Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 246 (2024) 1855–1864

Procedia Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

Music Generation with Machine Learning and Deep Neural Networks

Tudor-Constantin Pricop, Adrian Iftene

[a]*Faculty of Computer Science, "Alexandru Ioan Cuza" University, General Henri Mathias Berthelot Street, No. 16, 700259, Iasi, Romania*

Abstract

This paper explores advanced music generation through hybrid models combining deep neural networks, machine learning algorithms, variational autoencoders (VAEs), long short-term memory (LSTM) networks, and Transformers to create diverse and engaging musical experiences. Our research aims to advance the understanding of music's impact on our lives and develop methodologies to create diverse and engaging musical experiences tailored to individual preferences. We begin by extracting relevant features from a large and diverse collection of music samples from different genres. These features, encompassing spectral properties, rhythmic patterns, and tonal characteristics, serve as the foundation for our generation models. To generate music, we explore the potential of VAEs, LSTMs, and Transformers, each offering unique capabilities for handling different aspects of the task. VAEs are employed to learn a continuous latent space representation of the music samples, enabling the generation of novel compositions within a specified genre. LSTMs and Transformers, on the other hand, are used to model the temporal dependencies and intricate patterns inherent in music. While not claiming state-of-the-art performance, our approach demonstrates promising outcomes in generation tasks, showcasing its potential to enhance music-related applications such as recommendation systems and creative tools for composers.

Keywords: Deep neural networks; machine learning algorithms; variational autoencoders; long short-term memory networks; transformer

1. Introduction

The transformative power of music has been a cornerstone of human culture and expression throughout history. As diverse as our societies, the vast array of music genres reflects our rich cultural heritage and individual creativity. With the rapid advancements in technology, new paradigms in music analysis and synthesis are emerging, further

* Adrian Iftene. Tel.: +4-023-220-1091.
E-mail address: adrian.iftene@info.uaic.ro

1877-0509
10.1016/j.procs.2024.09.692

enhancing our understanding of this universal language. The motivation for this research paper stems from a desire to harness the potential of cutting-edge learning techniques, such as Machine Learning, Deep Neural Networks (DNNs), Long Short-Term Memory (LSTM) networks, Variational Autoencoders (VAEs), and Transformers, to address one important challenge in the field of music informatics: music generation [1]. By integrating distinct models such as the abovementioned, we harness the unique strengths of each to offset the limitations of the others. This strategy highlights the potential for substantial advancements in music generation, leading to more nuanced and expressive musical compositions.

The creative process of composing music has long been considered the exclusive domain of human intellect. However, recent advancements in artificial intelligence and machine learning have demonstrated the potential for machines to generate music with remarkable coherence and originality [2], [3], [4]. This research explores the application of LSTM, VAE, and Transformer models to the task of music generation, to develop systems capable of producing high-quality compositions across different genres while respecting the unique characteristics and structures inherent to each. By combining expertise in music theory, machine learning, and artificial intelligence, this research paper aims to contribute to the ongoing efforts to revolutionize the field of music informatics. It is hoped that the findings will not only serve as a foundation for future research endeavors but also pave the way for innovative applications that enrich our understanding and appreciation of the world of music.

The next section of this paper, Related Work, reviews the existing literature and technologies in the field of music generation, highlighting key advancements and current methodologies. This is followed by the Proposed Solution section, which details our approach, including the data collection, feature extraction, and the architecture of the hybrid models combining VAEs, LSTMs, and Transformers. Subsequently, the Results section presents the outcomes of our experiments, demonstrating the effectiveness and potential of our approach. Finally, the Conclusion section summarizes the findings, discusses the implications of our work, and suggests directions for future research .

2. Related Work

2.1. JukeBox

The paper [5] describes a generative model called JukeBox, that can create music and singing in raw audio domains. It uses a multi-scale VQ-VAE (Vector Quantized Variational AutoEncoder) to compress raw audio to discrete codes and models those using autoregressive Transformers. This model can be conditioned on the artist and genre to steer the musical style. It was trained on a large dataset of 1 million songs, paired with the corresponding metadata. After a slow training process due to the complexity of the model, the results shown are quite remarkable. It generates coherent music pieces with harmony, rhythm and even singing in multiple genres. Even if the model has its limitations in controlling the high-level attributes of the generated song, its capabilities in mimicking artists' styles and generating lyrics are still impressive.

