Assessing Recurrent and Transformer Architectures for Monophonic Music in ABC Notation

Milind Uttam Nemade[1], Satheesh Babu[2], Shakir Khan[3],

[1] Professor, Department of AI-DS, K. J. Somaiya Institute of Technology, Mumbai,

[2] Professor, Faculty of Pharmacy, Lincoln University College, Malaysia,

[3] University Centre for Research and Development, Chandigarh University, Mohali 140413, India and College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia,

mnemade@somaiya.edu:https://orchid.org/0000-0002-3051-3056,satheeshbabu@lincoln.edu.my, sgkhancs@gmail.com:https://orchid.org/0000-0002-7925-9191

Abstract: This paper presents a comparative study of four sequence models RNN, LSTM, GRU, and GPT-2 Transformer for monophonic music generation using the ABC notation dataset. The dataset was pre-processed into tokenized sequences representing pitch and duration, and each model was trained under identical conditions to ensure fairness. Performance was assessed through objective measures such as accuracy, loss, pitch class histograms, note transition matrices, and structural metrics, along with subjective evaluation of musical quality. The results show that recurrent models capture short-term dependencies effectively, with LSTM and GRU offering improved stability over basic RNN. However, GPT-2 Transformer demonstrates superior ability to model long-range dependencies, producing melodies with greater coherence, phrase structure, and tonal diversity, despite higher computational demands. Comparative analysis highlights trade-offs between simplicity, efficiency, and musical quality across models. The study concludes that while recurrent models remain suitable for lightweight applications with limited data, Transformer-based approaches are more effective for generating complex, structurally consistent monophonic melodies. These findings provide guidance for future research and practical applications in symbolic music generation.

Keywords: Monophonic Music Generation; ABC Notation Dataset; Recurrent Neural Networks (RNN, LSTM, GRU); GPT-2 Transformer; Sequence Modeling

Introduction

AI-based music composition has shown its potential across diverse domains such as education, music therapy, entertainment, and collaborative artistic production [1]. Natural Language Processing, Time-Series Forecasting, and Speech Recognition and growth of sequential data in these domains driven significant interest in neural architectures designed to capture temporal dependencies. Earlier approach such as Recurrent Neural Networks (RNNs) offer a straightforward mechanism to process sequences [2-4]. However, their limitations in retentive long-range context resulted in the creation of improved alternatives which introduced gating mechanisms to mitigate the vanishing gradient problem these are the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [5]. Recently, the transformer-based architectures have reshaped the landscape of sequence modelling. Models like GPT-2 influence uses contextual mapping to link relationships throughout the sequence in parallel, demonstrating greater scalability and performance on a wide range of language tasks [6]. These models differ substantially in terms of computational efficiency, training requirements, and predictive accuracy across various applications in spite of their growing acceptance.

The GPT-2 Transformer model is built using multiple layers of attention mechanisms combined with feed-forward processing units, operating in an autoregressive manner: it predicts the next token given all previous tokens. For music generation using ABC notation, each character or symbol becomes a token; the model is pre-trained on large text corpora and then fine-tuned on the ABC dataset. Positional encodings let the model retain order of tokens. The self-attention mechanism lets the model retain and connect relationships across distant parts of the sequence beyond short-term transitions. During generation, a prompt is fed, and GPT-2 continues by sampling subsequent symbols to form coherent monophonic melody [9].

