International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

MelodAI – Composing the Future of Music with Artificial Intelligence

Gangireddy Charishma, Cheerladinne Sai Deepthi, Bandaru Naga Rohnitha, Gugulothu Koteswara Rao Naik
(22BQ1A5442) (22BQ1A5424) (22BQ1A5413) (22BQ1A5451)
Department of Artificial Intelligence and Data Science
Vasireddy Venkatadri Institute of Technology (Autonomus)
Affiliated to JNTUK | NAAC 'A' Grade | Guntur, Andhra Pradesh — 522508

Abstract—Music composition has traditionally been an endeavour requiring significant human expertise, creativity, and an in-depth understanding of musical theory. The rapid advancement of deep learning methodologies has opened transformative possibilities for automating creative tasks, including music generation. This paper presents MelodAI, an artificial intelligence-driven system designed to generate original musical compositions by learning temporal patterns from existing MIDI-based musical datasets. The proposed system employs Long Short-Term Memory (LSTM) networks, a specialized class of Recurrent Neural Networks (RNN) capable of modelling long-range sequential dependencies, to capture and reproduce harmonic and melodic structures. MelodAI processes encoded MIDI data, trains a multi-layered LSTM model to learn note sequences and chord progressions, and generates novel compositions that are musically coherent and contextually relevant. The system achieves a training accuracy of approximately 94.7% and demonstrates strong qualitative performance through human evaluation studies. Experimental results indicate that the proposed architecture consistently outperforms baseline Hidden Markov Model and basic RNN approaches in generating structured and melodically appealing music. The system is capable of real-time generation and outputs standard MIDI files, making it immediately useful for entertainment, therapy, gaming, and education.

Index Terms—*Music Generation, Long Short-Term Memory (LSTM), Deep Learning, MIDI Processing, Recurrent Neural Networks, Sequence Prediction, Music21, Artificial Intelligence.*

I. INTRODUCTION

The convergence of artificial intelligence and human creativity represents one of the most compelling frontiers of contemporary research. Among the various domains of creative intelligence, music generation stands out as particularly challenging due to the multi-dimensional nature of musical structure, encompassing melody, harmony, rhythm, timbre, and temporal dynamics. The question of whether a machine can compose music that is not only syntactically correct but also aesthetically meaningful has attracted significant scholarly and industrial interest over the past two decades.

Traditional music composition demands years of formal training, a refined understanding of music theory, and a deep intuitive grasp of emotional resonance. These requirements create a high barrier to entry for aspiring composers, particularly in domains such as background scoring for games, films, and therapeutic environments, where demand for custom music far exceeds the availability of professional composers. Automated music generation systems offer a promising solution by enabling on-demand, customizable, and contextually appropriate composition.

Early approaches to algorithmic music composition relied on rule-based systems and probabilistic models such as Markov chains and Hidden Markov Models (HMM). While these methods produced structurally valid sequences, they frequently failed to capture the long-range dependencies and nuanced patterns that characterize meaningful musical phrases. The advent of deep learning, particularly Recurrent Neural Networks (RNN) and their more advanced variant, Long Short-Term Memory (LSTM) networks, has dramatically altered this landscape.

This paper introduces MelodAI, a deep learning-based music generation system that leverages LSTM networks trained on MIDI (Musical Instrument Digital Interface) encoded datasets. MelodAI is designed to learn the structural and stylistic characteristics of musical compositions and generate novel sequences that are harmonically coherent and melodically engaging. The system incorporates a user-friendly interface and outputs standard MIDI files for immediate playback and editing.

The remainder of this paper is organized as follows: Section II presents a review of existing literature; Section III defines the problem statement; Section IV outlines the objectives; Section V describes the methodology; Section VI explains the system architecture; Section VII discusses implementation details; Section VIII presents results; Sections IX–XII address applications, advantages, limitations, and future scope; and Section XIII concludes the paper.

II. LITERATURE REVIEW

The field of algorithmic and AI-assisted music composition has a rich history spanning several decades. Early symbolic approaches, such as those pioneered by Hiller and Isaacson in the Illiac Suite (1957), used rule-based methods to generate music conforming to classical counterpoint rules [1]. Subsequent decades saw the application of probabilistic models, most notably Markov chains, which modelled state transitions between musical notes. While computationally tractable, these models suffered from fixed context windows and an inability to capture global musical structure [2].

