

Comparative Analysis of cVAE, cGAN, and LSTM Models for Music Generation Based on Weather Conditions

Nedim Karavdić^{1*}, Bećir Isaković²

¹Department of Information Technology, International Burch University, Sarajevo, Bosnia and Herzegovina
(nedim.karavdic@stu.ibu.edu.ba, becir.isakovic@ibu.edu.ba)

Abstract – Generative Artificial Intelligence is gaining popularity every day and uses contextual data to personalize and enhance user experiences. This paper explores music generation that is conditioned on weather. It influences musical compositions by connecting MIDI music data with corresponding weather attributes, for example sunny and cloudy weather. In this paper, three generative models are compared for the task of weather based music generation; Conditional Variational Autoencoder (cVAE), a Conditional Generative Adversarial Network (cGAN) and Long Short-Term Memory (LSTM) network. Mentioned models are implemented and trained on combination of large MIDI corpus with historical weather information. Performance of models in this paper are evaluated using metrics that capture musical diversity and quality, such as pitch range, unique pitches and pitch variance, and for fidelity to real data is measured by mean squared error and KL divergence. Results of the paper showed that the cVAE produced the most diverse music for music that is context sensitive. Using cVAE in this approach helped the model in achieving the widest pitch range and variety of notes with low error. Runner ups in this comparison was cGAN that generated relevant music, but with slightly less diversity while LSTM showed higher error and inability to integrate the weather context. Contributions to this work is made of the first comparative analysis of these architectures for weather based music generation, exploration of their strength and limits and evidence that environmental data can be used to influence AI generated music. Results showed the relevance of blending weather data with music generation which can serve as a foundation for applications such as adaptive game soundtracks, mood based music therapy or dynamic background music that is being generated according to the user's environment.

Keywords – Music Generation, cVAE, cGAN, LSTM, MIDI.

I. INTRODUCTION

Music generation based on weather conditions combines generative artificial intelligence and environmental conditioning. Music composition, which stands for creative human task, can now be modeled by machine learning systems trained on datasets of songs. In this case contextual inputs provide influence to the generative process and allow music to reflect environmental scenarios. This paper explores how three different generative architectures (cVAE, cGAN and LSTM) can produce weather conditioned music, specifically scenarios of sunny and cloudy weather. Goal of this paper is to evaluate each model's capability to learn and generate MIDI sequences that are relevant to weather states. Previous research explored generative models for context aware music, for example music driven by emotional tags or similar scenarios. Besides that, there is a need to analyze and compare different generative models under specific contextual settings to identify effectiveness of the models in handling that specific conditioning. The motivation for using weather as context in this case comes from increasing interest in AI generated art for improving user experiences in entertainment, gaming and therapy industries. An example of this would be an adaptive game soundtrack that shifts to brighter melody on a sunny day or calmer tone when it is cloudy outside. Even though there is progress in music generation, there is a gap in understanding how different models perform under environmental conditions. This paper aims to bridge the gap by providing a comparison of cVAE, cGAN and LSTM models for weather

conditioned music generation. Authors of this paper focus on quantitative metrics that are: pitch range, note diversity, accuracy while having computational considerations in mind. In summary, this paper provides an analysis of three generative models for music generation based on weather. The following sections describe relevant details of the dataset and modeling approach, experimental results, discussion and conclusion with insights, limitations and future directions

II. MATERIALS AND METHOD

A. Data and preprocessing

In order to train and evaluate models, authors of this paper paired a dataset of MIDI music and weather information. Two data sources that were used are the "Lakh MIDI Dataset" [13] and a "Historical Weather Dataset" [14]. The Lakh MIDI Dataset is a large collection of approximately 170.000 MIDI files containing multiplied genres and styles of music. Each MIDI file consists of one or multiple instruments which can be represented as sequences of musical notes. The dataset was tokenized and truncated to on the order of millions of note events that served as training samples. For the weather data, authors of this paper used a historical hourly weather dataset that was sourced from Kaggle [14]. Weather dataset provides meteorological data such as temperature, humidity, wind speed and textual description of weather for each hour across multiple locations. Authors of this paper chose this dataset as the source of contextual features to condition generation of music. To be more specific, the authors of this paper focused