2.2. Groove2Groove

Another interesting approach regarding music generation in a specific style is presented in [6]. The paper references one-shot transfer, which involves taking a piece of music in one style (jazz) and transforming it into another style (rock), using only a single example of the target style. The model consists of a style encoder, which takes as input a single example of the target and encodes it, and a decoder which takes as input a MIDI file and the target style and outputs a new MIDI file containing the original content in the target style. The model is evaluated on a variety of musical styles (not only broad genres) and shows that it can perform the transfer with high fidelity. There is also a user study conducted by the team which shows that participants prefer the output of this model over other state-of-the-art methods. This approach has the potential to be very useful in a variety of musical applications like remixing music.

2.3. Theme Transformer: Symbolic Music Generation with Theme-Conditioned Transformer

A more recent paper [7] proposes a novel approach for generating symbolic music by conditioning a Transformer model on thematic material, rather than using the more common prompt-based conditioning. The theme-based conditioning approach involves using a separate Transformer encoder to process a conditioning sequence, which influences

the generation of music in the Transformer decoder through cross-attention mechanisms. This method aims to ensure that the conditioning theme repeats and varies appropriately throughout the generated piece. To achieve this, the authors developed a technique to automatically retrieve thematic material from music pieces using contrastive representation learning and clustering. This method segments music into fragments, clusters them to identify recurring themes, and selects representative fragments as the thematic condition. Regarding its performance, the model was evaluated both objectively and subjectively against traditional prompt-based models. The results showed that the Theme Transformer can generate polyphonic pop piano music that better incorporates and varies the thematic material, offering more musically coherent and interesting compositions compared to previous baseline models.

2.4. Existing Applications

Spotify is a popular digital music, podcast, and video streaming service that gives access to millions of songs and other content from creators all over the world [8]. Artificial Intelligence (AI) plays an integral role in Spotify's functionality, analyzing the acoustic, cultural, and personal inputs for every user, to create personalized recommendations. There are even some AI-driven systems that predict upcoming hits based on users' behavior patterns. Moreover, Spotify is developing several models to create music and provide songwriting assistance, showcasing AI's potential in the creative aspects of the music industry.

SoundHound is a versatile and popular music application that serves as a platform that identifies songs, displays real-time lyrics, and even offers a voice-controlled AI [9]. The functioning of SoundHound's song identification system is based on a methodology known as audio fingerprinting. A user plays a song near the device running SoundHound, the app listens to a segment of a song, transforms it into a unique numerical identifier (fingerprint), and matches it against a vast database of music. The process is very fast and can identify songs within a matter of seconds, even in noisy environments. Furthermore, it has a unique ability to identify a fingerprint even on a user's humming, singing, or whistling, without the need for the original recording or lyrics.

Soundraw is an AI-driven platform designed to democratize the music production process [10]. The app enables users to generate original music compositions based on various attributes, accessible to both novices and musicians alike. One of the most impressive features is the ability to create music according to a genre. The result can also be fine-tuned to reflect a desired mood or a theme, leading to highly personalized pieces of music tailored to users' preferences. Other powerful features include tempo and length altering and also audio editing tools like specifying notes, changing instruments, or adjusting the mix of the track.

3. Proposed Solution

3.1. Music Notation

Music notation serves as a visual coding system to depict music, using a set of symbols. Each music notation is different and is used in its way, for example, the tablature notation is instrument-specific and provides information about the physical placement of the performer's fingers. The tab consists of horizontal lines that represent the strings of the instrument and numbers that represent the frets where the fingers should be placed. It can also indicate techniques such as slides, harmonics, or vibrato. Despite the effectiveness of many traditional forms of musical notation for live performances, they have visible limitations when it comes to playing with music in the digital realm. Here, the Musical Instrument Digital Interface (MIDI) excels. MIDI [11] is a protocol that enables computers and musical instruments to communicate with each other. It does not contain any sound, as it encodes information about the audio track, like the pitch, velocity, vibrato, or volume. The nature of this notation allows for easy transposition, and changing time signatures or instruments (the piano part can be easily switched with any other instrument in the library). MIDI is also very efficient and economical: a symphony that might require hundreds of traditional sheet pages can be stored in a lightweight file, making it perfect to work with for generation tasks 1.

