gate and an output gate. The forget gate is responsible for deciding what sort of state information stored in this internal state can be forgotten and is no longer contextually relevant. The input gate is responsible for deciding what new information should be added or updated to this working storage state information. And the output gate is responsible for all the information that's stored in that state and which part of it should be output in a particular instance.

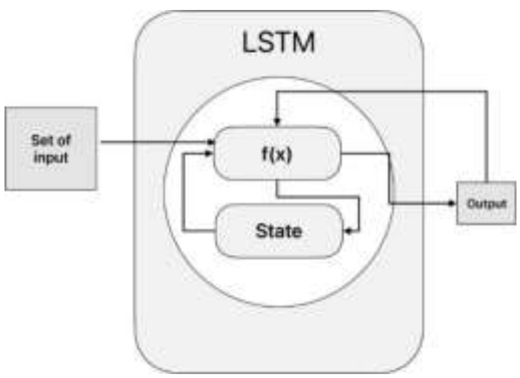

Figure 2. Basic architecture of LSTM

As long-term dependencies are taken into consideration by LSTMs, they are mostly used when working with sequential data. Sentiment analysis, chat bots, and video analysis are some of the most common applications of LSTMs. LSTMs can also be used to generate music as it can learn patterns and remember the dependencies in the data which in our case would be a musical piece.
D. Eck and J. Schmidhuber were the first researchers to experiment with LSTMs to generate music in 2002 [24]. They trained their model with blues musical pieces and showed how novel musical composition in the style of trained data can be generated using LSTMs successfully. The LSTM was able to generate music with proper structure.
There has been more research on LSTMs capabilities for generating music by many researchers in the past five years. The authors in [25] proposed an algorithm using LSTM that could generate melodies and musical pieces automatically. Their model which was designed to execute their algorithm could learn from a set of training data which in this case were musical pieces, analyze the data and generate a novel musical composition. Their model contained a single-layered network based on LSTM that processed musical data in the form of MIDI (musical instrument digital interface) files. They used Google Colab and Tensorflow for back-end and model training. Their model which took 54 minutes to train was able to generate polyphonic music, could learn harmonic structure and could recall previous dependencies.
The authors in [26] experimented with LSTM too. They trained their LSTM based model with 70 MIDI files and were able to generate a single piano track. They used Music21 and Tensorflow for back-end and data processing. Their model took 2 minutes per track to get trained and was able to predict the next sequence in the musical piece efficiently.
Additionally, here are some of the notable examples where LSTM has been used for generating music:

1. Magenta [27] - It is an open-source project developed by Google that uses deep learning techniques, including LSTM, to generate music. The project includes several models for music generation, including "MelodyRNN," which uses LSTM to generate melodies.
2. BachBot [28] - It is a neural network-based music generation system that uses LSTM to generate music in the style of Johann Sebastian Bach. The system was trained on a large dataset of Bach's compositions and uses LSTM to generate new music that matches the style and structure of Bach's music.
3. AIVA (Artificial Intelligence Virtual Artist) [29] - It is a music composition platform that uses machine learning algorithms, including LSTM, to generate original music. AIVA was trained on a large dataset of classical music and uses LSTM to generate new music that matches the user's input criteria.
4. Amper Music [30] - It is a music composition platform that uses deep learning techniques, including LSTM, to generate custom music for video and other media. The platform analyzes patterns in different music genres and uses LSTM to generate new music that matches the user's input criteria.

It can be concluded that the ability of LSTM to recall previous dependencies is beneficial in generating music with a better structure. Also, the music generated by LSTMs are not too repetitive and do not reuse the original musical composition non creatively. LSTM can successfully overcome the limitations encountered in the Markov chain efficiently. One of the limitations of LSTM for music generation is that it takes a lot of time to train which could make it less beneficial in real world applications. Also, LSTMs are more complicated and require more training data to get trained efficiently.

5. GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GAN), are a powerful type of neural network used for unsupervised machine learning. They were first introduced by Ian Goodfellow in 2014. They are commonly used in the field of image manipulation, generation of 3D objects and prediction of risks. They are made up of two competing models which run in competition with one another. These two are the generator submodule and the discriminator submodule. The generator's job is to create fake input or fake samples while the discriminator takes a given sample and figures out if it is a fake sample or if it's a real sample from the domain. And therein lies the adversarial nature of it. These generators create samples and update its model based on discriminators' decisions until it can generate a convincing sample. These generators and the discriminator are often implemented as Convolutional neural networks.