on the weather_description field to derive a simple categorical condition for music that will identify “sunny” or “cloudy” weather. The weather data was cleaned by handling missing values and mapped to binary labels. An example of this is sunny condition being encoded as 1 while cloudy condition would be mapped as 0. All other weather states were mapped into these categories for the purpose of the study, such as clear or sunny skies were the result of sunny weather, overcast and rainy weather were results of cloudy weather. When it comes to MIDI processing, the MIDI files were parsed using pretty_midi library to extract a sequence of note events and associated musical attributes that are note, pitch, timing, tempo and key. Each note event, which is characterized by its pitch value in MIDI representation, was tokenized into a numerical format. Authors of this paper focused on pitch sequence for generation. Other musical features like timing and velocity could be incorporated, but for simplicity the current approach treats the sequence of pitches as the primary data [6], [12]. All note pitch values were already in the MIDI scale of 0 to 172 and that range was used as-is as the normal input range for the models. These values can be normalized or standardized if required by a model, but in the current implementation the integer values were directly used with embedding or one hot encodings in the models since the 0-127 range was treated effectively at all times. Regarding the weather data alignment, because the music and weather datasets do not share any common timeline or field, authors of this paper assigned weather conditions to musical sequences synthetically rather than matching the values by timestamps or unrelated fields. First, the weather descriptions were converted into a binary label: entries that contain terms such as “sun” or “clear” were labeled as sunny (1) while other conditions hinting to cloudy weather were labeled as “cloudy” (0). Given that MIDI sequences have no associations with weather data, authors of these papers paired each note sequence with a weather label in a controlled manner. In this experiment, the weather label was randomly applied to all MIDI sequences with equal probability. This approach helped the research have an overall training set equipped with both conditions in equal proportion. Each training sample consists of a sequence of MIDI notes and a single binary weather label. In order to maintain consistency across training, in experiment, a fixed sequence length (N notes) was used. If a MIDI track was longer than N notes, it was either truncated to N or in LSTM, broken into multiple overlapping segments of length N, but if the track was shorter than N, the sequence was padded with zeros until it reached the required length. During model training, the weather label for each sequence was equipped with additional input features or conditioning signals. The prepared dataset was split training and validation sets with 80/20 split. 80% of paired sequences were used for model training and 20% were kept out for validation. Even though the split was done randomly graded by weather conditions to make sure that both cloudy and sunny labels are represented proportionally in both sets.

B. Conditional Variational Autoencoder (cVAE)

The first generative model in this research is a conditional Variational Autoencoder (cVAE), a latent variable model that learns a probability distribution over musical sequences conditioned on the weather context. The cVAE architecture in this research is designed to learn a representation of MIDI sequences while using the weather label as a conditioning variable so the latent space captures weather dependent

variations in the music. cVAE consists of an encoder, latent space sampler and a decoder with the overall training objective combining a reconstruction term and regularization term. In cVAE, the encoder network takes the music sequence as an input with its weather condition label and maps it to a latent representation. The note sequence is first passed through an embedding or a stack of layers to produce an encoded feature representation. The weather label is appended to this representation so that the encoder is aware of the context. After that, the encoder produces two output vectors: the mean vector and variance vector; they together define the distribution in the latent space to which the input sequence corresponds to. A latent vector is sampled from the distribution using the encoder’s output. The decoder network takes the sampled latent vector with a weather label as input and attempts to reconstruct the original music sequence that corresponds to the given latent code. The weather condition is provided to the decoder to make sure that generation still depends on the context. In training, the decoder’s output is compared to the original input sequence. It is important to note that the cVAE is optimized using a combination of reconstruction loss and Kullback-Leibler KL divergence loss. The reconstruction loss encourages the decoder to accurately reproduce the input notes. The KL divergence regularizes the learned latent distribution towards a prior distribution which prevents the encoder from memorizing the data and promotes continuous latent space that can generalize. The total loss is a weighted factor that balances reconstruction fidelity and how smooth the latent space is. cVAE model learns to generate music sequences by minimizing this loss. In summary, the cVAE provides a latent generative approach in which both encoding and decoding are influenced by the weather context, in this case. At generation time, one can sample from the prior and combine it with a chosen weather label which is passed through decoder to obtain a new music sequence that reflects the given weather in the label. This model is expected to capture a wide variety of music possibilities.

