



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## Music Generation Using Rnn-Lstm

<sup>1</sup>Mrs. K. Sai Jyothsna, <sup>2</sup>Muttineni Mohini Lakshmi, <sup>3</sup>Mallampalli Kamakshi,

<sup>4</sup>Naram Vignesh Kumar, <sup>5</sup>Mohammed Sofiya

<sup>1</sup>Assistant Professor, <sup>2,3,4,5</sup>Students, Department of Computer Science and Engineering  
<sup>1,2,3,4,5</sup>Dhanekula Institute of Engineering & Technology, Vijayawada, India

**Abstract:** The creation of music that is suited to particular genres is now possible thanks to recent developments in deep learning, which successfully capture complex musical patterns. This paper presents a web application built using Django that uses Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) to produce genre-specific compositions. The system uses sequence modeling to generate compositions that fit the musical styles of the six genres it is trained on (Classical, Blues, Jazz, Rock, Pop, and Country) using MIDI datasets. Coherence in the produced music is ensured by the model's efficient learning of melodic transitions, harmonic structures, and rhythmic variations. Evaluations attest to its capacity to produce works that are stylistically authentic while preserving the essential elements of each genre. With the help of the online application's user-friendly interface, users may create AI-powered music with ease. By integrating deep learning with web-based accessibility, this study highlights the potential of AI in music composition, demonstrating its role in automated, genre-specific creativity and expanding possibilities in AI-assisted musical innovation.

**Index Terms** - Django, AI Music Generation, RNN-LSTM, Deep Learning, MIDI, Genre-based Composition, Machine-Generated Music.

### 1. INTRODUCTION

The rise of artificial intelligence in creative domains has transformed the methods of music composition, analysis, and generation. While traditional music composition depends heavily on human skill, intuition, and extensive experience, machine learning models strive to replicate these creative processes through organized data handling. This research delves into the incorporation of deep learning techniques in automated music creation, aiming to develop a web application based on Django that produces compositions specific to various genres. Unlike typical music generation models that frequently yield random or stylistically mismatched sequences, this application guarantees that its compositions comply with distinct musical frameworks, tonal tendencies, and stylistic features representative of each genre. The main aim of this research is to evaluate how effectively deep learning models, especially Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM), can learn and reproduce complex musical patterns across various genres. Given that music is inherently sequential and follows structured dependencies, RNN-based models are particularly appropriate for this task as they can understand long-distance relationships among notes, harmonies, and rhythms. By utilizing MIDI datasets for training the model, it learns the hierarchical links between various musical components, allowing it to create compositions that feel authentic and unified. The application enables users to input various parameters such as genre selection, sequence length, and temperature modifications, ensuring flexibility in the creative process. This research presents several innovative elements that differentiate it from existing projects. Firstly, the application is specifically tailored to create genre-specific compositions, ensuring that the output retains the crucial traits of the chosen style. Additionally, the integration of Django offers a smooth and user-friendly web interface, facilitating real-time interaction and composition generation without the need for extensive technical knowledge. In contrast to conventional standalone music AI models, which often necessitate specialized software and manual configurations, this web-based method makes AI-

driven composition broadly accessible. From an academic perspective, this study adds to the ongoing discourse regarding AI in music generation by merging machine learning, music theory, and web-based implementation into a unified system. The work is divided into several sections such as an overview of AI-powered music generation and its importance, a review of current literature on deep learning applications in music creation, a comprehensive description of the RNN-LSTM framework, dataset processing methodologies, and Django integration, concluding with an assessment of the generated compositions. The study wraps up with reflections on the model's effectiveness, limitations, and prospects for enhancements in future research. This study advances the role of AI in the creative fields and demonstrates how technological progress can enhance artistic expression. By providing an automated but adaptable approach to music creation, the system lays the foundation for future developments in AI-aided creativity, framing machines as partners in the creative process instead of replacements for human composers.