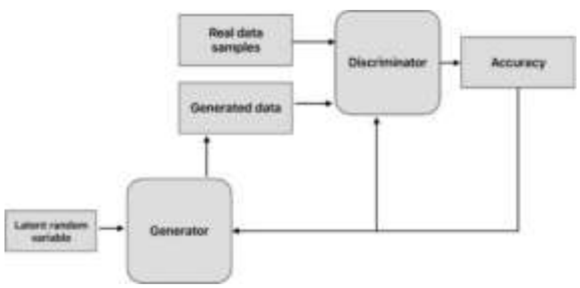

Figure 3. Basic architecture of GAN

Convolutional neural networks (CNN) are artificial neural networks with multiple layers that specialize in detecting patterns. CNNs have hidden layers called convolutional layers and these layers are precisely what makes a CNN perform its functionalities. This convolutional layer that receives input, transforms the input and outputs the transformed input to the next layer. This transformation is called a convolution operation. Each convolutional layer has a specific number of filters that detect patterns. The deeper the network gets, the more sophisticated these filters become. As CNNs are great at recognising patterns, they are commonly used in the area of visual recognition, facial detection and medical imagery. And when implemented with GAN, we get a very powerful tool for efficient data analysis and prediction.

GANs can be used to generate music as well as they are great at prediction and pattern detection. A lot of work in the field of generating music using GAN has been done. One of the examples of implementing GAN for music generation in the past five years was when the authors in [31] built a system based on GAN for fast audio generation. An autoencoder developed by the researchers was able to efficiently encode audio into a sequence with a low number of dimensions. After that, new sequences are generated by a GAN. They were able to generate techno and piano music successfully using their model. They showed how the process of generation could be conditioned on information like tempo and note density. GAN for music generation was also explored by the authors in [32]. They introduced MP3net which was a 2-D convolutional GAN that was capable of generating audio with long-range coherence. Their model was able to exhibit long-range coherence as each of their generators appended a higher octave and escalated the resolution along the frequency and time axis respectively. The temp and the rhythmic structure were intact in the music generated by their model. Their model was capable of generating music with proper harmonic structure and consistent chord structure as their training dataset.

There has also been an instance where GAN was used with an inception model which was able to generate music with variable length automatically. The authors in [33] proposed this type model. In order to process the time relationship of the data sequence, they added a time distribution layer and the quality of the music generated improved when this concept was combined with an inception model.

Additionally, here are some of the notable examples where LSTM has been used for generating music:

1. AIVA (Artificial Intelligence Virtual Artist), mentioned previously, also uses GANs in addition to LSTM for music generation. The GAN component is used to generate a wider variety of melodies and harmonies than what the LSTM alone could produce.
2. DALL-E 2 [34] - It is a generative art model developed by OpenAI that uses GANs to create images from textual descriptions. While not specifically designed for music generation, the model has been used to generate music album covers based on textual descriptions of the music.
3. GANSynth [35] - It is a system that uses GAN to generate realistic and expressive sound samples. It was developed by researchers at Google and can generate music in a variety of styles such as classical, pop, and jazz.
4. MuseGAN [36] - It is a GAN-based music generation system that generates multi-track music using four GANs. It was developed by researchers at the National University of Singapore and can generate music in various styles such as rock, jazz, and pop.
5. MidiNet - It is a deep neural network that uses GAN to generate music in MIDI format. It was developed by researchers at the University of California, San Diego and can generate music in a variety of genres.

It can be understood that GAN and CNN allowed models to generate music more instantaneously as compared to LSTMs. Before CNN and GAN were introduced for music generation, the models were not able to process raw audio samples directly. The training dataset was usually fed to the model in Midi formats that represents musical composition in a symbolic manner. But with CNN and GAN, the models can directly process and analyze raw audio which reduces dimensionality of the problems associated with music generation with deep learning techniques.

6. DISCUSSION

Markov chain, LSTM, and GAN are three popular techniques used in the field of music generation. Each method has its strengths and weaknesses, and the choice of method depends on the specific application and data available.

Markov chain is a simple and effective method for music generation that models the probability of each note or chord transition in a given piece of music. It is easy to implement and requires less computational resources compared to other methods. However, Markov chain models suffer from the "memoryless" property, which means that they can only capture short-term dependencies and may result in repetitive and less coherent music.

In contrast to Markov chain, LSTM is a type of recurrent neural network that can capture long-term dependencies in sequences of music notes, unlike Markov chain. Through training on vast music datasets, LSTM models can learn the structure and patterns of music, allowing them to generate new music pieces by sampling from the acquired distribution. Compared to Markov chain models, LSTM models generate more varied and consistent music. Nonetheless, training LSTM models requires larger amounts of data and time. It's also worth noting that the music produced may not always correspond to the desired genre or style.