C. Conditional Generative Adversarial Network (cGAN)

The second model in this research is a conditional Generative Adversarial Network (cGAN) and it consists of two competing neural networks, a generator and a discriminator which are trained in opposition. In a conditional GAN, both networks are conditioned on an external variable, which is in this research the weather label. This helps the generator learn to produce music sequences aligned with a given weather condition and the discriminator learns to distinguish between fake and real music, and also whether the sequence is matched with the provided condition. The generator in cGAN receives a random input vector and a weather condition label as input. The weather label can be incorporated by concatenating a one hot or binary representation label to the noise vector. After that, the generator network maps combined input through a series of layers to produce an output sequence of MIDI notes. An example of this would be the generator using a series of fully connected layers to up sample the noise vector into a structured sequence output. In this research, the generator produces a sequence of length N in which each element is a predicted MIDI pitch value. Some architectural features such as ReLU activations and batch normalization are set in the generator to assist training. Rectified Linear Unit (ReLU) helps with gradient flow while batch normalization stabilizes the learning

by normalizing layer inputs. This has been shown to speed up GAN training and mode collapse reduction. The final layer of the generator produces outputs in the range that correspond to valid MIDI note values. The trained generator should be capable of taking a noise sample and specified weather label, and outputting a sequence of notes that could represent a musical phrase with characteristics of that specified weather label. The discriminator in cGAN is a binary classifier that takes a pair as input, in the case of this research is music note sequence and a weather label. It receives either a real sequence from the training data with the correct label or fake sequence produced by the generator. The discriminator network is responsible for processing the note sequence with the weather label and outputs a probability or score that indicates whether the sequence is real and matches the condition or fake. The discriminator is trained to assign high confidence to real conditioned pairs and low confidence to generator's outputs. In this research, by conditioning on weather, the discriminator also learns to validate if the music is appropriate for the given weather context. When it comes to adversarial training, the cGAN training also follows the standard min-max game optimization. The generator is trained to "fool" the discriminator, to be specific, to generate sequences that the discriminator classifies as real for the given condition while the discriminator is simultaneously trained to distinguish the generator's fakes from true data. Once training is completed, the generator can be used independently to create new music sequences. It takes a random noise vector and chosen weather label as an input and produces a musical sequence that reflects the characteristic that is learned during training, an example of it in this research would be the "sunny" label producing music that sounds more upbeat or bright compared to "cloudy" label. The adversarial setup might be unstable or sensitive to hyperparameters, for example learning rates, batch sizes and frequency with which the generator and discriminator are updated.

D. Long Short-Term Memory (LSTM) Network

The third model that is evaluated in this research is a recurrent neural network based on a Long Short-Term Memory (LSTM) architecture. LSTM is chosen for its strength in sequence modelling [12]. Unlike the cVAE and cGAN which incorporate a latent variable or adversarial framework, the LSTM provides more direct approach, LSTM learns to predict the next notes in a sequence given the past notes with an additional input (which is the weather context in this case) for the next step. The LSTM model in this research is a single stream sequence predictor that outputs one note at a time and the note is conditioned on both previous note and weather label [3]. Input representation in LSTM consists of the current note (or the previous one) and the weather condition. In this research, the sequence of notes is fed into the LSTM one token at a time. Binary weather feature is appended to the note's input vector so that the model receives a compound input at each step. An example of this would be the current note represented by a one-hot vector of length 128 for pitch class) or an embedding, the weather label can be appended as an additional feature dimension. This approach helps LSTM to know whether the sequence it is generating is supposed to correspond to a sunny or cloudy. LSTM layer is the core of the model with a certain number of hidden units. The LSTM's gating mechanism enables maintenance and update of an internal state that carries information about the sequence

