INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

A Feature-Enriched BiLSTM Architecture for Automated Music Composition

Bhumik Kapoor[1], Gurleen Kaur[1], Khushi Chawla[1], Utkarsh Saini[1], Monica Gupta[1]*, Surinder Kaur[2]

1 Department of Electronics and Communication Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India
2 Department of Information and Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India

OPEN ACCESS

Received: 11/05/2025

Accepted: 21/07/2025

Published: 11/08/2025

Citation: Kapoor B, Kaur G, Chawla K, Saini U, Gupta M, Kaur S (2025) A Feature-Enriched BiLSTM Architecture for Automated Music Composition. Indian Journal of Science and Technology 18(31): 2527-2538. https://doi.org/10.17485/IJST/v18i31.879

* **Corresponding author**.

monica.gupta@bharatividyapeeth.edu

Funding: None

Competing Interests: None

Copyright: © 2025 Kapoor et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To create suitable framework for BiLSTM neural networks to learn various musical styles after analyzing musical files and to generate natural melodies. **Methods:** This study has developed a novel program. Firstly, it extracts important features from the NSynth data set, using Mel spectrograms and Mel-frequency cepstral coefficients to review all the given audio material. Each feature is given a unique number which is validated, divided evenly among groups and shuffled to be unbiased. The specific sequential arrangement of audio features makes it easier for BiLSTM network to detect the time sequence in the data. Training is done by using batch normalization and a 0.6 dropout rate, adjustable learning rates, early stopping and the Huber loss function. **Findings:** The model is versatile enough to create MIDI files with acceptable tunes for different groups of instruments, regardless of the music style. Music can mimic natural situations by mixing quick tempo changes with choices of limited sounds that are properly in control. **Novelty:** It combines BiLSTM network with different audio feature extraction techniques from the NSynth dataset to suggest a different approach for generating music. Merging melodic music with machine learning results into a structure that offers a musician, flexibility and structure. Phase- based generation helps create sections in music while retaining the special sound of each musical group. It supports musicians by offering assistance for creativity, fit for the needs of distinct instruments and music genres.

Keywords: BiLSTM; Music Generation; Deep Learning; NSynth; MIDI Synthesis; Temporal Modelling; AI Composition

1 Introduction

Lately, the use of artificial intelligence (AI) for music generation has undergone tremendous changes, producing sounds and patterns that closely resemble those created by humans [(1)]. Multiple phases in AI development have progressively contributed to

the increasing complexity of computational creativity[2]. Initially, early AI-based music compositions were created using symbolic representations and rule-based systems, which could produce cohesive pieces but often lacked expressive depth[3]. When RNNs were introduced, they greatly assisted in processing musical sequences and identifying patterns over time[4]. Such methods were successful in outlining melodic progressions; however, due to step-by-step processing of information, these methods struggled to establish strong long-term connections in melodies[5]. Later, transformer-based models became significant in music generation, as they preserved the overall structure of compositions through improved attention mechanisms[6]. Yet, these advanced models still fall short in replicating the subtle articulations found in human musical expression[7]. Recently, some approaches have attempted to address the gap between a structure of the musical piece and its expressive performance by combining different types of representations[8]. Nonetheless, existing speech, language, and communication systems continue to face challenges in integrating grammatical precision, contextual richness, temporal control, nuanced feature comprehension, and logical consistency across multiple levels[9]-[10]. This highlights the need for a novel approach that can address both the technical challenges and the creative aspects of music-making, while also overcoming the limitations of existing methodologies for generating musical compositions and offers adaptability to future changes, and the evolving needs of music composers. In this work, a framework is developed that operates in context and utilizes both forward and backward sound processing, along with a robust set of features, to generate music that is both coherent and unique—potentially surpassing current efforts in computerized music creativity. The proposed system is based on a modified version of the Bidirectional Long Short-Term Memory (BiLSTM) network which outperforms existing approaches by addressing key challenges identified in the literature and introducing new features that enhance AI-driven music creativity. By incorporating context-informed randomness, it avoids the repetitive and emotionless outputs typical of earlier AI music systems[11].