3.2. Datasets

When seeking the perfect dataset for the music genre classification task, it is essential to consider the size of the dataset, its diversity, balance, and the quality of the genre labels. A synthesis of many musical datasets is presented in

Octave	Note numbers											
	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

Fig. 1. Musical notes in MIDI format, having a vocabulary-like representation, making it suitable for generation tasks

[12]. For our experiments, we took into account only three of the most commonly utilized datasets for this purpose: GTZAN [13], Free Music Archive (FMA) [14], and the Million Song Dataset (MSD) [15]. Each dataset has its unique strengths and weaknesses, that were carefully analyzed before choosing the perfect candidate for analysis and experimentation.

Usually, **GTZAN** is the first choice for music genre recognition tasks, primarily due to its simplicity, reduced size, and balanced structure. It provides a neat, evenly distributed collection of a thousand audio tracks across ten popular genres. Because of the balanced genres, the need for additional preprocessing is eliminated; otherwise, the class imbalance could have caused bias in the models. The collection contains a folder with the 30-second audio files, each subfolder is labeled with one of the 10 genres and contains the respective 100 files. The other folder contains the visual representation of each audio file (Mel Spectrograms), also distributed in 10 balanced subfolders.

The **Free Music Archive (FMA)** dataset is a richly annotated, high-quality, and diverse collection of music that can be utilized for various research tasks related to music analysis, including genre tagging. It provides several levels of annotation, for each track there is a unique ID, title, album, and release year. Additionally, there are track genres and sub-genres, providing a robust groundwork for a broad spectrum of music analysis tasks. Tracks are categorized according to a hierarchical taxonomy that spans 16 top-level genres (like Pop, Rock, and Electronic), which are further broken down into 161 sub-genres (such as Psych-Rock, Lo-Fi, and Drone). One important feature of this dataset is its availability in subsets of different sizes: small, medium, large, and full versions.

The **Million Song Dataset (MSD)** is a publicly accessible collection of audio features and metadata for a million contemporary music tracks. With a million songs included, its sheer size offers an extensive range of data for all kinds of tasks in music research, including music genre classification. Each song includes data points like the release year, the artist, the popularity, the key, tempo, or duration. In addition to this, there are segments, bars, and beats indicating the rhythm and timbre of audio segments, which give an idea of the sound color and melody. The data for each track has been computed using an API called The Echo Nest, which gives objective information about each song, unaffected by subjective factors like personal opinion or cultural context. However, there are also a few drawbacks to this dataset. Firstly, it includes only metadata and precomputed features, it does not contain the actual audio files for the songs, due to copyright reasons. For researching deep learning techniques using Mel spectrograms, this dataset is limited. Secondly, the MSD is heavily weighted towards popular Western music. Even if it offers an extensive range of data for this segment, it does not provide a diverse or representative sample of all genres within world music.

Additionally, we use the **Spotify API**, which is a rich source of music data, providing access to a wide variety of information such as track details, artist information, album details, playlists, and even audio analysis and features, but the most important thing is that it contains genre labels for each track, making it a suitable source for building up

a dataset for music genre recognition. Downloading from the Spotify API requires client credentials obtained after making an account on the Spotify Developer platform [8]. The genres used to quiz the API are the following: blues, classical, country, disco, reggae, metal, hip-hop, jazz, pop, and rock. It can be seen that the list is very similar to the one used in GTZAN, to make pertinent comparisons between the 2 datasets. The final version of the dataset contains two folders, *Audio_Spotify* with 500 audio tracks per genre and *Audio_Spotify_Test* with 300 tracks per genre, and can be found here [16].

The first step towards generating music in a particular style requires an adequate dataset that represents the genre well. While studying the 10 popular music genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock), the one that stood out from the rest was jazz. Firstly, it is a highly improvisational style, being the perfect playground for an AI model to "improvise" within the given musical parameters. Jazz is often seen as a deeply emotional genre, with musicians using various techniques to express their feelings through instruments, and while AI cannot experience human emotions, it can be trained to understand and replicate some musical patterns that emanate specific grooves and feelings. Such a jazz-inclined dataset can be found here [17].

With raw audio files, the generation of new, coherent melodies from scratch is a particularly challenging task due to the high dimensional, time-series nature of this data. To simplify the process of understanding the underlying structure of an audio track, one should consider using other musical notations which in exchange for a simpler, programmatically-friendly interface, gives up the subtle variations in timing and dynamics. Such a practical notation is called MIDI and contains structured, symbolic information about a piece of music. The chosen dataset contains 935 audio files in MIDI format, including several pre-computed properties in CSV format.