The introduction of neural networks into music generation offered a significant paradigm shift. Mozer [3] was among the first to apply recurrent neural networks to music composition, demonstrating that RNNs could learn temporal dependencies in musical sequences. However, vanilla RNNs suffered from the well-documented vanishing gradient problem, limiting their effective memory horizon. The seminal work of Hochreiter and Schmidhuber [4] introduced LSTM networks in 1997, which addressed this limitation through input, output, and forget gates that regulate information flow through the network.

The application of LSTMs to music generation was explored extensively by Johnson [5] in the Magenta project by Google Brain, demonstrating that deep LSTM models could generate piano music with notable structural coherence. Briot et al. [6] provided a comprehensive survey of deep learning approaches in music generation, identifying LSTM-based architectures among the most consistently performant. More recently, attention-based architectures, particularly the Transformer model [7], have been applied with considerable success. Huang et al. [8] proposed Music Transformer, employing relative attention for extended piano compositions. OpenAI's MuseNet [9] demonstrated that large-scale Transformer models trained on diverse musical corpora can generate multi-instrument compositions.

A comparative overview of key existing systems is presented in Table I. As indicated, prior deep learning systems are either computationally expensive, domain-restricted, or limited in output controllability. MelodAI addresses these gaps by combining LSTM sequence modelling with a lightweight, accessible design optimized for real-world deployment.

TABLE I: Comparison of Existing Music Generation Systems

System	Model	Output	Coherence	Access
Mozer 1994 [3]	Vanilla RNN	Mono	Low	High
Magenta [5]	Deep LSTM	Piano	Moderate	Moderate
Music Transformer [8]	Transformer	Piano	High	Low
MuseNet [9]	GPT-2	Multi	Very High	Very Low
MelodAI (Ours)	3-Layer LSTM	MIDI	High	High

III. PROBLEM STATEMENT

Despite the growing demand for original music content across entertainment, education, and therapeutic domains, several critical challenges impede accessibility and scalability of music composition:

- Traditional music composition demands years of expert training, creating significant barriers for non-professional users and developers of interactive systems.
- Existing AI models for music generation, particularly older probabilistic models, frequently lack melodic coherence, producing sequences that are locally plausible but globally inconsistent.
- Manual identification of musical patterns in large corpora is both time-intensive and subjective, making scalable data-driven analysis difficult.
- Industries such as gaming, film production, and digital therapeutics require rapid, customizable music generation without the overhead of commissioning professional composers.
- Many existing deep learning systems require substantial computational infrastructure, limiting deployment for smaller institutions and independent developers.

IV. OBJECTIVES

The primary and secondary objectives of this research are as follows:

- To design and implement an end-to-end AI-driven system for generating original music compositions using deep learning techniques.
- To leverage Long Short-Term Memory (LSTM) networks for sequence modelling of musical notes, chords, and durations extracted from MIDI files.
- To preprocess and encode raw MIDI musical data into structured numerical representations suitable for neural network training.
- To train and optimize a multi-layered LSTM model through hyperparameter tuning to achieve high generation accuracy.
- To generate musically coherent MIDI output playable in real time and editable using standard music production tools.
- To evaluate the system against baseline models using quantitative metrics and qualitative human evaluation studies.

V. METHODOLOGY

A. Data Collection

The MelodAI system is trained on a curated corpus of MIDI files sourced from the Lakh MIDI Dataset [10], comprising over 176,000 unique MIDI files, and the Classical Music MIDI Archive. A subset of approximately 2,500 MIDI files across three genres—classical piano, jazz, and ambient electronic—was selected. This multi-genre dataset ensures the model learns generalized musical patterns rather than overfitting to genre-specific conventions. All selected files were verified for structural completeness, and corrupted or incomplete MIDI files were discarded during initial data audit.

B. Data Preprocessing

MIDI files encode music as event sequences rather than raw audio, providing a structured symbolic representation well-suited for sequential modelling. The Music21 Python library [11] was employed to parse MIDI files and extract notes, chords, and rest durations. Each note was encoded by pitch value and duration; chords were represented as sorted tuples of constituent pitches. The full vocabulary of unique musical elements was compiled and assigned integer indices.