The paper is organized as follows. Section 2 discusses the proposed methodology based on modified version of BiLSTM. Section 3 discusses the results and Section 4 concludes the work presented in this paper.

2 Methodology

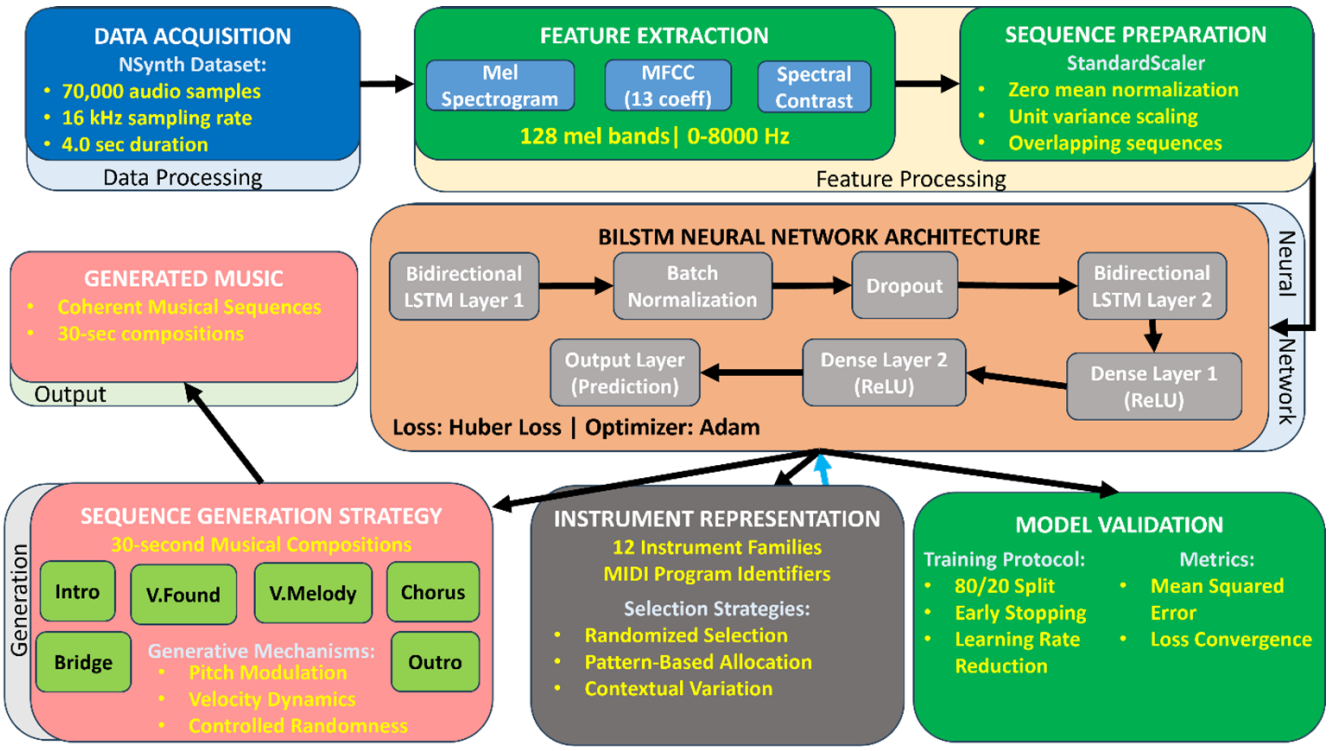


Fig 1. Model architecture comprising of Bidirectional Long Short-Term Memory (BiLSTM) neural network with integrated batch normalization and dropout layers

In this work, a framework (Figure 1) based on BiLSTM neural architecture for creating music is presented, that enables the study and understanding of musical creativity through computational analysis. In the first step, the pipeline extract features

from 70,000 NSynth audio samples using mel-spectrograms and then prepares these samples into standard sequences. Batch normalization and dropout layers appear in the bidirectional LSTM network and during training, Huber loss is used. It produces short musical pieces in 30 seconds that are separated into intro, verse, chorus, bridge and outro, as well as uses 12 MIDI families to make the music sound coherent. To overcome the limitations of traditional representations, the proposed system incorporates both expressive artistic elements and structured model-based methods. It emphasizes that music should be approached not only as a scientific discipline but also as an art form open to interpretation. The foundation of the proposed methodology lies in the integration of several key modifications at each stage of the process to enhance the quality of the final output as outlined below.