Firstly, the dataset is preprocessed, transforming the musical information into a format similar to text processing. For each audio track the notes are extracted and outputted in their musical notation (F2), also the chords are extracted as sequences of integer pitch classes, all notes of the chord being transposed to a single octave. Since this process requires a considerable amount of time to complete, the preprocessed notes are saved as text in a file for further usage.

The sequences of notes and chords are then prepared for the model as follows: each sequence of 100 notes represents the input for the next note in the sequence of all notes. The model will learn to predict a new pitch for a musical pattern given as input and will shift the sequence and repeat this process until the desired length of the audio track is reached.

Regarding Deep Learning methodologies, for each audio track, 4 Mel spectrograms were created: the original, unmodified version, one with a random frequency mask, one with a time mask, and the last one which has noise added to it (noise with mean 0 and standard deviation 0.3). This process is called Data Augmentation and it is used to increase model robustness and to avoid overfitting in neural networks. Thus, the dataset of visual representations reached a total of 32,000 files, distributed evenly in 10 folders, and can be found here [18].

For the Machine Learning techniques, 2 CSV files were created, found in the original dataset, that contain the mean and variance of each song for several features, including zero-crossing rate, spectral features, harmonics, percussive, mfcc, and others. These audio properties were computed, similarly to the GTZAN dataset, for the entire 30-second tracks and for smaller 3-second windows (to increase the amount of data fed to the models).

3.3. Model

The model used for generating new, pleasant Jazz music is a generative model called Variational Autoencoder (VAE), which is composed of 2 primary components: an *encoder* and a *decoder*. The goal is to ensure that the reconstructed input is as close as possible to the original input, while also ensuring that the latent space has certain properties (for example that it follows a standard normal distribution). The encoder and the decoder are given as parameters to the network, as they are defined separately.

The architectures of the 2 hybrid models: VAE 2 and Transformer 3 have a similar structure, but differ in the way the recurrent data is processed, as one leverages LSTMs and the other Transformers as encoders/decoders mechanisms. The *encoder* consists of 2 LSTM layers, each with 256 units. This is followed by 2 Dense layers that output a mean and a log variance, and have the dimension of the latent space, which is 64. These variables represent the parameters of the Gaussian distribution that the encoder learns to map the data to. The sampling layer that is applied afterward uses the previously mentioned parameters to generate a latent vector, which is the encoded representation of the input. This layer uses the mean and the variance to sample from the distribution defined by these parameters. The purpose of it is to introduce randomness in the encoding process, which ensures that points that are close in the input space are

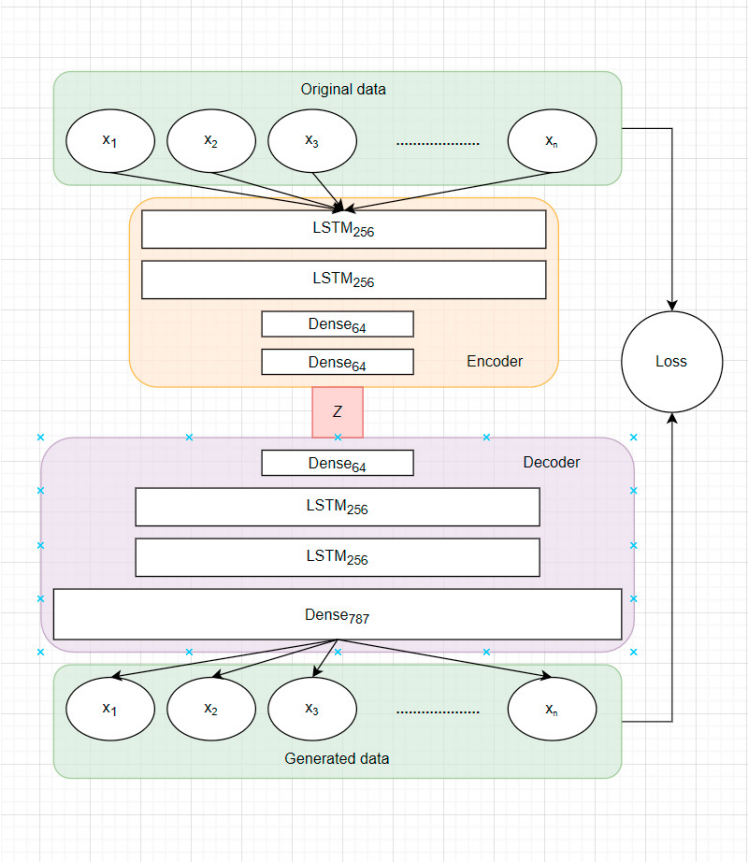

Fig. 2. Architecture of VAE that uses LSTM to encode/decode data.