Input sequences of fixed length N= 100 were constructed by sliding a window across each tokenized note stream. The corresponding target was the immediately subsequent element. All inputs were normalized to [0, 1] by dividing indices by vocabulary size. One-hot encoding was applied to target labels for categorical cross-entropy computation. The dataset was partitioned into training (80%), validation (10%), and test (10%) subsets using stratified sampling.

C. Model Architecture – LSTM Networks

The Long Short-Term Memory network, introduced by Hochreiter and Schmidhuber in 1997 [4], is a specialized variant of the Recurrent Neural Network designed to overcome the vanishing gradient problem. The fundamental innovation lies in its memory cell and three gating mechanisms: the forget gate, the input gate, and the output gate. Together, these mechanisms allow the LSTM cell to selectively retain information over arbitrarily long sequences—a property critical in music, where motifs established early in a composition may recur hundreds of time steps later.

The MelodAI architecture consists of a stacked LSTM network comprising: an Embedding layer mapping integer-encoded note indices to dense 256-dimensional vectors, followed by three LSTM layers with 512, 512, and 256 units respectively. Dropout layers with a rate of 0.3 are interleaved between LSTM layers to prevent overfitting. A fully connected Dense layer with softmax activation produces a probability distribution over the musical vocabulary. The full architecture is summarized in Table II.

TABLE II: MelodAI LSTM Model Architecture

L#	Layer / Config	Output	Params
1	Embedding (Vocab×256)	(100, 256)	Learnable
2	LSTM (512, ret_seq=T)	(100, 512)	1,574,912
3	Dropout (0.3)	(100, 512)	0
4	LSTM (512, ret_seq=T)	(100, 512)	2,099,200
5	Dropout (0.3)	(100, 512)	0
6	LSTM (256 units)	(256)	787,456
7	Dense (Vocab, Softmax)	(Vocab)	Variable

D. Training Process

The model was trained using the Adam optimizer [12] with an initial learning rate of 0.001 and a batch size of 64 over 200 epochs. Categorical cross-entropy was employed as the loss function. A ReduceLROnPlateau callback reduced the learning rate by a factor of 0.5 when validation loss stagnated over 10 consecutive epochs. Model checkpointing retained weights corresponding to minimum validation loss. Training was performed on an NVIDIA RTX 3080 GPU using TensorFlow 2.9. Gradient clipping with a threshold of 1.0 was applied to stabilize early training phases.

E. Music Generation Process

During inference, the trained model receives a seed sequence of N=100 elements randomly sampled from the test corpus. A temperature-controlled sampling strategy is employed: a temperature parameter τ scales the logits before softmax. Lower τ (e.g., 0.5) produces conservative outputs while higher values (e.g., 1.2) introduce creative variability. For primary experiments, $\tau = 0.9$ was selected as a balance between coherence and novelty. Generated tokens are decoded into a Music21 stream and exported as a standard MIDI file.

VI. SYSTEM ARCHITECTURE

The MelodAI system is composed of five principal functional modules operating in a sequential pipeline. The Data Ingestion Module accepts MIDI files and interfaces with Music21 to extract musical primitives. The Preprocessing Module converts these primitives into integer-encoded sequences, constructs the vocabulary, generates training windows, and applies normalization and one-hot encoding. The Model Training Module manages LSTM construction, compilation, and optimization, incorporating callbacks for learning rate scheduling and checkpointing.

The Generation Module performs iterative next-note prediction from a user-defined or randomly selected seed and assembles the output token stream. The Output Module decodes tokens into a Music21 score, applies post-processing such as quantization and tempo assignment, and exports the MIDI file. Users interact through a Flask-based web interface exposing genre selection, sequence length, and temperature controls. This frontend-backend separation ensures the deep learning pipeline can be independently scaled without modifying the user-facing application.

VII. IMPLEMENTATION DETAILS

The MelodAI system was implemented entirely in Python 3.9 using the following technology stack:

- TensorFlow 2.9 / Keras: Primary deep learning framework for constructing, training, and serializing the LSTM model via the Keras Sequential API.
- Music21 (v7.3): Toolkit for MIDI parsing, musical element extraction, and score reconstruction, providing support for both monophonic and polyphonic structures.
- NumPy and Pandas: Efficient numerical array manipulation, sequence windowing, and dataset management during preprocessing.
- Scikit-learn: Dataset splitting with stratified sampling to maintain genre distribution across training, validation, and test subsets.
- Flask (v2.1): Lightweight web framework managing HTTP routing and API endpoints for generation requests.
- Pygame (v2.1): In-browser MIDI playback functionality enabling users to audition generated compositions within the interface.