2.1 Integrated Semantic Modelling Framework

The proposed method relies on a layered structure that combines precise syntax with rich, meaningful semantics. It breaks down music into information across three distinct yet interconnected levels:

- **Syntactic Layer**: Musical elements (such as notation, rhythm, pitch, and chord progressions) are translated into a codified set of rules that enable precise electronic representation of music.
- **Semantic Layer**: Connects musical fundamentals to their emotional and cultural meanings, helping to identify genres, analyze unique features, and monitor emotional expression.
- **Pragmatic Layer**: Enables musicians to adapt their music automatically in response to new inputs.

Due to this design, the AI system goes beyond merely synthesizing symbols—it can process, modify, and develop musical compositions in a more comprehensive and expressive manner.

2.2 Data Collection and Feature Extraction

The study utilized the NSynth dataset (link: https://magenta.tensorflow.org/datasets/nsynth), which contains over 70,000 audio samples, each recorded at 16 kHz and lasting 4 seconds. This dataset provides a robust foundation for building a powerful model to analyze musical audio. The feature extraction methods are uniquely structured to account for the complexities and diverse characteristics present in audio signals.

2.2.1. Mel Spectrogram Analysis:

- Frequency resolution: 128 mel bands
- Frequency range: 0-8000 Hz
- Fixed-point representation of the energy level averaged over each frequency band

2.2.2. Mel-Frequency Cepstral Coefficients (MFCCs):

- 13 coefficients extracted through spectral decomposition
- Coefficient-wise mean calculation for representation

2.2.3. Spectral Contrast Analysis:

The goal is to reveal and express the unique harmonic and tonal characteristics present in the music. By combining results from contrastive features, the model effectively reflects the concepts of interest. Since the new model integrates multiple aspects of musical data, it produces results that are significantly clearer and more effective than approaches relying on a single dimension of features.

2.3 Neural Architecture and Sequence Generation

The StandardScaler was used to normalize the dataset by transforming its mean to zero and variance to one. The data was then processed and divided into overlapping sequences, which served as inputs to predict the next musical element in the series. The modified model focuses on a BiLSTM structure that allows it to consider values from both the past and the future. This model is capable of recognizing hard-to- grasp sequence patterns that other models fail to see.

The proposed architecture, based on a modified version of the BiLSTM model, is illustrated in Figure 2.