## 2. LITERATURE REVIEW

The application of deep learning in music composition has been extensively investigated, with various models suggested by researchers to improve the quality and coherence of music created by AI. Huo et al. [1] introduced a framework for soundtrack synthesis based on LSTM, showing how recurrent networks can successfully capture long-term dependencies in musical sequences, which enables more organized compositions. Similarly, Sambaragi et al. [2] introduced a simplified approach to music generation, focusing on reducing computational complexity while preserving musical coherence, making AI-driven composition more accessible. Recognizing the limitations of sequential models, Zhang and Liu [3] proposed a CNN-LSTM hybrid architecture, where convolutional layers extract spatial features before LSTMs process sequential dependencies, improving the model's ability to generate more dynamic and expressive music. Beyond individual model architectures, Wang et al. [4] provided a comprehensive review of intelligent music generation systems, examining advancements in deep learning techniques and their impact on automated composition. In contrast, Li [5] focused on chord-based music generation, emphasizing the importance of harmonic structures in AI-composed pieces to ensure musical realism. Extending AI applications to traditional compositions, Li et al. [6] developed a Bi-LSTM deep reinforcement learning model for Xi'an drum music, showcasing how AI can preserve cultural musical styles while introducing automation. Emotion-driven music generation has also gained attention in recent studies. Shi and Vear [7] introduced an improved C-RNN-GAN, incorporating generative adversarial networks to enhance emotional expressiveness in AI-composed music. Meanwhile, Dadman and Bremdal [8] adopted a user-centric approach to symbolic music generation, allowing real-time feedback to refine generated melodies, demonstrating the role of human-AI collaboration in music composition. Earlier foundational works by Conner et al. [9] and Minu et al. [10] further reinforced the effectiveness of LSTM-based models, validating their capability to learn complex musical structures and generate compositions that maintain both coherence and diversity. Building upon these studies, this research proposes a Django-based genre-specific music generation system that extends deep learning applications by tailoring AI-generated compositions to distinct musical genres. Unlike previous works that focus on general or emotion-based music synthesis, this study integrates a structured genre-aware approach, ensuring that AI-generated music aligns with the stylistic identity of different genres while providing a user-friendly web-based interface for real-time composition.

## 3. METHODOLOGY

### 3.1 Research Design

This research employs a deep learning-based approach for AI-driven music generation. A supervised learning model is trained on a dataset of musical compositions, using a sequence-to-sequence learning framework. The system is integrated into a Django-based web application, allowing users to generate music compositions based on selected parameters. The design follows a modular architecture, with separate components for data preprocessing, model training, and music generation.

### 3.2 Data Collection

The assortment features MIDI files that are freely accessible across multiple genres from free midi websites, showcasing a diverse array of musical styles. The selected genres cover classical, jazz, rock, pop, blues, and country. These files were chosen for their musical intricacy and quality to enhance a productive learning experience.

### 3.3 Model Architecture

The proposed music generation model is built upon a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM), designed to learn and predict sequential patterns in musical compositions. The architecture is structured to process input sequences derived from MIDI datasets, capturing melodic, harmonic, and rhythmic dependencies for each genre. The model is optimized for long-term dependencies in musical progression, ensuring coherence in generated compositions.

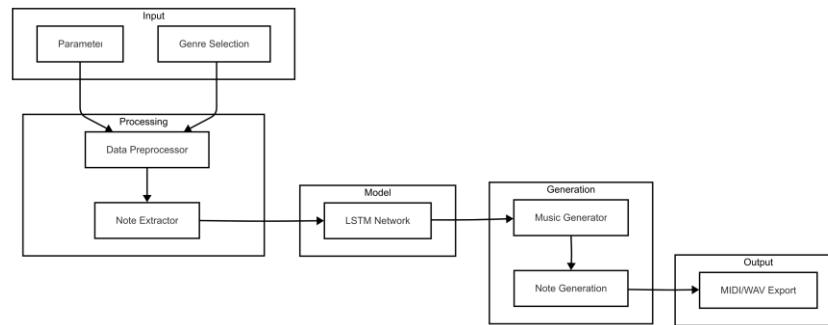


Fig. 3.3 Architecture Diagram

#### 3.3.1 Neural Network Structure

The architecture consists of the following key components:

**Embedding Layer:** Maps discrete musical tokens (notes/chords) to continuous vector representations, improving the model's ability to generalize across sequences.

**LSTM Layers:** Composed of stacked LSTM cells that retain long-range dependencies and capture the hierarchical structure of musical sequences.

**Batch Normalization & Dropout Layers:** Applied to stabilize training and mitigate overfitting by normalizing activations and introducing stochastic regularization.

**Fully Connected Dense Layer:** Computes probability distributions over the possible next notes in a sequence, enabling stochastic generation of musical elements.

#### 3.3.2 Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM)

Recurrent Neural Networks (RNNs) are created to process sequential data by retaining a hidden state that reflects dependencies over time steps. Nevertheless, conventional RNNs encounter the vanishing gradient issue, which hampers their ability to learn long-term relationships in sequences like music. To address this challenge, Long Short-Term Memory (LSTM) networks implement gated mechanisms that control the information flow, enabling the model to preserve important features across longer sequences.