history. As the LSTM processes the sequence of notes, it learns to memorize or forget musical patterns and potentially adjust them based on the weather context. For example, if the certain note transitions or rhythms were more common under "sunny" training samples, the LSTM can learn to favor those transitions when the weather input indicates sunny. The hidden state after each time step captures the information of all past notes seen so far because the weather label does not change during a given sequence, it acts as a guiding bias on the hidden state dynamics throughout the sequence. LSTM also consists of an output layer, the LSTM's output at each time step is passed to a final dense layer that produces a probability distribution over the next note. In this research, authors of the paper applied a softmax activation on this output which yielded a 128-dimensional output where each value represents the probability of a particular MIDI pitch being the note in the sequence. During training, this prediction is compared to the actual next note in the training sequence and the error is used to adjust the model weights via backpropagation through time. Authors trained the LSTM to maximize the likelihood of the correct sequence of notes based on weather conditions. During generation time, the trained LSTM can be used to generate new music sequences by iterative prediction. It starts with an initial primer and a chosen weather condition, then repeatedly samples the next note from the softmax output and feeds it back as input to predict subsequent notes. Since the weather label remains fixed during generation, the LSTM's patterns of note prediction will be biased according to that label, for example if during "cloudy" conditions, if the model learned that cloudy sequences have a narrower pitch range, it may reflect it in the output sequence. Compared to the cVAE and cGAN which generate an entire sequence in one go, the LSTM generates sequence step by step and can maintain long-term structure with its memory cells. But, as a likelihood-biased model, LSTM does not explicitly enforce the realism of sequences beyond what is learned from training data and it might be more prone to exposing the conditional weakness, for example if the model has difficulty integrating weather signals, it might ignore it and generate a generic sequence.

E. Evaluation Metrics

In order to assess the performance of each model and compare their outputs, authors of this paper used several evaluation metrics that put strong focus on musical diversity and fidelity to real data patterns. For evaluation of musical diversity of generated sequences, authors of this paper use pitch range and unique pitches metrics. The difference between the highest and lowest pitch in a generated sequence defines pitch range. If a model covers a broad span of notes that are correlated with richer melodic content, it results in a higher pitch range. When it comes to unique pitches, they count how many distinct pitch values are present in the output that is generated. Less repetition and more variety in the melody means there is a large number of unique pitches. When these metrics are combined together, they provide insights into how diverse the model's compositions are. If a model repeats a small set of notes it would result in a low pitch range and low unique pitch count, while a model that is capable of producing varied melodies would score higher. Mean Squared Error (MSE) is a statistical measure of fidelity that is computed between generated sequences and real sequences. In this research, the MSE was used to measure the average squared difference between the distribution of generated notes and the

distribution of notes in the real validation data for each condition, in this case weather condition. Lower MSE shows that the model’s output is closer to the actual data patterns. In this research, MSE was calculated separately for sequences conditioned on “sunny” and “cloudy” to see how accurately each model captures the characteristics of each weather subset. For example if sunny real music tends to have specific pitch frequencies, and the model’s sunny outputs deviates, the MSE would be high. Since the LSTM can be evaluated in a supervised prediction manner, MSE can also be interpreted as the error in predicting the next note. In the comparisons of this research, a notably higher MSE means the model struggled to reproduce realistic sequences under the given condition. Kullback-Leibler divergence (KL) is used to quantify the difference between two probability distributions. Authors of this research applied KL divergence to compare the distribution of pitch usage in generated music against real music. For each condition, authors derived the probability distribution over pitches from a large set of generated sequences from the real dataset. If a pitch distribution of generated music is similar to real music, it results in smaller KL divergence. That hints that the model was able to capture overall style or even note frequency characteristics of the training data. KL also complements MSE since it is focusing on distributional similarity rather than pointwise error. Using these metrics, authors of this research compare the models in terms of how diverse their outputs are and how accurate they reflect the characteristics of music that is found in the training data. In the next section, authors will present the results for each model across these metrics with analysis of the generated musical sequences.