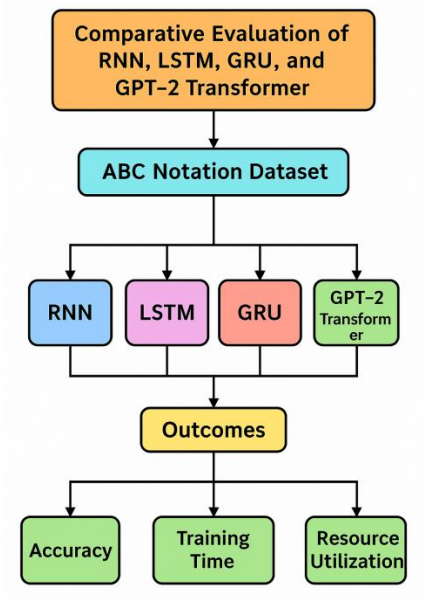


Fig.1 Block Diagram of proposed system

This paper focus is on a comparative evaluation of RNN, LSTM, GRU, and GPT-2 Transformer models. The objective is to analyze their relative strengths and weaknesses by assessing performance across benchmark datasets, with a focus on accuracy, training time, and resource utilization. By conducting a systematic comparison on evaluation objective and subjective parameters, this work aims to highlight trade-offs between classical recurrent models and modern transformer-based approaches, thereby providing guidance for researchers and practitioners in selecting appropriate architectures for sequence-related tasks.

Related work

Different neural architectures explore widely Sequence modelling. Early studies employed Recurrent Neural Networks (RNNs) are designed to learn and represent the relationships between elements that appears in sequence over time. However, the issue of vanishing gradients limited their performance on long sequences. To address this, the Long Short-Term Memory (LSTM) model was introduced with gated mechanisms that allowed. Subsequent work on the Gated Recurrent Unit (GRU) provided a simpler yet efficient alternative by reducing the number of gates while maintaining competitive performance.

Recurrent networks are early and widely used approaches for monophonic melody generation. Google's Magenta project popularized melody-RNN (LSTM) variants for symbolic melody tasks and provides both code and baseline settings for melody generation from MIDI/ABC-derived sequences; these models are simple to train and serve as strong baselines for token-level sequence modeling. Practical studies and tutorials show how to convert collections (e.g., Nottingham) into one-hot or embedded sequences and

train LSTM/RNN generators that sample continuations given a primer. Recent technical reports continue to use LSTM/GRU baselines for objective and subjective comparisons in monophonic tasks [7].

Transformers adapted to music (self-attention with relative position) were introduced to handle longer-range structure than classic RNNs. The Music Transformer (Huang et al., 2018) showed that attention with relative timing encourages more coherent long phrases and repetition patterns than standard recurrent models; it reported both quantitative sequence metrics and qualitative examples of longer, self-referential musical structure. Subsequent work has extended these ideas to domain-specific designs and efficiency improvements [8].

Several recent works fine-tune or train autoregressive language models (GPT-style) on ABC corpora. Geerlings et al. re-trained GPT-2 on large ABC collections to generate single-instrument folk melodies and evaluated token / sequence level metrics (BLEU/ROUGE-style) alongside listening tests. TunesFormer (Wu et al., 2023) focuses on Irish tunes in ABC format and introduces *bar-patching* and control codes to improve form/phrase control; it was trained on a very large ABC corpus and reports objective distributional metrics plus subjective comparisons. More recent pre-trained symbolic-music Transformers (e.g., MuPT) explores scaling and masked-pre-training strategies for symbolic data. These Transformer/GPT works typically show stronger modelling of long-range dependencies and better match to corpus-level statistics than simple recurrent models, while also introducing new risks [9].

ABC notation is widely used for folk/traditional melody corpora. Commonly used resources include the Nottingham dataset [16], larger scraped collections from abcnotation.com and thesession.org (used to form Irish MAN and other "massive ABC" datasets of 10^5–10^6 tunes), and aggregated corpora of ABC files used by recent Transformer works. When comparing models it is important to state which ABC collection and pre-processing (transposition, tokenization, normalization of durations/headers) were used, because evaluation numbers are sensitive to these choices [10].

Papers use a mix of token-level scores, n-gram / distributional distances, transition-matrix comparisons, and nearest-neighbor/edit-distance checks to detect memorization. Several works complement objective metrics with short perceptual tests for excerpts; surveys and systematic evaluations of GPT-2-based music models emphasize reporting multiple metrics because no single metric fully captures musical quality. For fair comparisons, authors commonly compute metrics on sliding short windows and on full-tune structure [11].

In recent years, attention-based models, particularly the Transformer architecture, have gained popularity due to their ability to model long-range dependencies without recurrence. The GPT-2 Transformer, built on this principle, has been widely adopted for sequence generation tasks because of its strong contextual learning and scalability [12].