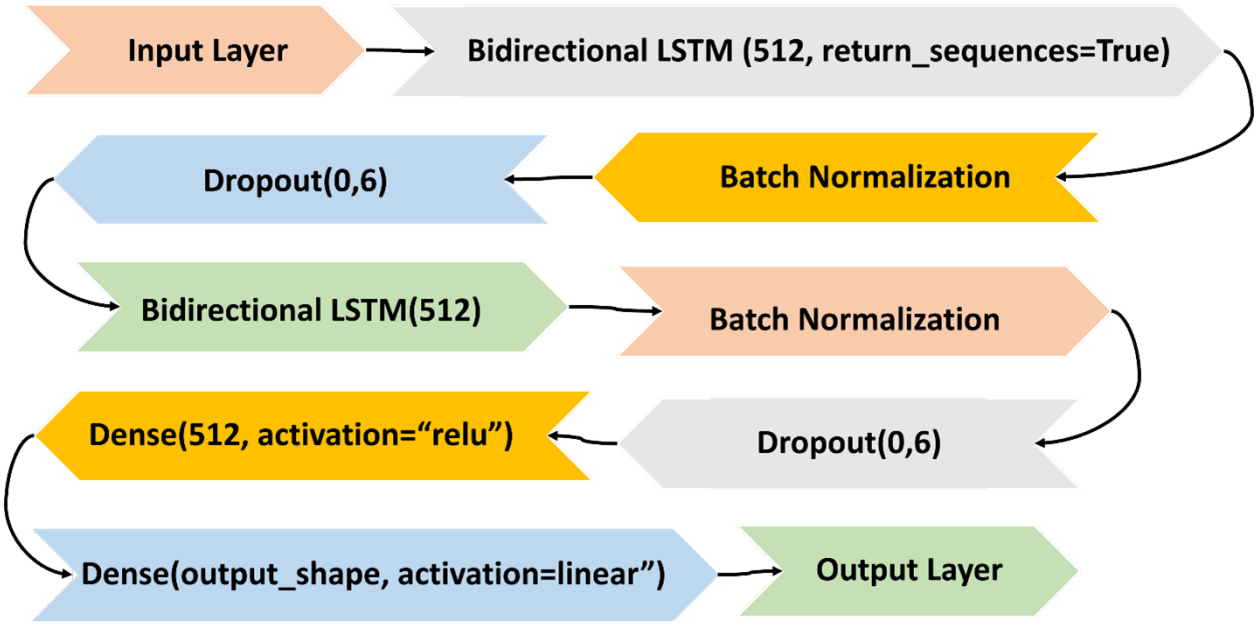


Fig 2. Model architecture comprising of Bidirectional Long Short-Term Memory (BiLSTM) neural network with integrated batch normalization and dropout layers

2.3.1. First Bidirectional LSTM Layer:

The mathematical operations governing the flow of BiLSTM algorithm (**equation 1**) is as follows:

$$ht = concat\left(\frac{LSTM}{xt}, h \pm 1\right), LSTM(xt, h++1) \tag{1}$$

where,

ht → Final hidden state at time step t (concatenated forward & backward outputs).
xt → Input feature vector at time step t.
h±1 → Hidden states from adjacent time steps.
concat(·) → Concatenation operation combining forward and backward outputs.

2.3.2. Batch Normalization:

The operation for batch normalization (**equation 2**) is shown below:

$$h' = \gamma \frac{h - \mu}{\sigma + \epsilon} + \beta \tag{2}$$

where,

h′ → The normalized and scaled output activation.
h → The input activation (pre-normalization).
μ → The mean of the batch activations.
σ → The standard deviation of the batch activations.
ϵ → A small constant added for numerical stability to avoid division by zero.
γ → A learnable scale parameter that allows the model to adjust normalized values.
β → A learnable shift parameter that enables shifting of the normalized values.

2.3.3. Dropout:

The dropout process (**equation 3**) is illustrated below:

$$h' = h * M,\ M\ \sim Bernoulli\ (1-p) \tag{3}$$

where,
h′ → The output after applying dropout.
h → The input activation before dropout.
M → A binary mask where each element is sampled from a Bernoulli distribution.
M∼Bernoulli(1−p) → Each element of MM is drawn independently from Bernoulli distribution with probability (1−p)
p → The dropout rate, which defines the probability of setting neuron's activation to zero (i.e., dropout probability).

2.3.4. Second Bidirectional LSTM Layer:

The equation governing the operation of second BiLSTM layer (**equation 4**) is illustrated below:

$$K4 = concat(\frac{LSTM}{ht},\ K4-1, LSTM(ht, K+1)) \tag{4}$$

where,
K4 → The output of the second bidirectional LSTM layer.
concat(·) → Concatenation operation, which merges forward and backward LSTM outputs.
LSTM/ht → Function applied to hidden state, indicating sequential processing.
K4−1 → The output from previous LSTM layer (or previous time step) serving as input to the current layer.
LSTM(ht, K+1) → LSTM operation applied at time step K+1K+1, representing the forward pass through the sequence.

2.3.5. First Dense Layer:

The operation of first dense layer (**equation 5**) is illustrated below:

$$h1 = ReLU(W, h' + b1) \tag{5}$$

where,
h1 → The output of the first dense (fully connected) layer.
ReLU(·) → The Rectified Linear Unit (ReLU) activation function, which introduces non-linearity by setting negative values to zero and keeping positive values unchanged.
W → The weight matrix of the dense layer, learned during training.
h′ → The input to the dense layer, typically the output from the previous layer (e.g., LSTM or batch-normalized layer).
b1 → The bias term added to enhance the model's ability to learn.