also close in the latent space. This makes the latent space continuous, which is useful for the generation of new data points. The *decoder* also consists of 2 LSTM layers of 256 units, but the input is expected to have the dimension of the latent space. Before entering the LSTM cells, the input is fed to a Dense layer that upscales the latent vector to a larger representation. Afterward, the vector is passed to a Dense layer that outputs a probability (SoftMax activation) for each unique chord or note in the vocabulary.

The metrics tracked during the training process are the reconstruction loss, Kullback-Leibler (KL) loss, and the total loss. These losses are computed manually, as Keras does not yet support the VAE pre-computed model. Training a VAE can take considerable time, for a bunch of reasons: musical data is often high-dimensional and complex, the model is also complex, including LSTM and Dense layers, and the process involves multiple stages as it must learn both to encode and decode the representations accurately. After the training process concludes, the model can be used to make predictions in the following way. From the input sequence given as training input to the model, select a random shorter sequence. This selected sequence is fed as input and receives a note as output. For 100 iterations, the input sequence is shifted, adding the predicted note to its end and removing the start note, receiving another predicted note. At the end of the loop, there is a sequence of 100 generated notes which can be transformed into a midi file and played as a new, AI-composed song.

Another approach, inspired by [19], is to combine the VAE with the strengths of Transformers, which are generally thought of as the potent successors of RNNs. The primary advantage of Transformers over LSTMs lies in the attention mechanism, which allows them to better capture long-term dependencies in the data. This is very beneficial in music generation, where rhythm, harmony, and other musical attributes that define a genre are often spanned across longer sequences. Research in various areas of sequence data processing, including natural language processing and music generation, has also shown that Transformers models often outperform LSTM-based models in terms of performance