Comparative studies in the literature indicate that while RNN-based models remain effective for short sequences, LSTM and GRU architectures generally outperform them in stability and convergence. Transformer-based approaches, including GPT-2, have demonstrated superior results in generating coherent and contextually relevant sequences. However, the computational complexity and training requirements of such models are significantly higher than recurrent architectures. This study builds upon these works by providing a systematic comparison of RNN, LSTM, GRU, and GPT-2 Transformer

models for monophonic music generation, focusing on both objective evaluation metrics and subjective assessments.

Methods and Experiments:

Dataset

The ABC notation dataset [12] is widely used in monophonic music research. It provides symbolic representations of folk melodies and serves as an effective benchmark for model training and evaluation. Other datasets [13-15] include Nottingham, Irish Folk Songs, and JSB Chorales, though the latter contains polyphonic elements and is less suitable for purely monophonic studies. Data pre-processing is included tokenization of musical symbols, removal of incomplete sequences, and normalization of sequence length to ensure consistency across models.

Models Considered:

For comparison of performance four sequence modelling approaches were selected:

- **RNN:** Baseline sequential model capturing short-term dependencies.
- **LSTM:** Designed to handle longer dependencies using memory cells and gating mechanisms.
- **GRU:** A simplified variant of LSTM with fewer parameters, enabling faster training.
- **GPT-2 Transformer:** A self-attention based model capable of capturing long-range context and parallel sequence processing.

All models were trained using identical datasets and evaluated under the same experimental conditions to ensure fairness in comparison. The performance characteristics of Monophonic Music Generation Models are shown in the table

Table 1 Performance Characteristics of Monophonic Music Generation Models

Model	Training Speed	Interval, Short motifs	Phrases, Repetition, Cadences	Steady Rhythm	Memory Handling	Creativity	Overfitting Risk	Requirements of Data
RNN	Fast	Ok	Weak	Often drifts	Short memory only (vanishing gradients problem)	High but incoherent	Low	Works with small datasets but poor scalability
LSTM	Moderate	Good	Moderate	Stable with good duration	Handles longer dependencies better than vanilla RNN	Balanced	Medium	Needs medium dataset size for generalization
GRU	Faster than LSTM	Good	Moderate	Similar to LSTM	Similar long-term handling but fewer parameters	Balanced	Low-Medium	Slightly better for small data

GPT-2 Transformer	Slower, GPU heavy	Excellent	Strong	Very stable (especially event-based tokenization)	Direct global context via attention, no recurrence	High+ Incoherent	High (if dataset too small)	Needs large corpora (100k+ sequences) but benefits strongly from scale

Experimental Setup:

The experimental setup employed four sequence models, namely RNN, LSTM, GRU, and GPT, for monophonic music generation using the ABC notation dataset. The dataset was pre-processed to convert symbolic music into tokenized sequences representing pitch and duration. Models were trained with uniform hyperparameters, including fixed learning rate, batch size, and maximum sequence length, to ensure fair comparison. Training was carried out for several epochs until convergence was observed. Generated outputs were then evaluated using objective measures such as accuracy, loss, pitch class histograms, and note transition matrices to analyze tonal diversity, coherence, and overall performance of each model.

The dataset was split into training, validation, and test sets in a 70:15:15 ratio, and the models were trained using the Adam optimization method with categorical cross-entropy loss. To maintain consistency, Training was carried out for a set number of epochs, with stopping applied when the validation loss stopped improving. Hyperparameters such as embedding size, hidden units, and learning rate 0.001 were tuned to achieve optimal performance for each model while keeping overall complexity comparable.