Meanwhile, GANs are a potent method for music generation that utilizes a generator and a discriminator. The generator's task is to produce music that is similar to the training data, while the discriminator's job is to differentiate between real and generated music. This competition helps the generator to enhance its ability to generate high-quality music that is virtually indistinguishable from human-composed music. Despite producing music of very high quality, GANs demand substantial computational resources and can be prone to issues such as mode collapse.

Table 1. Comparison between Markov chain, LSTM and GAN for music generation

Factors	**Markov Chain**	**Long Short-Term Memory**	**Generative Adversarial Networks**
Architecture	Probabilistic state transition model	Recurrent neural network with memory	Two competing neural networks generating data
Type of training data usually used for music generation	MIDI files, symbolic music representation, ABC notation	MIDI files, symbolic music representation, audio files (when implemented with CNN)	Spectrograms, audio files, MIDI files
Computation time for training	Fast	Medium	Slow
Computation time for generation	Fast	Slow	Slow
Memory Requirement	Low	High	Very high
Model complexity	Simple	Complex	Very complex
Quality of generated music	Simple and repetitive	Complex and varied	Complex and varied
Flexibility in generating new music styles	Limited	Strong	Strong
Creativity	Limited	Medium	High
Major limitation	Lack of long-term memory	Need for huge training data	Need for careful tuning
Ease of use	Easy	Medium	Difficult

An overall comparison between Markov Chain, LSTM, and GAN for music generation is presented in table 1. In summary, the Markov chain is a simple and efficient method for generating music, but it is limited in its ability to capture long-term dependencies. LSTM models can capture longer-term dependencies and produce more coherent and diverse music but require larger amounts of training data and longer training times. GANs can produce very high-quality music that is indistinguishable from human-composed music, but they require significant computational resources and may suffer from mode collapse. In practice, the choice of method for music generation depends on the specific application and the resources available.

Furthermore, in order to gain direct experience in music generation through artificial intelligence, I tried out Magenta Studio, a tool for music generation developed by Google's Magenta project. This software employs machine learning algorithms to produce innovative and individualized melodies, chords, and rhythms. Specifically, I utilized Magenta Studio's "Continue" feature, which predicts possible notes to follow a given composition by analyzing the input musical data. To test this tool, I provided it with a MIDI file featuring a one-bar melody with two seconds of running time at 120 bpm as input. As a result, the software generated a three-bar melody that complemented the original input melody.