III. RESULTS

After training cVAE, cGAN and LSTM models on the conditioned music dataset, authors were able to generate a large number of music sequences from each model for both weather conditions, which are evaluated using the defined metrics. Quantitative results can be seen in the Table 1. Speaking of **musical diversity**, the cVAE model was able to achieve the highest pitch range (approximately 0.96) among the researched models. This result shows that cVAE’s generated songs spanned almost the full range of notes seen in the data. cVAE also had the largest pitch standard deviation (0.15) and the greatest number of unique pitches, which is around 1.97 million distinct note events. These results shows that the cVAE’s outputs were the most diverse. In practical examples this means that the listeners would find pieces generated by cVAE to have varied melodies and more jumps in pitch and richer harmonic content. The cGAN also produced diverse music but with a bit lesser extent. cGAN performed with a pitch range of 0.91 and unique pitch count of approximately 1.92 million.

Table 1. Quantitative Results

Metric	cVAE	cGAN	LSTM
Pitch Range	0.96	0.91	0.89
Pitch Std Dev	0.15	0.11	0.13
Unique Pitches	1,973,829	1,923,194	1,890,737
MSE (Sunny)	0.0385	0.0401	39.51
MSE (Cloudy)	0.0423	0.0437	39.51
KL Divergence (Sunny)	0.0248	0.0256	0.0248

KL Divergence (Cloudy)	0.0263	0.0269	0.0248
------------------------	--------	--------	--------

These results hint that the cGAN outputs still varied, but the model might stick closer to the common pattern in the training data. The LSTM showed the lowest pitch range (0.89) and unique pitch count of approximately 1.89 million. The LSTM’s pitch standard deviation was 0.13 which was greater than cGAN’s which implies some variability, but not necessarily new notes. Overall, from diversity point of view it can be seen that the cVAE’s latent space sampling might be better choice for exploration while the GAN and LSTM tended to generate limited distribution of notes. In terms of **fidelity and accuracy**, MSE showed sharp contrast. Both the cVAE and cGAN achieved low MSE for sunny and cloudy conditions which resulted in generated sequences being closer to real sequence patterns. The LSTM’s MSE was high for both sunny and cloudy conditions. High MSE in this context suggests that the LSTM’s predicted note sequences deviated from the actual music sequences in the validation set. There are several reasons for this scenario. One of the scenarios is that the LSTM struggled with external conditioning and might not have learned to use the weather input, so it produced notes that did not align with the output that is expected. Another reason for this might be that LSTM in supervised evaluation frequently guessed wrong pitches and led to large squared errors. Final factor would be that the LSTM overfit on a certain pattern and failed to generalize or might have learned to output a generic sequence which is different compared to the real sequence. In contrast, the cVAE and cGAN benefited from the reconstruction training objective (cVAE) or adversarial feedback (cGAN) in terms of keeping their outputs mirroring real data distributions. Regarding the fidelity and accuracy, the conclusion is that the LSTM model in this configuration and environment was not effective in accurately reproducing the weather based musical sequences. Latent Space Visualization for cVAE can be seen in Figure 1 and for cGAN in Figure 2.

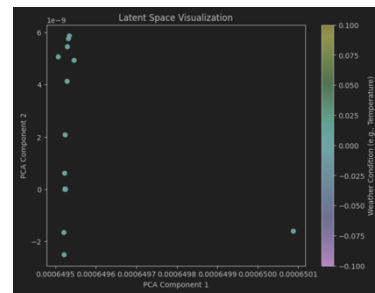


Fig. 1 Latent Space Visualization for cVAE

In the PCA scatter plot of the cVAE’s latent embeddings there is a visible separation between clusters that correspond to different weather labels. Sequences labeled “sunny” tend to group in one region of the 2D latent space while “cloudy” sequences cluster in another region.