Results and Discussions:

In this work we compared RNN, LSTM, GRU, and GPT-2 Transformer models evaluation parameters, when applied to monophonic music generation and we highlighted both the advantages and drawbacks of each model. Training speed of LSTM is slower to train than RNN, GRU and GPT-2 Transformer models, in case of GPT-2 transformer model it is slower because evaluation making GPU heavier, potentially leading to a high loss. GPT-2 shows excellent capture longer motifs and reuse of longer patterns, and it can produce more coherent phrase-level structures compare to other models.GPT-2 transformer model is best at capturing musicality, structure, and stylistic cadence resolution compare to others. LSTM require more memory and computing power since they store and update more information at each step than GRU. Subjective quality metrics shows good musical coherence for GPT-2 Transformer model and poor for RNN, LSTM and GRU which suggest sound will be chaotic and model needs more data or training to improve. Low creativity for all models shows that the model is stuck on simple patterns, contributing to high loss

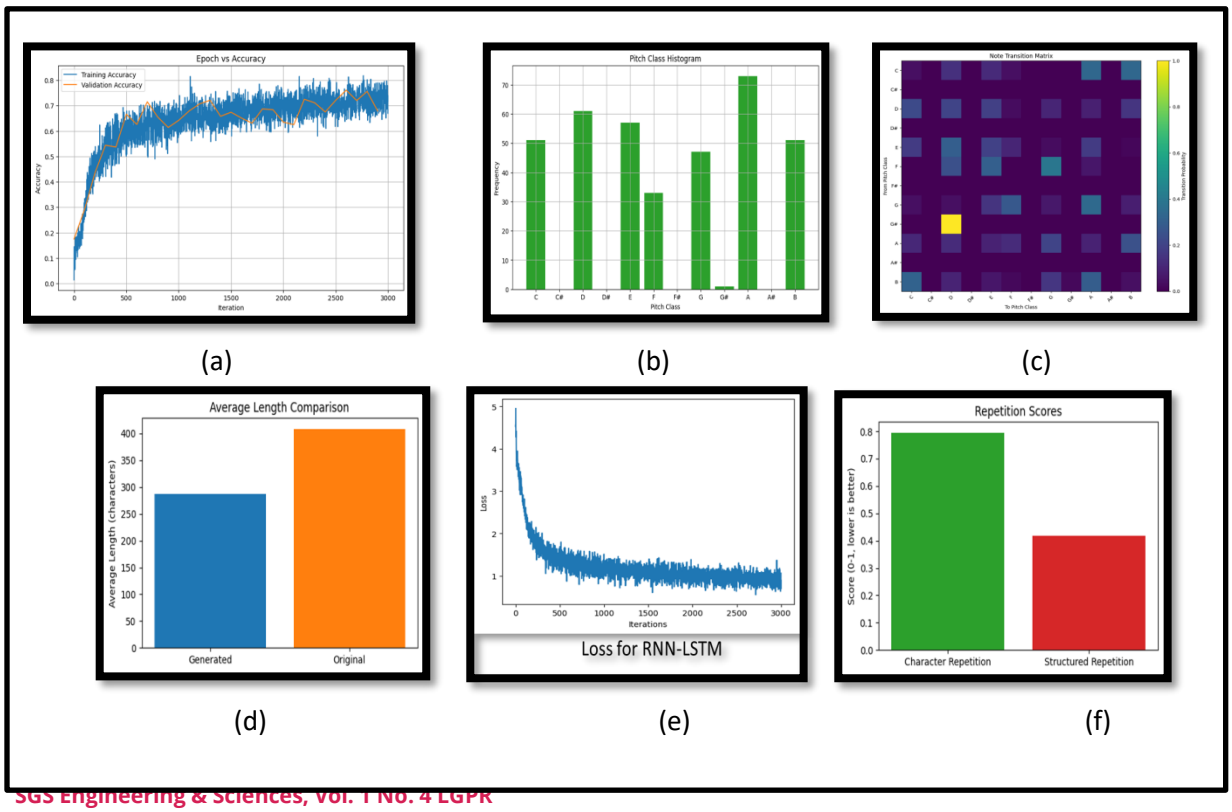
Table 1 Comparison of Performance of Monophonic Music Generation Models