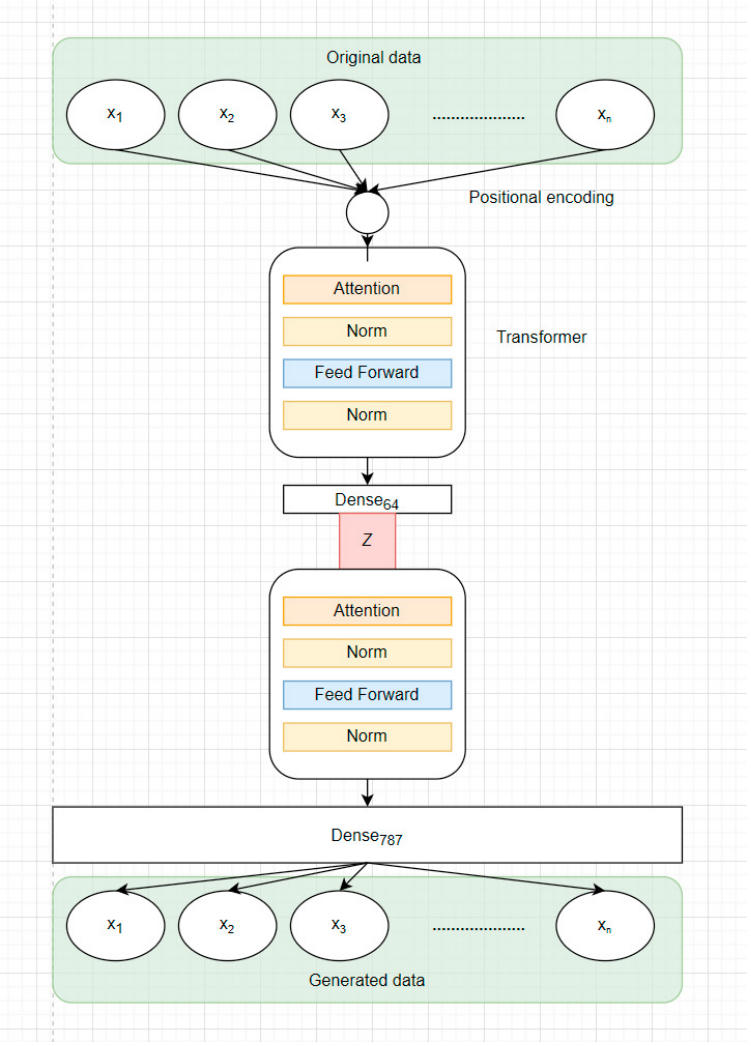

Fig. 3. Architecture of VAE that uses Transformer to encode/decode data.

and quality. Concerning pre and post-processing steps required by the model to work with appropriate data, they are similar in both cases (for the simple and augmented VAEs). Consistent differences appear inside the model, as the LSTM cells inside the encoder and the decoder are replaced with Transformer blocks. The encoder follows the standard transformer architecture, with multi-head self-attention and position-wise feed-forward networks, as for regularization there are LayerNormalization and Dropout Layers. The decoder takes sampled points in the latent space as input, applies the transformer encoding, and outputs the reconstructed notes through a Time Distributed layer.

The two simple yet innovative hybrid models: VAE-LSTM and VAE-Transformer distinguish themselves from the existing literature by effectively combining the strengths of Variational Autoencoders (VAEs) with Long Short-Term Memory networks (LSTMs) and Transformers. While traditional VAEs are adept at capturing latent representations, they often lack the sequential learning capabilities required for generating coherent musical sequences. Conversely, LSTMs and Transformers excel at handling sequential data but may struggle with the complexity of learning intricate latent structures in music. By integrating VAEs with LSTMs, the VAE-LSTM model leverages the VAE's ability to learn rich latent spaces and the LSTM's prowess in sequence modeling, thus enhancing the model's capacity to generate nuanced and contextually consistent musical pieces. Similarly, the VAE-Transformer model capitalizes on the Transformer's attention mechanism, allowing it to capture long-range dependencies and intricate patterns in music more effectively than traditional sequence models. These hybrid models, though simple in design, serve to demonstrate

the potential of combining different architectural strengths to better incorporate and generate the subtle complexities inherent in various musical genres.

Table 1. VAE-LSTM vs VAE-Transformer final losses

Model	Total loss	Reconstruction loss	KL
VAE-LSTM	4.8046	4.8003	2.7044e-05
VAE-Transformer	4.7117	4.6439	0.0677

4. Evaluation

Evaluating AI-generated music is a challenging task, coming from the fact that musical appreciation is highly subjective and nuanced. In what concerns computational evaluation, music is a complex, multi-dimensional medium that encompasses many aspects like rhythm, timbre, and dynamics, which cannot be captured in a single quantitative metric. Some researchers have built specific metrics that measure the re-creation fidelity of the generated music like chroma similarity, grooving similarity, or instrumentation similarity, all of these found in [19].

However, the work in this paper is limited to the classical metrics for Variational Autoencoders: the reconstruction, KL-divergence, and total loss. The reconstruction loss models how well the model can recreate the input data after encoding it into the latent space and decoding it back into the original space. A high reconstruction accuracy means that the generated music closely resembles the training data in terms of many musical attributes like rhythm, and note sequences. Improving only this loss could easily lead to overfitting and generating music that lacks creativity. This is where the KL divergence intervenes. This metric measures the difference between the learned latent distribution and the normal distribution. Now, the model is encouraged to maintain a structured latent space that allows for more diversity and creativity. The total loss is the weighted sum of these 2 losses and tries to keep a balance between the 2: focusing on the reconstruction leads to overfitting and lack of creativity while focusing on the KL leads to diverse but low-quality output. Final results 1 show that the models are competitive with each other, but also leave room for improvements.

As can be easily seen from the training steps, this process is arduous, working with the notes directly from the memory and with the notes being memorized in a generator. There is also a similarity regarding the values for the loss function in the VAE model, where the total loss drops from 10 to around 4.8 during the first epoch, and it oscillates around this value for the next epochs, and in the TransformerVAE model, where the total loss jumps from 7 to 4.8 in the first epoch, then it starts to slowly decrease again. The KL-loss in the latter model is however a few times higher than the former, perhaps showing that this model is learning a better representation of the data. The time needed for one epoch to complete the training phase is also relatively large, reaching approximately 10,000 seconds.