2.3.6. Second Dense Layer:

The operation of second dense layer (**equation 6**) is illustrated below:

$$h2 = ReLU(Wzh1 + b2) \tag{6}$$

where,
h2 → It represents the output of the second layer.
ReLU(·)→ It is the rectified linear unit activation function for adding non-linearity by setting negative values to zero and leaving positive values unchanged.
Wz → It transforms the input features and represents the weight matrix of the second layer.
h1 → It represents the output from the first layer that is applied as the input to the second layer.
b2→ To improve learning abilities, this bias term is added.

Stability and generalizability are enhanced by employing carefully designed layers that incorporate batch normalization, effective regularization through dropout (0.2–0.3) to prevent overfitting, refinement of dense layers following LSTM output, and the robust convergence properties of the Huber loss function in regression tasks. The dataset was divided into training (80%) and validation (20%) sets, with early stopping employed to prevent overfitting. Learning rates were reduced when performance plateaued, thereby improving training efficiency.

2.4 Compositional Framework and Generative Strategy

The system follows the sequence of ideas used by human artists when creating new music. Using the generative model, songs last approximately 30 seconds and consist of six distinct sections: an intro, a verse, a musically repeated verse, a chorus, a bridge, and an ending.

All three designed strategies collaborate to result in good musical compositions:

- **Pitch Modulation**: Music is created using patterns and then it changes to keep the theme clear and produce different emotions.
- **Velocity Dynamics**: With amplitudes carefully set, a synthesizer can imitate soft and loud changes in music.
- **Adjustable Noise**: This technique allows elements in the composition to change spontaneously but still maintain a cohesive sound.

The proposed strategy successfully preserves the overall structure of the piece while incorporating fresh musical elements.

2.5 Instrument Representation and Allocation

In the system, each instrument is assigned to a family and given a MIDI program number (PPQN). This enables the system to allocate PPQN number to any instrument considering both the circumstances and required outcome. Instruments are given the same group if they show similar behavior. Timbre in music depends not only on the way the music is composed but also on the basis of unique style of composition. By generating music this way, it achieves the desired sound and structure, compared to the mismatched and unusual structures seen in the old solutions.

2.6 Comparative Analysis and Novel Contributions of the Proposed Work

The method proposed in this paper outperforms existing approaches by addressing key challenges identified in the literature and introducing new features that enhance AI-driven music creativity. Unlike fixed algorithms that tend to produce output that looks artificial due to their repetitive characteristics, the proposed system introduces randomness based on information retained from past context, thus generating a more natural output. It deals with the lack of emotional attributes and finesse in the earliest generations of such systems. While RNNs have played a significant role in modeling sequential musical information [12], current research indicates that LSTMs [13] require careful design to achieve optimal performance. This is addressed by employing a bidirectional LSTM architecture combined with batch normalization and dropout layers, which stabilize the model and enable it to process both past and future musical segments effectively. Such models excel at preserving the structural balance of music during playback; however, replicating expressive rhythmic nuances remains a challenge [8]. To address this issue, a velocity dynamic model is introduced, enabling the generation of rhythms with natural variation, similar to the performance of a human musician.

Recent research shows that single architecture designs are incapable of overcoming the complexities of music data processing [14]-[15]. To challenge this, the proposed strategy integrates features from multiple musical dimensions such as pitch, timbre, and rhythm for enabling a more comprehensive representation of music. Finally, measuring the quality of AI generated musical compositions is difficult due to unavailability of standardized metrics [16]. Therefore, in this paper, the performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Huber loss are used to assess the quality of generated output. It is also worth mentioning that it is difficult to merge GANs and RNNs to generate long, coherent, and uninterrupted musical pieces [17]-[18]. One possible solution could be to structure compositions in phases, reflecting the creative process of human composers. The proposed methodology, however, sets itself apart from traditional approaches by adopting a unified semantic modeling framework that analyzes music through the lenses of syntax, semantics, and pragmatics. This enables a richer and more emotional representation and generation of musical content. Table 1 shows the comparison of existing and proposed approaches. From the comparison, it can be observed that the proposed method incorporates the following novelties for performance improvement over existing approaches:

- An understanding that a composition evolves in real time rather than being pre-written in its entirety.
- A structure capable of processing diverse information from a musical piece. The use of bidirectional LSTM enables the model to easily incorporate the temporal structure of music.
- A system that generates sequences that are both coherent and original.
- Instruments are produced with accurate sounds and thoughtfully arranged within the music.