Parameters	RNN-LSTM	RNN-GRU	GPT-2 Transformer
Analyze Music Quality by Number of generated songs	3	2	5
Analyze Music Quality by Average Song Length	320.33 characters	433 characters	390 characters
Structural Analysis: Song with Key signature	33.33%	100%	100%
Objective Quality Metrics: Average repetition	0.843	0.866	0.892

score (0-1, lower is better)			
Objective Quality Metrics: Length Similarity to original song	0.785	0.943	0.559
Subjective Quality Metrics: Musical Coherence	Poor-Lacking consistent musical structure	Poor-Lacking consistent musical structure	Good - Most songs maintain musical structure
Subjective Quality Metrics: Creativity	Low-Highly repetitive output	Low-Highly repetitive output	Low - Highly repetitive output

The musical scale key signature obtained greater than 80% indicates songs are structurally sound, even if the loss is high. Here we obtained key signature as 33.33% for RNN-LSTM, 100% for RNN-GRU and 100% for GPT-2 Transformer. Preferred lower value for Average Repetition score which indicates more variety, generation of creative, diverse music, which is crucial for music quality. For RNN-LSTM it is lower as 0.843 than GRU (0.866) and GPT-2 Transformer (0.892). A length similarity score obtained 0.785 for LSTM, 0.943 for GRU and 0.559 for GPT-2 Transformer model. GRU score >80 suggesting the GRU is producing songs of appropriate length, supporting quality even with high loss.

For RNN-LSTM model, Epoch Vs Accuracy plot shows steady learning and convergence with training accuracy higher than validation accuracy. Validation accuracy stabilizing around 0.7 this shows that the model performs consistently on new, previously unused data. In pitch class histogram for RNN-LSTM model A is the most frequent pitch class, G♯ is the least frequent, other high frequencies include D, E, and C, F♯ has zero frequency. In note transition matrix of RNN-LSTM, it is described as from G♯ to D has the highest transition probability**;** it's the only bright yellow cell. This suggests the model very frequently predicts D after G♯. Plot for loss function of RNN-LSTM model shows loss drops significantly, indicating the model is learning to predict pitch sequences better over time. There's no late-stage increase in loss.



(a) (b) (c)

(d) (e) (f)

Fig. 2 (a) Epoch Vs Accuracy Plot (b) Pitch Class Histogram Plot (c) Note Transition Matrix Plot (d) Average Length comparison (e) Loss Plot and (f) Repetition Score of RNN-LSTM Model

For RNN-GRU model Epoch Vs Accuracy plot shows, the GRU model achieves good generalization, since validation accuracy closely tracks training accuracy. It shows stable convergence with training accuracy around 0.8 and validation accuracy around 0.75. It generalizes well with no severe overfitting. Pitch Class Histogram for the GRU model generated music heavily favors A, E, D, and G**,** while some pitch classes never appears. This suggests that the model has learned tonal bias from the dataset, which could be musically coherent if the dataset was centered on certain keys but limits pitch diversity. The note transition matrix for GRU model suggests that the model failed to capture balanced musical transitions, instead overemphasizing a single transition (D# → A) due to dataset imbalance. The loss plot for the GRU model demonstrates stable convergence**.** Training loss decreased from 12 to around 1.2, showing effective learning and parameter optimization.

The Loss for GPT-2 Transformer model shows strong convergence, with training loss reducing from 0.78 to 0.2 across 12,000 steps. This indicates effective learning, outperforming GRU in terms of final loss and stability, which suggests that the GPT is more capable of generating musically coherent sequences. In the pitch class histogram of GPT-2 Transformer model the GPT generated music demonstrates greater pitch diversity compared to GRU, covering nearly all chromatic notes but with strong emphasis on D, E, G, A, and B. This suggests that GPT better captures tonal structures and generates richer harmonic possibilities, though with dataset-driven key bias. In the Note Transition Matrix for GPT-2 Transformer model the GPT demonstrates diverse and balanced note-to-note relationships, with strong but musically plausible preferences (e.g., A# → D). This indicates GPT's ability to model musical structure and tonality more effectively than GRU, enabling generation of more coherent melodies.