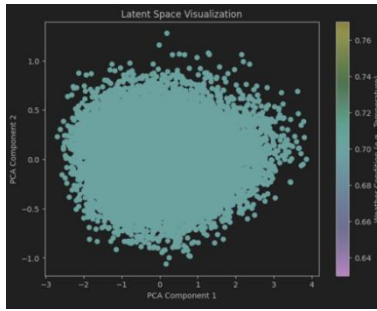


Fig. 2 Latent Space Visualization for cGAN

For the cGAN model, the latent space visualization is less clearly separated by weather labels. This is because the cGAN's latent noise distribution is random and independent of the label. The model is not trained to organize noise points by condition since the label is fed separately to the generator. Unlike the cVAE, the cGAN's latent space doesn't exhibit clear clustering by weather, the latent points remain largely interspersed which is expected because cGAN generates from unstructured noise. Beyond the numbers, authors of this paper also examined sample outputs from each model for each weather condition to assess how well the weather context was reflected in the music. Speaking about cVAE outputs, the cVAE-generated sequences showed that musical notes varied. For sunny conditions some generated samples had an upbeat character and tended to include higher register notes and major intervals. This resulted in a bright and cheerful note. Same model for cloudy conditions produced lower pitched and slower melodies. It is important to mention that these differences were not drastic, for example it did not result in a genre change, but they showed that the cVAE did pick up on contextual cues from the weather label. The sequences had a good amount of variation even when generating multiple samples with the same weather label. The cVAE could produce different melodies, this is mostly done thanks to the sampling of the latent space. Most importantly for cVAE outputs, the notes followed each other in sensible ways. When it comes to cGAN outputs, the cGAN's music outputs were plausible as short musical phrases. They resembled actual snippets of melodies from the training data. During the research, authors noticed that cGAN generation had differences which were less pronounced than they are in the cVAE. Many cGAN sequences tended to stick to mid-range pitches and melodic progressions regardless of their condition. Authors also noted that during the music test, there were notable shifts in the music, for example in cloudy conditions cGAN might go a bit more on minor intervals, and under sunny weather it might resolve to consonant intervals more frequently. The cGAN did not produce extreme outliers, its music almost always sounded legitimate but it lacked some of the surprise or range that the cVAE had. Finally, LSTM outputs were repetitive and less distinctly influenced by the weather condition. Many sequences from the LSTM showed a tendency to oscillate between a limited set of pitches. Example of this is output staying within a narrow range and reusing few notes with less diversity. LSTM would occasionally produce a sequence that starts promising, after which it is stuck in a loop.

IV. DISCUSSION

This result shows that cVAE's generated songs spanned almost the full range of notes seen in the data. These results show that the cVAE's outputs were the most diverse. In