#### 3.3.3 LSTM Cell Computation

The LSTM mechanism consists of three gates input, forget, and output which regulate the flow of information through the network. The operations in an LSTM cell are defined as follows:

**Forget Gate:** Determines which information from the previous state should be discarded:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

**Input Gate:** Decides which new information should be added to the memory cell:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

**Cell State Update:** Computes the new memory content:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

**Output Gate:** Regulates the output of the LSTM cell:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

where:

- $f_t$ ,  $i_t$ ,  $o_t$  represent forget, input, and output gate activations, respectively.
- $x_t$  is the input vector at time  $t$ .
- $h_t$  and  $C_t$  denote the hidden state and memory cell state.
- $W$ ,  $U$ , and  $b$  are weight matrices and bias terms.

- $\sigma$  is the sigmoid activation function, while  $\tanh$  is the hyperbolic tangent function.
- $\odot$  represents element-wise multiplication.

### 3.3.4 Loss Function & Optimization

The model is trained using categorical cross-entropy loss, ensuring the probability distribution of predicted notes closely matches the actual sequence:

$$L = -\sum_{j=1}^N y_j \log(\hat{y}_j)$$

where:

- $y_j$  is the actual note in the sequence.
- $\hat{y}_j$  is the predicted probability for that note.

To optimize the model, the Adam optimizer is employed, adjusting the learning rate dynamically:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t} + \epsilon} m_t$$

where:

- $m_t$  and  $v_t$  are moment estimates.
- $g_t$  represents the gradient at step  $t$ .
- $\theta_t$  denotes the parameter update.
- $\alpha$  is the learning rate, and  $\epsilon$  ensures numerical stability.

### 3.3.5 Music Sequence Generation

Once trained, the model generates compositions using a seed sequence, iteratively predicting the next note based on learned probability distributions. A temperature parameter controls the randomness of note selection:

$$P(y_j) = \frac{\exp\left(\frac{\hat{y}_j}{T}\right)}{\sum_k \exp\left(\frac{\hat{y}_k}{T}\right)}$$

where:

- $T$  is the temperature value.
- Lower  $T$  results in more deterministic outputs, while higher  $T$  increases randomness.

## 3.4 Algorithm (Step-by-Step Workflow)

The genre-specific music generation system follows a structured pipeline that includes data preprocessing, model training, music generation, and evaluation. The steps involved in the algorithm are outlined below:

### Step 1: Data Preprocessing

- 1.1 Load MIDI files from the dataset, categorized by genre.
- 1.2 Extract musical elements such as notes, chords, and rhythmic durations.
- 1.3 Convert extracted features into numerical representations using integer encoding.
- 1.4 Standardize the dataset and organize it into pairs of inputs and outputs for training.
- 1.5 Split the dataset into training, validation, and test sets, making sure to maintain a balanced distribution of genres.

### Step 2: Model Training

- 2.1 Initialize an LSTM-based neural network with multiple layers.
- 2.2 Configure hyperparameters such as learning rate, sequence length, and batch size.
- 2.3 Train the model on the prepared dataset, capturing temporal dependencies in musical sequences.
- 2.4 Implement Early Stopping and Model Checkpoints to prevent overfitting.
- 2.5 Validate the model by evaluating its ability to predict note sequences accurately.

Step 3: Music Generation

- 3.1 Select a seed sequence from the dataset or a user-defined input.
- 3.2 Predict the next note in the sequence based on the model’s learned patterns.
- 3.3 Append the predicted note to the sequence and iterate the process.
- 3.4 Convert the generated numerical sequence back into MIDI format for playback.
- 3.5 Store the generated composition for further analysis and user interaction.

Step 4: Evaluation & Post-Processing

- 4.1 Measure the quality of generated music using evaluation metrics such as BLEU score, melodic consistency, and harmonic similarity.
- 4.2 Perform a manual assessment to validate genre adherence and musical structure.
- 4.3 Store successful compositions in the Django-based web application for user playback and interaction.
- 4.4 Optimize the model by adjusting hyperparameters based on evaluation results.