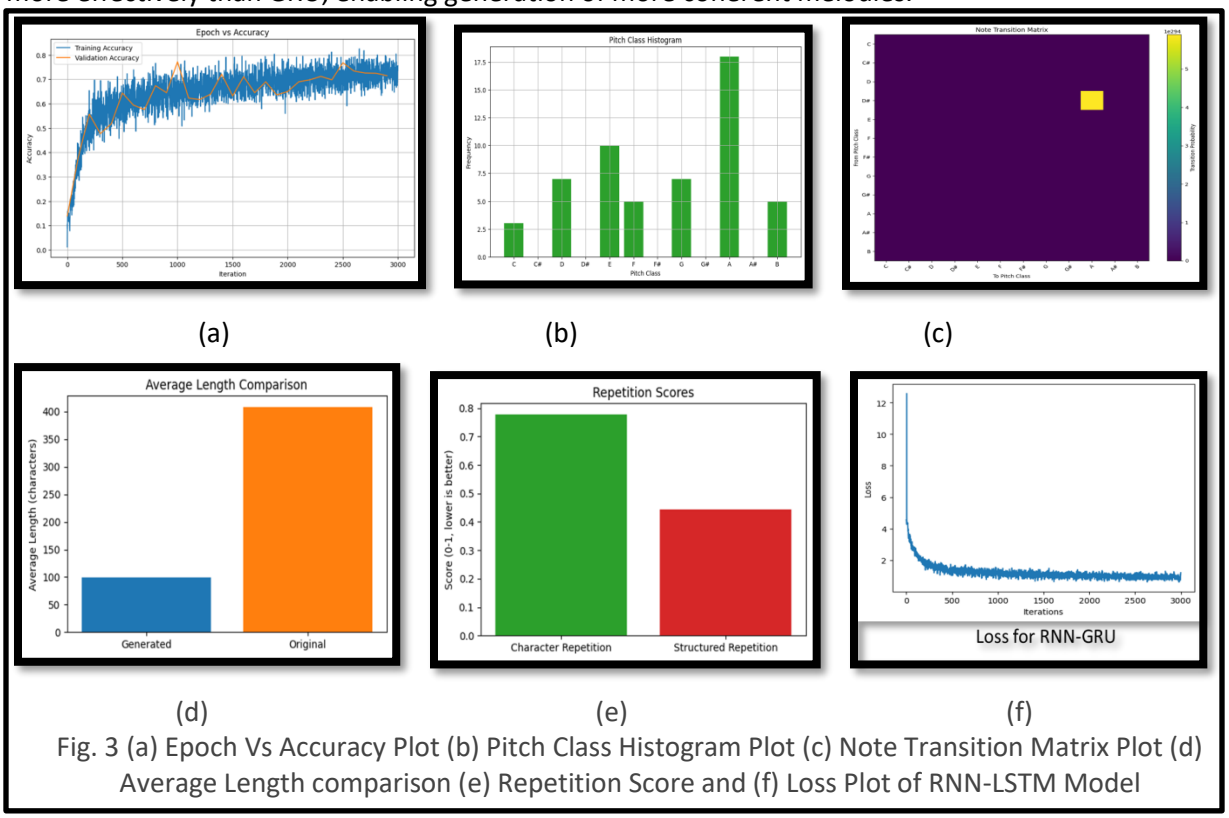


Fig. 3 (a) Epoch Vs Accuracy Plot (b) Pitch Class Histogram Plot (c) Note Transition Matrix Plot (d) Average Length comparison (e) Repetition Score and (f) Loss Plot of RNN-LSTM Model