practical examples, this means that the listeners would find pieces generated by cVAE to have varied melodies and more jumps in pitch and richer harmonic content. The cGAN also produced diverse music but to a slightly lesser extent. These results hint that the cGAN outputs still varied, but the model might stick closer to the common pattern in the training data. The LSTM's pitch standard deviation was 0.13, which was greater than cGAN's, implying some variability, but not necessarily new notes. Overall, from a diversity point of view, it can be seen that the cVAE's latent space sampling might be a better choice for exploration, while the GAN and LSTM tended to generate a limited distribution of notes. In terms of fidelity and accuracy, MSE showed a sharp contrast. Both the cVAE and cGAN achieved low MSE for sunny and cloudy conditions, which resulted in generated sequences being closer to real sequence patterns. High MSE in this context suggests that the LSTM's predicted note sequences deviated from the actual music sequences in the validation set. There are several reasons for this scenario. One of the scenarios is that the LSTM struggled with external conditioning and might not have learned to use the weather input, so it produced notes that did not align with the output that is expected. Another reason for this might be that LSTM in supervised evaluation frequently guessed wrong pitches and led to large squared errors. A final factor would be that the LSTM overfit on a certain pattern and failed to generalize or might have learned to output a generic sequence which is different compared to the real sequence. In contrast, the cVAE and cGAN benefited from the reconstruction training objective (cVAE) or adversarial feedback (cGAN) in terms of keeping their outputs mirroring real data distributions. Regarding fidelity and accuracy, the conclusion is that the LSTM model in this configuration and environment was not effective in accurately reproducing the weather-based musical sequences. The cGAN's latent space visualization is less clearly separated by weather labels because the cGAN's latent noise distribution is random and independent of the label. The model is not trained to organize noise points by condition since the label is fed separately to the generator. Unlike the cVAE, the cGAN's latent space doesn't exhibit clear clustering by weather, which is expected because cGAN generates from unstructured noise. Regarding the cVAE outputs, the differences in generated music for different weather conditions were not drastic (e.g., they did not result in a genre change), but they showed that the cVAE did pick up on contextual cues from the weather label. The cVAE could produce different melodies; this is mostly done thanks to the sampling of the latent space. The cGAN outputs lacked some of the surprise or range that the cVAE had. It is worth noting that LSTMs can sometimes be improved with techniques like temperature sampling or adding attention mechanisms to better incorporate context; however, in the configuration of this research, it definitely underperformed in expressing weather conditions in music. In summary, in the results of this research it is revealed that the cVAE model offers the best performance overall for generating weather-based music with the highest diversity and low error rates. The cGAN model was also shown to be effective with highly realistic music and moderate diversity. The LSTM model was capable of learning sequences, but showed obvious limitations in this contextual generation task, which is backed up by its lower diversity and high errors. LSTM struggled to produce outputs as rich or accurate compared to the two other models. Besides giving out metrics, these results show the importance of selecting

appropriate generative architecture based on the outcome that is desired. In the context of this research, if creativity and exploration are values, it would be a great choice to start with cVAE, while cGAN is more suitable for more realistic music; otherwise, LSTMs may require more improvement and may not be suitable alone for this task.

V. CONCLUSION

In this work, a comparison between three generative AI models (cVAE, cGAN, LSTM) was conducted for the task of music generation based on weather conditions. Authors of this paper leveraged a dataset that combines MIDI music sequences with corresponding weather labels in order to enable models to learn how environmental context can influence musical production. The finding of this research shows that the cVAE offers the highest diversity and fidelity in generated music. cVAE was able to outperform the other models in metrics such as pitch range, pitch variance and number of unique notes. cVAE showed the ability to generate a broad spectrum of notes while cGAN also performed well. It was able to produce context-appropriate sequences with slightly less diversity than the cVAE, but it was excellent in creating realistic outputs and it occasionally missed some of the more extreme variations that was found in the real data. The LSTM model struggled to integrate the weather context effectively, it was able to learn the basic structure of music, but it showed limited diversity and extremely high error when trying to match real sequences. Each model comes with its limitations. Even though the cVAE outperformed other models and relies on a latent space, its challenge is to make sure that it generates sequences that are not only diverse, but also musically coherent over longer time frames. The cGAN has more delicate training in order to avoid issues like instability or mode collapse, although authors of this paper did not observe severe mode collapse in the results. Training cGANs on symbolic music may require hyperparameter tuning and data augmentation, and a GAN in general might need an even more informative conditioning mechanism. The LSTM's limitation was evident in this study, but without additional architecture components, which means that vanilla LSTM might not be the best pick to utilize global context like weather since it focuses on local sequence patterns. Authors of this research concluded that LSTMs alone are not ideal for strongly conditioned generation until they are enhanced with other mechanisms such as conditional training strategies or attention to condition. Mentioned limitations lead to several future work directions. Exploring hybrid architectures could gain a lot of benefits. For example, the combination of cVAE's strength and cGAN might leverage cVAE's latent diversity and cGAN's realism to create even higher-quality music. Hybrid models are often pointed out as a go to choice for music generation in research. Integration of sequential modeling power of LSTM with cGAN might address LSTM's weakness, for example LSTM can ensure temporal coherence while cGAN's discriminator guides the overall output distribution. Another promising direction for future work should be implementation of attention mechanism or transformer architecture in place of, or with the LSTM. Transformers can potentially handle conditional outputs more effectively by attending directly to a context token throughout the sequence generation. From a data perspective, expansion of representation of weather and context could lead to more