The trained model weights are serialized in HDF5 format and reloaded by the generation module upon user request. Generated MIDI files are stored in an output directory and served via a download endpoint.

VIII. RESULTS AND DISCUSSION

The MelodAI system was evaluated using quantitative performance metrics and a structured human evaluation study. Quantitative metrics included training accuracy, validation accuracy, categorical cross-entropy loss, and perplexity on the held-out test set. Qualitative evaluation was conducted through a listener study involving 30 participants with varying levels of musical training, rating compositions on melodic coherence, rhythmic regularity, and overall musicality on a 5-point Likert scale.

TABLE III: Quantitative Comparison of Music Generation Models

Model	Tr. Acc.	Val. Acc.	Perplexity	Human (/5)
HMM (Baseline)	72.4%	68.1%	38.6	2.4
Vanilla RNN	81.3%	76.5%	22.3	2.9
1-Layer LSTM	88.7%	85.2%	14.8	3.5
MelodAI (Ours)	94.7%	92.1%	8.2	4.2

The results in Table III demonstrate a clear performance hierarchy. The HMM baseline achieved the lowest accuracy and highest perplexity, reflecting its fundamental limitation in capturing long-range dependencies. The vanilla RNN showed marginal improvement but remained hampered by vanishing gradients over sequences of length 100. The single-layer LSTM achieved a substantial improvement, validating gated memory mechanisms for musical sequence modelling. The proposed MelodAI three-layer LSTM achieved the best results, with a test accuracy of 92.1% and perplexity of 8.2. Human evaluators assigned MelodAI an average of 4.2/5, with particularly high ratings in melodic coherence (4.4) and rhythmic regularity (4.3).

Training and validation loss decreased smoothly over 200 epochs without significant divergence, confirming that the applied regularization strategies successfully mitigated overfitting. The model demonstrated sensitivity to the temperature parameter: at $\tau = 0.9$, the most musically satisfying results were obtained, consistent with subjective evaluations. Qualitatively, generated compositions exhibited tonal centre establishment, motif repetition, and phrase-level structure aligned with training genre conventions.

TABLE IV: Hyperparameter Tuning Results for MelodAI

Trial	Emb. Dim	LSTM Units	Dropout	LR	Accuracy
1	256	512/512/256	0.3	0.001	0.9470
2	256	256/256/128	0.4	0.001	0.9312
3	128	512/256	0.3	0.0005	0.9218
4	128	256/256	0.5	0.001	0.9081

IX. APPLICATIONS

The MelodAI system has broad applicability across diverse industries:

- Gaming and Interactive Media: Real-time adaptive background music generation responding to in-game states and player actions without reliance on manually curated soundtracks.
- Music Therapy: Generation of personalized, mood-appropriate musical sequences for clinical therapeutic applications, including management of anxiety, depression, and neurodevelopmental conditions.
- Film and Advertising Post-Production: Rapid prototyping of background scores and jingles during early production phases, reducing dependency on commissioned composers.
- Music Education: A creative tool for students to explore harmonic possibilities, understand compositional structure, and experiment with AI-assisted arrangement.
- Personal Content Creation: Enabling hobbyists, podcasters, and content creators to produce original royalty-free music for personal and commercial projects.
- Assistive Technology: Supporting individuals with impairments that limit traditional instrument playing by providing an accessible avenue for musical self-expression.

X. ADVANTAGES

- Structured Sequence Learning: The multi-layer LSTM captures both short-range intervallic relationships and long-range harmonic dependencies, producing compositions with measurable structural coherence.
- Generalizability Across Genres: Training on a multi-genre corpus enables stylistically varied outputs based on the choice of seed sequence.
- Real-Time Generation: The inference pipeline produces MIDI outputs in under two seconds on consumer-grade hardware, facilitating interactive applications.
- MIDI Compatibility: Standard MIDI output ensures immediate compatibility with professional DAWs including Ableton Live, FL Studio, and GarageBand.
- Accessibility: No musical training is required from the end user, democratizing creative music production for a wide audience.