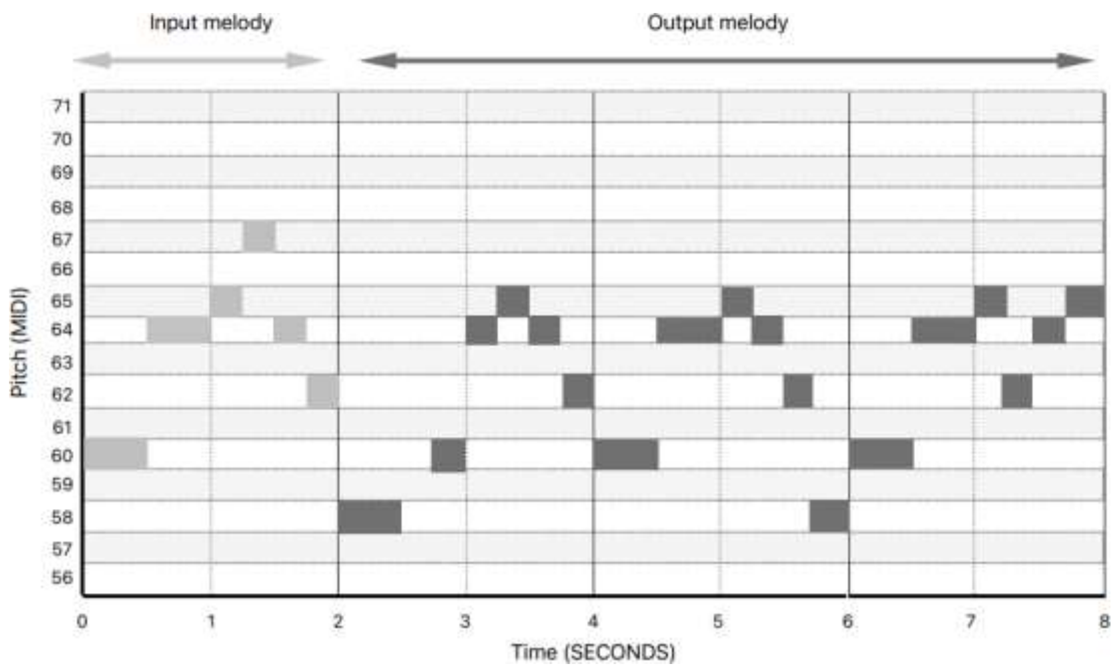

Figure 4. Input melody and generated melody

Figure 4 presents a graphical representation of both the input and the resulting music generated. The input melody lasts for two seconds, followed by the generated music. An observation table has been presented below in table 2 to facilitate an understanding of the effectiveness of the generated music. Magenta Studio's "Continue" exhibited proficient analysis of the input melody, generating a continuation melody that remained faithful to the D natural minor scale. Notably, the software incorporated nuanced variation into the continuation melody, taking care not to overindulge as to avoid a

.

random and inorganic sound. Furthermore, the software demonstrated its innovative capability by introducing a rest note, an omission not present in the original melody. It should be acknowledged, however, that the resulting melody's perceived simplicity may not resonate with all listeners, subject to individual preferences. In conclusion, while the software's music generation capacity is undoubtedly efficient, the contribution of a trained music professional may be necessary to fully optimize its potential.

Table 2. Observation of generated music

Bar	Seconds	Observation
2	2-3	• It starts with a fresh note not present in the input melody while keeping the melody's scale intact. • A rest of 0.25 seconds has been generated which elevates tension in the melody.
	3-4	• This part has similar pattern of four notes with a duration of 0.25 seconds as the once in input melody with some variations.
3	4-5	• This section is same as the first two notes of the input melody which helps in creating similarities as it is supposed to be a continuation of the input melody.
	5-6	• This section has four notes with a duration of 0.25 seconds but this time these notes are going down in pitch continuously which brings variety in its pattern.
4	6-7	• It starts with the same notes as the previous bar which might sound a bit repetitive .
	7-8	• This section maintains a similar pattern of four notes with a duration of 0.25 seconds with some variations.

7. CONCLUSION

The scope of music generation using deep learning is vast and rapidly growing. Deep learning algorithms have been successful in generating music that sounds like it was created by human composers, and they have the potential to revolutionize the music industry. One of the most significant advantages of deep learning in music generation is its ability to learn and replicate complex musical patterns and structures. Deep learning models can analyze vast amounts of music data and learn to recognize patterns in melody, harmony, rhythm, and other musical elements. These models can then use this knowledge to generate new music that follows similar patterns and structures.

The goal of this paper was to discuss Markov Chains, Long Short-Term Memory (LSTM) and Generative Adversarial Networks (GAN), three popular deep learning techniques used in the field of music generation. This paper tries to compare these three techniques and understands that the Markov chain is a straightforward and effective technique for creating music, but it has a drawback in its capability to take into account long-term connections. LSTM models, on the other hand, can handle more extended dependencies and generate more varied and consistent music. However, they need a vast amount of training data and more time for training. GANs are capable of generating music of superior quality that closely resembles human-produced music, but they require significant computational power and may face the issue of mode collapse.

References

[1] Mage. “How is AI used to generate music?” *DEV Community*, 19 January 2022, https://dev.to/mage_ai/how-is-ai-used-to-generate-music-4a4h. Accessed 7 March 2023.

[2] Schroer, Alyssa. “AI Music: 9 Examples to Know.” *Built In*, 6 October 2022, https://builtin.com/artificial-intelligence/ai-music-examples. Accessed 18 February 2023.

[3] Shahriar, Sakib. "GAN computers generate arts? a survey on visual arts, music, and literary text generation using generative adversarial network." *Displays* (2022): 102237.

[4] Kaliakatsos-Papakostas, Maximos, Andreas Floros, and Michael N. Vrahatis. "Artificial intelligence methods for music generation: a review and future perspectives." *Nature-Inspired Computation and Swarm Intelligence* (2020): 217-245.

[5] Civit, Miguel, et al. "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends." *Expert Systems with Applications* (2022): 118190.

[6] Briot, Jean-Pierre, and François Pachet. "Music generation by deep learning-challenges and directions." *arXiv preprint arXiv:1712.04371* (2017).

[7] Fernández, Jose D., and Francisco Vico. "AI methods in algorithmic composition: A comprehensive survey." *Journal of Artificial Intelligence Research* 48 (2013): 513-582.