nuanced results. In this research, authors reduced weather weather to a binary label for clarity. Future work could include richer sets of weather conditions such as rainy, stormy, snow, etc., or even continuous features like temperature or humidity levels to see if models can musically express this data. Additionally, conducting user studies or listening tests would be valuable to check how listeners perceive the differences in music for sunny and cloudy conditions for each model. Human evaluation can reveal if statistical differences that were measured in this research translate into recognizable emotional or contextual differences in the music. If listeners can tell the difference between generated music in sunny conditions from generated music in cloudy conditions, that would validate the effectiveness of conditioning, otherwise it should be indicated that there is a need for stronger conditioning signals or more expressive models. Having real world applications in mind, the future research might integrate this weather conditioned music generation into a practical system. For example, a mobile application that generates a background soundtrack for your day based on the current weather, or an interactive installation where the music in space changes with the weather in real time. This research could also be useful for real world products such as adaptive gaming equipment or dynamic music systems for spas and wellness centers. Mentioned applications could benefit from further improvements and research on real-time generation and stability in integration in order to avoid any unpleasant musical outputs. In conclusion, this research demonstrated that the combination of weather context and music generation is feasible, but it can also influence the characteristics of the generated music. Among the models that are compared in this work, cVAE and cGAN stood out as effective approaches while LSTM was less successful. Authors provided insights into how each model's architecture has an impact on its ability to weave contextual information into creative output. These results contribute to the broader understanding of context-aware generative art and lay the foundation for more sophisticated systems that bin AI generated music to the environment or user context. Future explorations that combine model strengths and address current limitations will pave the way to richer and more controllable music generation experiences, doesn't matter if they are influenced by the weather or other facets of the world around us.

ACKNOWLEDGMENT

Authors would like to express gratitude towards International Burch University for providing the resources and supportive environment that was necessary for this research.

REFERENCES

- [1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," 2016.
- [2] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," 2017.
- [3] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial Neural Audio Synthesis," 2019.
- [4] Z. Zheng, C. Cai, and Y. Zhang, "Real-Time Intelligent Big Data Processing: Technology, Platform, and Applications," 2019.
- [5] A. N. Navaz and T. Karthikeyan, "Real-Time Data Streaming Algorithms and Processing Technologies: A Survey," 2019.
- [6] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep Learning Techniques for Music Generation," 2020.
- [7] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A Generative Model for Music," 2020.

- [8] W. X. Zhao, Y. Wu, J. Liu, and J. Xu, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," 2023.
- [9] J. Wen and K. M. Ting, "Computational Intelligence Techniques for Music Composition: A Review," 2023.
- [10] Google Cloud, "Enabling Real-Time AI with Streaming Ingestion in Vertex AI," 2022. [Online]. Available: <https://cloud.google.com/blog/products/ai-machine-learning/real-time-ai-with-streaming-ingestion-in-vertex-ai>
- [11] Y. Zhao, X. Wang, and C. Liu, "Domain Adversarial Training on Conditional VAE for Controllable Music Generation," 2022.
- [12] D. Conner, T. Johnson, and K. Lee, "Music Generation Using LSTM Networks for Sequence Modeling," 2022.
- [13] Lakh MIDI Dataset.
- [14] Historical Hourly Weather Data 2012-2017