XI. LIMITATIONS

- Monophonic Bias: The primary model is optimized for single-voice melodic generation; polyphonic generation requires more complex multi-stream architectures.
- Dataset Dependency: Output quality is bounded by the training corpus; biases present in data may manifest as stylistic tendencies in generated music.
- Lack of High-Level Structural Planning: The LSTM operates at individual note prediction level and does

not incorporate mechanisms for planning high-level forms such as verse-chorus structures or key modulations.
- Emotional Specificity: The system does not support explicit conditioning on emotional or mood parameters, limiting applicability in emotionally targeted contexts.

XII. FUTURE SCOPE

- Transformer-Based Architecture: Integration of Music Transformer [8] with relative positional attention is expected to yield further improvements in long-range structural coherence.
- Conditional Generation: Incorporating conditioning vectors for genre, mood, tempo, and instrumentation will enable fine-grained stylistic control.
- Multi-Instrument Polyphony: Extending the architecture to model multiple simultaneous voice streams will enable full orchestral arrangement generation.
- Reinforcement Learning from Human Feedback (RLHF): Incorporating human aesthetic judgements as reward signals could substantially improve subjective quality of generated compositions.
- Real-Time Audio Synthesis: Integrating with neural audio synthesis models such as WaveNet [13] will enable high-fidelity audio output directly without external MIDI synthesizers.

XIII. CONCLUSION

This paper has presented MelodAI, a deep learning-based music generation system employing a multi-layered Long Short-Term Memory architecture to learn and reproduce musical patterns from MIDI-encoded training corpora. The system was evaluated against HMM and RNN baselines and demonstrated superior performance, achieving a test accuracy of 92.1%, a perplexity of 8.2, and an average human evaluation score of 4.2 out of 5. These results validate the effectiveness of deep LSTM architectures for capturing the complex temporal dependencies inherent in musical sequences.

MelodAI addresses a genuine and growing need for accessible, intelligent music generation tools applicable across entertainment, therapy, education, and content creation. By combining robust deep learning sequence modelling with a practical MIDI output pipeline and a user-friendly interface, the system demonstrates that AI-driven music composition is technically feasible and practically valuable. The modularity of the architecture ensures that future enhancements, including Transformer models and conditional generation, can be integrated without systemic redesign. Future research will focus on expanding generative range, improving structural planning, and integrating real-time audio synthesis.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Department of Computer Science and Engineering, National Institute of Technology, Bengaluru, for providing computational infrastructure. The authors also acknowledge the creators of the Lakh MIDI Dataset and the Music21 library for their invaluable open-source contributions to music information retrieval and computer-aided musicology.

REFERENCES

[1] L. A. Hiller and L. M. Isaacson, Experimental Music: Composition with an Electronic Computer. New York: McGraw-Hill, 1959.

[2] J. Allan, “Markov models for music,” in Proc. ICMC, 1998.

[3] M. C. Mozer, “Neural network music composition by prediction,” Connection Science, vol. 6, no. 2–3, pp. 247–280, 1994.

[4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] D. Johnson, “Generating polyphonic music using tied parallel networks,” in Proc. EvoMUSART, 2017, pp. 128–143.

[6] J.-P. Briot, G. Hadjeres, and F. Pachet, “Deep learning techniques for music generation – A survey,” arXiv:1709.01620, 2017.

[7] A. Vaswani et al., “Attention is all you need,” in Advances in NIPS, vol. 30, 2017.

[8] C.-Z. A. Huang et al., “Music Transformer: Generating music with long-term structure,” in Proc. ICLR, 2019.

[9] C. Payne, “MuseNet,” OpenAI Blog, Apr. 2019. [Online]. Available: https://openai.com/blog/musenet

[10] C. Raffel, “Learning-based methods for comparing sequences,” Ph.D. dissertation, Columbia University, 2016.

[11] M. S. Cuthbert and C. Ariza, “Music21: A toolkit for computer-aided musicology,” in Proc. ISMIR, 2010, pp. 637–642.

[12] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, 2015.

[13] A. v. d. Oord et al., “WaveNet: A generative model for raw audio,” arXiv:1609.03499, 2016.