3.5 Model Performance: Training and Validation Accuracy

The LSTM-based music generation model was trained and validated separately for each genre to ensure that stylistic characteristics were accurately learned and replicated. Performance evaluation was conducted using training and validation accuracy alongside loss values, which indicate the degree of prediction error. The results are summarized as follows:

| Genre | Training Accuracy (%) | Training Loss | Validation Accuracy (%) | Validation Loss |
|-------|-----------------------|---------------|-------------------------|-----------------|
| Blues | 91.22                 | 3.832         | 85.98                   | 3.352           |
| Rock  | 91.57                 | 3.275         | 87.88                   | 2.583           |
| Pop   | 90.40                 | 3.145         | 83.85                   | 4.007           |

Table 3.5 Model Performance Metrics

The training accuracy across all genres remains consistently above 90%, demonstrating the model’s ability to learn structured relationships in musical sequences. Meanwhile, the validation accuracy confirms that the system can generalize to unseen compositions while maintaining genre-specific characteristics. Loss values provide additional insights into model performance. A lower loss value suggests better convergence, with rock achieving the most optimal results, followed by blues, while pop exhibits a slightly higher loss, indicating a more complex structure in its rhythmic and melodic variations.

4. RESULTS AND DISCUSSION

4.1 Preliminary Data

The evaluation of the genre-specific music generation system was conducted on unseen data, ensuring that the model’s performance reflects its ability to generalize beyond the training set. The results were assessed using N Gram-based metrics and musical structure analysis, including BLEU score, melodic consistency, harmonic similarity, and repetition structure similarity.

To illustrate the effectiveness of the system, two types of visual representations were generated for each genre:

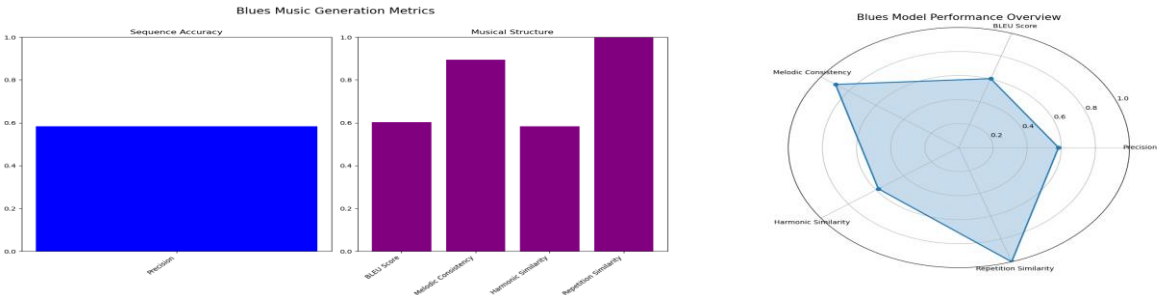


Fig. 4.1.1 Blues Music Generation metrics chart and radar chart

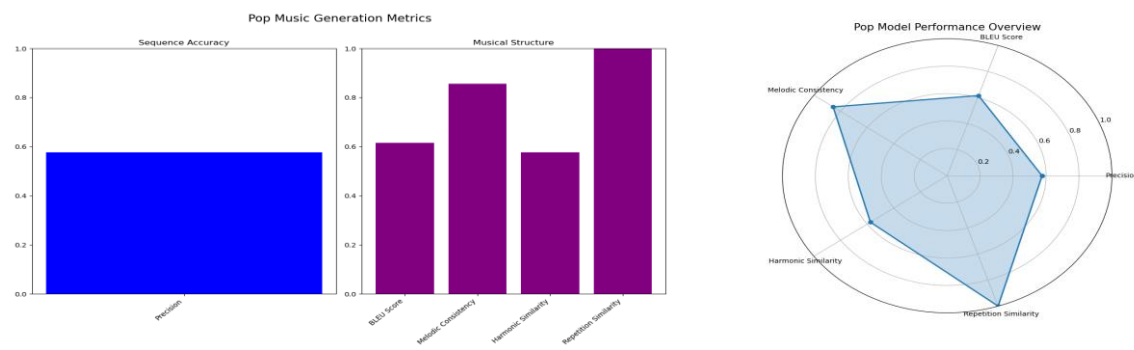


Fig. 4.1.2 Pop Music Generation metrics chart and radar chart

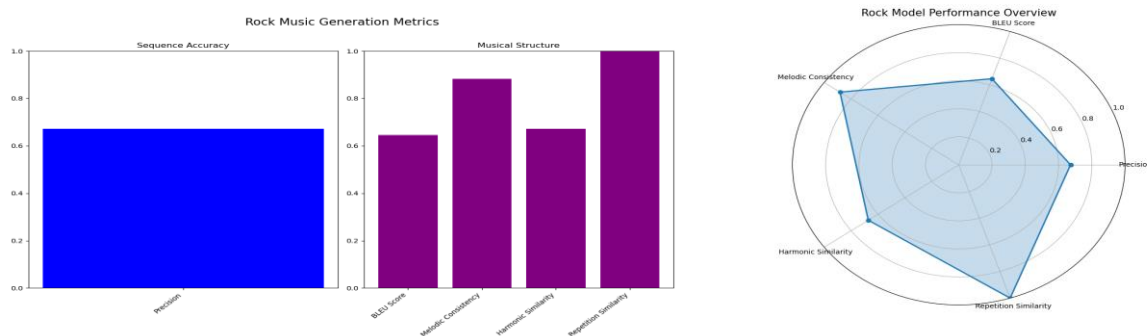


Fig. 4.1.3 Rock Music Generation metrics chart and radar chart

Additionally, piano roll view of MIDI files for the generated compositions were produced, validating the structural coherence of the AI-generated pieces.

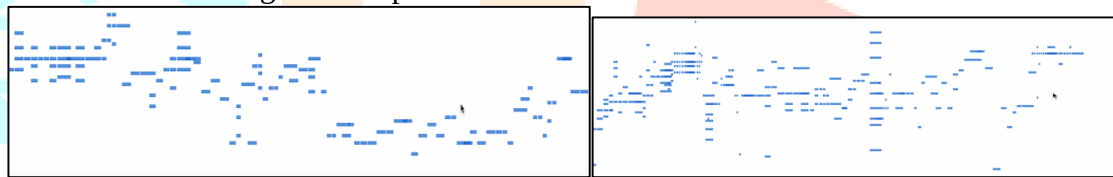


Fig. 4.1.4 Blues &amp; Pop piano roll view of generated MIDI files



Fig. 4.1.5 Rock piano roll view of generated MIDI file

The front-end interface was documented through visual representations, showcasing the interactive features and functionality of the Django-based web application.

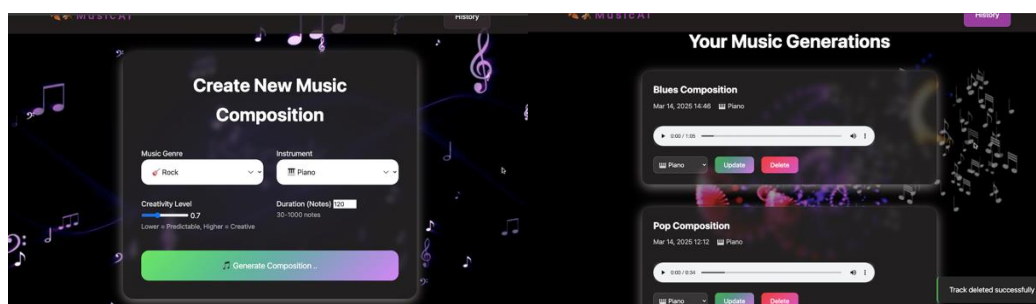


Fig. 4.1.6 Front-end Interface Home and History Page

## 4.2 Evaluation Results

The performance results for the Blues, Pop, and Rock genres on unseen data are presented below:

| Genre | BLEU Score | Melodic Consistency | Harmonic Similarity | Repetition Structure Similarity |
|-------|------------|---------------------|---------------------|---------------------------------|
| Blues | 0.6032     | 0.8940              | 0.5839              | 1.0000                          |
| Pop   | 0.6155     | 0.8554              | 0.5759              | 1.0000                          |
| Rock  | 0.6459     | 0.8827              | 0.6723              | 1.0000                          |

Table 4.2 Music Generation Metrics

The rock genre exhibited the highest BLEU score (0.6459) and harmonic similarity (0.6723), indicating stronger sequence alignment and better harmonic progression in AI-generated compositions. Meanwhile, the blues genre achieved the highest melodic consistency (0.8940), suggesting smoother note transitions and a more structured musical flow. The repetition structure similarity remained at 1.0000 across all three genres, indicating that the AI-generated compositions maintained repetitive patterns similar to human-composed music.