[8] Herremans, Dorien, Ching-Hua Chuan, and Elaine Chew. "A functional taxonomy of music generation

systems." *ACM Computing Surveys (CSUR)* 50.5 (2017): 1-30.

[9] Herremans, Dorien, et al. "Generating structured music for bagana using quality metrics based on Markov models." *Expert Systems with Applications* 42.21 (2015): 7424-7435.

[10] Papadopoulos, Alexandre, Pierre Roy, and François Pachet. "Assisted lead sheet composition using flowcomposer." *Principles and Practice of Constraint Programming: 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings 22*. Springer International Publishing, 2016.

[11] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[12] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017), Suzhou, China, October 2017.

[13] Papadopoulos, Alexandre, Pierre Roy, and François Pachet. "Avoiding plagiarism in Markov sequence generation." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. No. 1. 2014

[14] Pachet, François, Alexandre Papadopoulos, and Pierre Roy. "Sampling Variations of Sequences for Structured Music Generation." *ISMIR*. 2017.

[15] Alexander Osipenko. "Markov Chain for music generation." *Towards Data Science*, 7 July 2019, https://towardsdatascience.com/markov-chain-for-music-generation-932ea8a88305. Accessed 3 March 2023.

[16] Sandred, Orjan, Mikael Laurson, and Mika Kuuskankare. "Revisiting the Illiac Suite–a rule-based approach to stochastic processes." Sonic Ideas/Ideas Sonicas 2 (2009): 42-46.

[17] Shapiro, Ilana, and Mark Huber. "Markov chains for computer music generation." *Journal of Humanistic Mathematics* 11.2 (2021): 167-195.

[18] "Home." *YouTube*, https://pjb.com.au/midi/midimarkov.html. Accessed 4 March 2023.

[19] Newton, Ed. "Jukedeck: building a creative AI business - The Entrepreneurship Centre blog." *The Entrepreneurship Centre blog* -, 14 February 2019, https://entrepreneurship.blog.jbs.cam.ac.uk/2019/02/14/jukedeck-building-a-creative-ai-business/. Accessed 4 March 2023.

[20] Behera, Varun. "Eminem Bot - Varun's Portfolio." *Varun's Portfolio*, https://portfolio.varunbehera.com/projects/mnmbot. Accessed 5 March 2023.

[21] "magenta/README.md at main · magenta/magenta · GitHub." *magenta/README.md at main · magenta/magenta · GitHub*, https://github.com/magenta/magenta/blob/main/magenta/models/melody_rnn/README.md. Accessed 6 March 2023.

[22] Carnovalini, Filippo, and Antonio Rodà. "Computational creativity and music generation systems: An introduction to the state of the art." *Frontiers in Artificial Intelligence* 3 (2020): 14.

[23] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[24] Eck, Douglas, and Juergen Schmidhuber. "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks." Proceedings of the 12th IEEE workshop on neural networks for signal processing. IEEE, 2002.

[25] Mangal, Sanidhya, Rahul Modak, and Poorva Joshi. "Lstm based music generation system." *arXiv preprint arXiv:1908.01080* (2019).

[26] Xu, Xin. "LSTM networks for music generation." arXiv preprint arXiv:2006.09838 (2020).

[27] Magenta Tensorflow, https://magenta.tensorflow.org/. Accessed 27 March 2023.

[28] Liang, Feynman. "Bachbot: Automatic composition in the style of bach chorales." University of Cambridge 8 (2016): 19-48.

[29] *AIVA - The AI composing emotional soundtrack music*, https://www.aiva.ai/. Accessed 2 March 2023.

[30] "Home." *YouTube*, https://www.linkedin.com/company/amper-music-inc-/about/. Accessed 4 March 2023.

[31] Pasini, Marco, and Jan Schlüter. "Musika! fast infinite waveform music generation." arXiv preprint arXiv:2208.08706 (2022).

[32] Broek, Korneel van den. "Mp3net: coherent, minute-long music generation from raw audio with a simple convolutional GAN." arXiv preprint arXiv:2101.04785 (2021).

[33] Li, Shuyu, and Yunsick Sung. "INCO-GAN: variable-length music generation method based on inception model-based conditional GAN." *Mathematics* 9.4 (2021): 387.

[34] "DALL·E 2." *OpenAI*, https://openai.com/product/dall-e-2. Accessed 7 March 2023.

[35] Engel, Jesse, et al. "Gansynth: Adversarial neural audio synthesis." *arXiv preprint arXiv:1902.08710* (2019).

[36] Dong, Hao-Wen, et al. "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 20