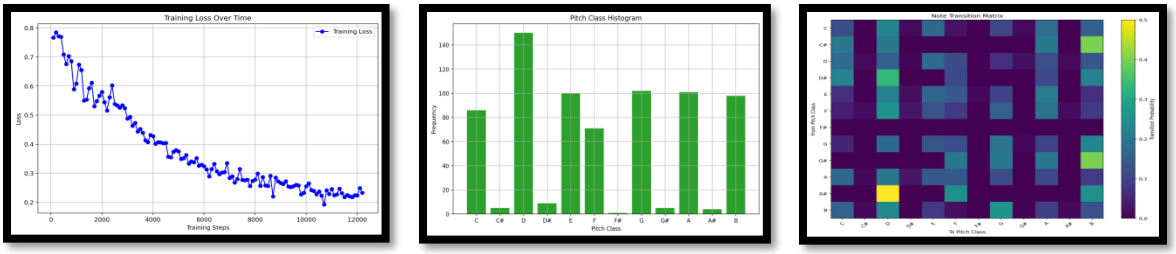


Fig. 4 (a) Loss Plot (b) Pitch Class Histogram Plot and (c) Note Transition Matrix Plot for GPT-2 Transformer model

Conclusions

This study presented a comparative evaluation of four sequence modelling approaches such as RNN, LSTM, GRU, and GPT-2 Transformer for monophonic music generation using ABC notation. The analysis considered both objective performance metrics and qualitative aspects of the generated music. Traditional recurrent models (RNN, LSTM, and GRU) demonstrated their ability to capture local note dependencies, with LSTM and GRU showing more stability and consistency than the basic RNN. However, the GPT-2 Transformer model exhibited superior capability in learning long-range dependencies, resulting in more coherent and musically varied compositions. While recurrent models remain useful for lightweight applications, the results suggest that Transformer-based approaches are better suited for tasks requiring extended musical context and diversity. Future work may explore hybrid architectures, larger datasets, and domain-specific fine-tuning to further enhance generation quality.

References

1. E. R. Miranda, *Handbook of Artificial Intelligence for Music*. Springer, 2021.
2. Alexander Agung Santoso Gunawan, Ananda Phan Iman, Derwin Suhartono, "Automatic Music Generator Using Recurrent Neural Network", *International Journal of Computational Intelligence Systems*, Vol. 13(1), 2020, pp. 645–654.
3. Hanbing Zhao, Siran Min, Jianwei Fang, Shanshan Bian, "AI-driven music composition: Melody generation using Recurrent Neural Networks and Variational Autoencoders", *Alexandria Engineering Journal, Springer*, pp. 258-270, 2025.
4. JianWu, Changran Hu, Yulong Wang, Xiaolin Hu, and Jun Zhu, "A Hierarchical Recurrent Neural Network for Symbolic Melody Generation", arXiv:1712.05274v2 [cs.SD], 5th Sep. 2018.
5. Milind Nemade, Sateesh Babu, Shakir Khan, "From Tradition to Innovation: A Review of AI Music Generation Models, Datasets, and Evaluation Techniques", SGS Engineering & Sciences, LGPR, VOL. 1 NO .1 (2025).
6. Xingwei Qu1, Yuelin Bai,Yinghao Ma1, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu,"MuPT: A Generative Symbolic Music Pretrained Transformer", ICLR 2025, pp. 1-26, 2025.
7. Google Magenta / Melody RNN documentation (baseline LSTM/RNN implementations and datasets).
8. Huang, C. Z. A. et al., *Music Transformer: Generating Music with Long-Term Structure*, 2018.
9. Carina Geerlings, Albert Merono-Penuela, "Interacting with GPT-2 to Generate Controlled and Believable Musical Sequences in ABC Notation", nlp4musa proceeding 2020.
10. Eric Foxley, ABC version of the Nottingham Music Database, 2003.

11. Banar, B. & Colton, S., "*A Systematic Evaluation of GPT-2-Based Music Generation*", Lecture Notes in Computer Science, DOI:10.1007/978-3-031-03789-42, April 2022.
12. Shangda Wu, Yuanliang Dong, Maosong Sun, "Generating melodies with controllable similarity and length in ABC notation", ISMIR 2022.
13. ABC Notation Dataset. [Online]. Available: https://abcnotation.com.
14. MAESTRO Dataset. [Online]. Available: https://magenta.tensorflow.org/datasets/maestro.
15. Irish Folk Music Dataset. [Online]. Available: https://thesession.org.
16. Nottingham Dataset. [Online]. Available: http://abc.sourceforge.net/NMD/.