In tandem with our exploration of hybrid models for music generation, our endeavor also delved into genre detection, a crucial aspect in assessing the fidelity of generated music. Careful consideration was given to selecting and analyzing models that could achieve optimal performance within the scope of our study. This involved evaluating a spectrum of deep learning and machine learning algorithms, including Multilayer Perceptrons (MLP), Convolutional Neural Networks (CNN), k-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest, XGBoost, ensemble methods, among others. While the generated songs might not always convincingly mimic human-produced music to listeners, they consistently maintain the distinct characteristics of the training dataset's genre, in this case, Jazz. Despite any discernible artificiality in the audio output, these compositions effectively encapsulate the compositional structure and unique elements emblematic of the targeted genre 2. Importantly, these genre-specific traits are discernible not only to human listeners but also to other machines, particularly the models engaged in genre classification tasks [20]. This underscores the fidelity of our models in capturing genre-specific nuances, despite any perceptible divergence from human-produced music.

An interesting phenomenon appears when training both VAEs, due to the nature of their underlying probabilistic models. They can sometimes predict a sequence of repetitive notes when performing the maximum likelihood estimate, a sequence that changes along with the initial seed. The causes for this "laziness" of the models might be the

Table 2. Generated song passed through DL model - still Jazz.

probability for **blues**	8.52%
probability for **classical**	9.94%
probability for **country**	8.52%
probability for **disco**	8.52%
probability for **hip-hop**	8.66%
probability for **jazz**	17.69%
probability for **metal**	8.52%
probability for **pop**	8.53%
probability for **reggae**	12.56%
probability for **rock**	8.52%

lack of diversity in the dataset, the overfitting of the model into the same high-occurring note, or the poor choice of architecture and hyperparameters. However, when looking at the probability distribution of the predictions, there were several other notes in the same exponential range, indicating that all the other notes could be a fit good enough for the prediction. To mitigate this issue, a "temperature" parameter was introduced, replacing the "ArgMax" function that selected the final prediction. This parameter effectively controls the sharpness of the probability distribution, at high values outputting a more uniform distribution that encourages diversity, and at low values outputting a more peaked distribution that makes the model more confident in its reconstructed predictions. By carefully tuning this hyperparameter, more pleasing and diverse results can be achieved. A few samples generated by both of these models can be downloaded from the notebooks' repository.

While the existing literature in the domain of music generation often presents comprehensive frameworks incorporating various components to capture the complexity of musical compositions, our hybrid models, despite their simplicity, offer a compelling perspective. It's acknowledged that music encompasses a vast array of intricacies, from melody and harmony to rhythm and structure, demanding sophisticated modeling techniques. However, our hybrid models highlight a different approach by emphasizing the inherent strengths of individual components. While a single model might excel in certain aspects, it may fall short in others.

By combining distinct models like VAEs, LSTMs, and Transformers, we leverage the unique capabilities of each to compensate for the limitations of the others. In doing so, our approach underscores the potential for significant advancements in music generation by synergizing the powerful features of different models, thus paving the way for more nuanced and expressive musical compositions.

5. Conclusions

This paper presented an extended work on the complex tasks of music generation. The models were picked to mimic the way humans produce music and to add a little bit of creativity to it. The complete set of analyses, methodologies, and results described in the paper have been documented and are available in a structured collection of Jupyter notebooks. This collection can be accessed at the associated GitHub repository, providing an in-depth overview of the research and facilitating replication or further exploration of the work.

In future research, there exist promising avenues for refining the task of music generation to attain a closer resemblance to human-produced music and to evoke artistic sensibilities. This could involve detaching the generated music from the strict dependency on input data, thereby fostering the creation of compositions with more independent creative expression.

Moreover, developing bespoke architectures tailored to specific musical attributes could yield compositions imbued with deeper emotional resonance and thematic coherence. Additionally, embracing the concept of "one-shot transfer" presents an intriguing prospect, wherein both the stylistic essence and content of music are distilled from a single audio track, potentially broadening the scope of creative possibilities. Furthermore, to enhance the evaluation process and provide more robust insights into the quality of generated music, incorporating a diverse array of mathematical

and statistical measures could offer a comprehensive framework for assessment and also with improved hardware resources, models could undergo more extensive training epochs, leveraging the observed constant improvements in loss rates. This multifaceted approach aims to not only advance the technical prowess of AI-driven music generation but also to infuse it with the richness of human emotion and artistic expression.