## 4.2 Discussion

The findings indicate that models based on LSTM are effective in capturing musical structures across various genres, showing significant differences in performance. The increased BLEU score and harmonic similarity observed in the rock genre imply that rock music may feature more organized chord progressions, facilitating the model's learning process. On the other hand, the blues genre demonstrates strong melodic consistency, suggesting that AI-generated blues pieces maintain their characteristic smooth transitions and improvisational style. These results are consistent with earlier studies, reinforcing the idea that deep learning can emulate intricate musical structures. Nonetheless, variations in performance by genre underscore the influence of dataset complexity and stylistic differences on how well the model learns. Although the findings affirm the capability of AI-driven composition, additional improvements could enhance harmonic richness and melodic nuance in genres such as pop and blues. This research effectively tackles the posed question by showing that deep learning models are capable of producing genre-specific music while preserving structural integrity. The incorporation of structured evaluation metrics, genre-specific modeling, and an intuitive web application adds to the evolving domain of AI-generated music, providing a valuable resource for music creation and exploration.

## 5. CONCLUSION

This research presented a Django-based system for genre-specific music generation utilizing Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM) to produce AI-composed musical works across the Blues, Rock, and Pop genres. The model was trained using MIDI datasets, capturing the distinctive melodic, harmonic, and rhythmic features inherent to each genre. By combining sequence learning with structured evaluation metrics, the system showcased a high degree of accuracy in generating compositions that resemble those created by humans.

The evaluation results on previously unseen data indicate that the model successfully learns and reproduces genre-specific traits, as demonstrated by BLEU scores, melodic coherence, harmonic similarity, and similarities in repetition structure. The rock genre displayed strong sequence alignment and harmonic progression, while blues compositions maintained smooth transitions and cohesive melodic flow. These results validate the efficacy of deep learning in music creation, providing measurable evidence of AI's capability to produce structured and stylistically appropriate music.

In addition to its technical contributions, this research underscores the significance of AI-driven creativity in music generation. The Django-based web application renders AI-assisted composition more accessible, fostering a connection between machine learning and musical innovation. By enabling users to generate, assess, and engage with AI-generated compositions, this study establishes a groundwork for further advancements in AI-supported music production.

This work illustrates that deep learning models can effectively create music with stylistic integrity, endorsing the wider implementation of AI in creative and artistic sectors. Future developments could concentrate on broadening genre variety, enhancing harmonic expressiveness, and improving real-time user interaction, thus further advancing AI's role in automated music composition.

## 6. ACKNOWLEDGMENT

We genuinely appreciate our faculty and colleagues for their continual support and essential perspectives that have advanced this research. Their expertise and helpful feedback have been vital in shaping this study. We also appreciate the contributions from the wider academic community, whose research has laid a solid groundwork for this work. Furthermore, we are thankful for the discussions and collaborative initiatives that have helped to enhance various elements of this project. Finally, we acknowledge the resources and opportunities that have made the successful completion of this study possible, promoting an atmosphere for exploration and innovation in AI-driven music generation.

## 7. REFERENCES

- [1] Huo, Yuanzhi, Mengjie Jin, and Sicong You. "An LSTM-based Framework for the Synthesis of Original Soundtracks." *IEEE Access* (2024).
- [2] Sambaragi, Laxmi M., et al. "Music Generation: A simplified approach." *2024 3rd International Conference for Innovation in Technology (INOCON)*. IEEE, 2024.
- [3] Zhang, Man, and Dongning Liu. "CNN-LSTM based Multimodal Models for Music Generation." *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. IEEE, 2024.
- [4] Wang, Lei, et al. "A review of intelligent music generation systems." *Neural Computing and Applications* 36.12 (2024): 6381-6401.
- [5] Li, Fanfan. "Chord-based music generation using long short-term memory neural networks in the context of artificial intelligence." *The Journal of Supercomputing* 80.5 (2024): 6068-6092.
- [6] Li, Peng, et al. "A novel Xi'an drum music generation method based on Bi-LSTM deep reinforcement learning." *Applied Intelligence* 54.1 (2024): 80-94.
- [7] Shi, Xuneng, and Craig Vear. "Generating emotional music based on improved c-rnn-gan." *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Cham: Springer Nature Switzerland, 2024.
- [8] Dadman, Shayan, and Bernt Arild Bremdal. "Crafting Creative Melodies: A User-Centric Approach for Symbolic Music Generation." *Electronics* 13.6 (2024): 1116.
- [9] Conner, Michael, et al. "Music generation using an LSTM." *arXiv preprint arXiv:2203.12105* (2022).
- [10] Minu, R. I., et al. "LSTM-RNN-based automatic music generation algorithm." *Intelligent and Cloud Computing: Proceedings of ICICC 2021*. Singapore: Springer Nature Singapore, 2022. 327-339.