References

[1] Bogdan-Antonio, Crețu, Alexandru, Vranceanu, Andi, Cojocariu, Cristian, Simionescu, and Adrian, Iftene (2022) "Music Generation using Neural Nets", In International Conference on INnovations in Intelligent SysTems and Applications (IEEE INISTA 2022), Biarritz, France, August 8-10, 1–6.

[2] Magar, Anand, Acharya, Adarsh, Bothe, Sakshi, Bihani, Harsh, and Desai, Tejas (2023) "Automatic Music Generation", *International Journal for Research in Applied Science and Engineering Technology*, **11**: 1945–1950. https://doi.org/10.22214/ijraset.2023.56835

[3] Lu, Gening (2023) "Deep Learning-Based Music Generation", *Applied and Computational Engineering*, **8**, 366–379. https://doi.org/10.54254/2755-2721/8/20230188

[4] Shiromani, Ruchir, Mittal, Tanisha, Mishra, Anju, and Kapoor, Anjali (2023) "Analysis of Automated Music Generation Systems Using RNN Generators", *In Hasteer, N., McLoone, S., Khari, M., Sharma, P. (eds) Decision Intelligence Solutions, InCITe 2023, Lecture Notes in Electrical Engineering, Springer, Singapore*, **1080**. https://doi.org/10.1007/978-981-99-5994-5_22

[5] Dhariwal, Prafulla, Jun, Heewoo, Payne, Christine, Kim, Jong Wook, Radford, Alec, and Sutskever, Ilya (2020) "Jukebox: A Generative Model for Music", 2020, *arXiv:2005.00341v1*. https://arxiv.org/pdf/2005.00341.pdf

[6] Cifka, Ondřej, Simsekli, Umut, and Richard, Gaël (2020) "Groove2Groove: One-Shot Music Style Transfer with Supervision from Synthetic Data", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **28**: 2638–2650, *doi: 10.1109/TASLP.2020.3019642*. https://ieeexplore.ieee.org/document/9178446

[7] Shih, Yi-Jen, Wu, Shih-Lun, Zalkow, Frank, Müller, Meinard, and Yang, Yi-Hsuan (2022) "Theme Transformer: Symbolic Music Generation with Theme-Conditioned Transformer", 2022. https://arxiv.org/abs/2111.04093

[8] Spotify for Developers. https://developer.spotify.com/

[9] SoundHound Music: Discover, Search, and Play Any Song by Using Just Your Voice. https://www.soundhound.com/soundhound

[10] Soundraw: Welcome to the AI Music Generation. https://soundraw.io/

[11] A Beginner's Guide to MIDI: What is it? How does it work? https://musicianshq.com/a-beginners-guide-to-midi/

[12] Defferrard, Michael, Benzi, Kirell, Vandergheynst, Pierre, and Bresson, Xavier (2017) "FMA: A Dataset for Music Analysis", *arXiv:1612.01840v3*. https://arxiv.org/pdf/1612.01840.pdf

[13] GTZAN Dataset - Music Genre Classification. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[14] FMA: A Dataset For Music Analysis. https://github.com/mdeff/fma

[15] Million Song Dataset. http://millionsongdataset.com/pages/getting-dataset/

[16] Spotify Dataset. https://www.kaggle.com/datasets/pricoptudor/spotify-dataset

[17] Jazz ML-ready MIDI. https://www.kaggle.com/datasets/saikayala/jazz-ml-ready-midi?select=Jazz-midi.csv

[18] Spotify Image Dataset. https://www.kaggle.com/datasets/pricoptudor/spotify-image-dataset

[19] Shih-Lun, Wu, and Yi-Hsuan, Yang (2023) "MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer With One Transformer VAE," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, **31**, 1953–1967, *doi: 10.1109/TASLP.2023.3270726*. https://ieeexplore.ieee.org/document/10130842

[20] Tudor-Constantin, Pricop, and Adrian, Iftene (2024) "Enhancing Music Genre Classification with Artificial Intelligence", In 16th International KES Conference on Intelligent Decision Technologies (IDT-24), Santa Cruz, Madeira, Portugal, 19-21 June 2024.