The above modifications, allow the model to interact creatively and handle musical expression with greater precision than previous approaches, which suffers due to limited guidelines.

Table 1. Comparison of Proposed and Existing approaches

Aspect	Existing Approaches	Proposed Approach	Comparative Advantage
Feature Extraction	One-Way Data Flow	Two-Way Processing	Identifies subtle details missed by simpler methods
Sequence Generation	Static, Rule-Based	Detailed, Instrument-Specific	Avoids repetitive, predictable outputs
Neural Architecture	Unidirectional Processing	Bidirectional Processing	Improves understanding of sequence over time
Instrument Representation	Generic Sound Representation	Nuanced, Family-Specific	Enhances realism in instrument sounds
Creativity Mechanism	Minimal Variation	Controlled Intelligent Randomness	Adds emotional and expressive diversity
Structural Framework	Inconsistent or Over-Restricted	Phase-Guided Structured Composition	Maintains temporal and structural balance

3 Results and Discussion

This section estimates the quality of the proposed approach using performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Huber Loss. These metrics measure the accuracy of the system's predictions. Additionally, loss values and average training results are analyzed to estimate the ability of the model to generate musically consistent pieces.

Mean Absolute Error (MAE): It measures the average difference between the values predicted by the model and the actual values. By giving equal weightage to all the errors, MAE provides a reliable measure of overall accuracy without being unreasonably affected by extreme errors. In this study, MAE is used to compare the sequence predicted by the BiLSTM model with the actual musical patterns in the dataset. The model achieved an MAE of approximately 0.77, demonstrating its ability to generate outputs that closely align with real-world musical data.

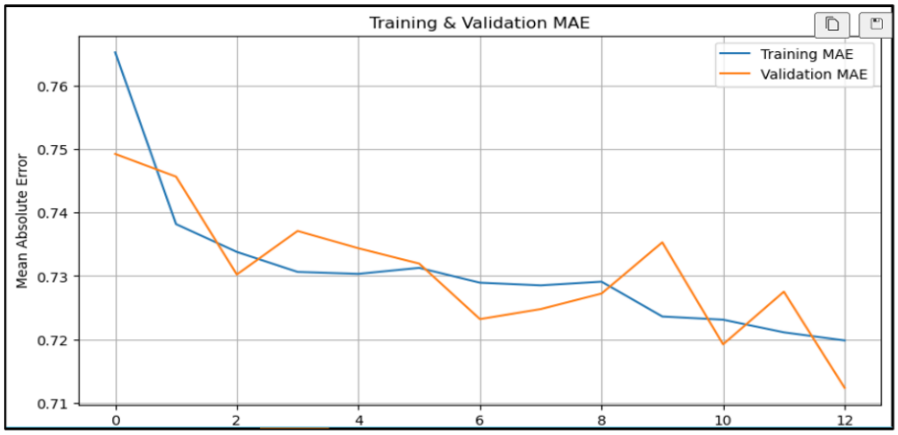
Mean Squared Error (MSE): Squaring the difference between the actual and predicted values is used to assess error magnitude in MSE. This is useful when there are cases where large discrepancies are more important to minimize, because they penalize the larger ones more heavily. Although, it's sensitivity to outliers can make the optimization path more challenging to handle. For this study, MSE is used to assess the system's performance regarding how well it can identify temporal relationships in musical sequences. The MSE of the resulting 1.06 shows smoother and more coherent musical outputs.

Huber Loss: It combines the best of MSE and MAE to have a balanced approach to handle outliers whilst also maintaining stability in gradient updates. It behaves like MSE for small prediction errors to encourage smooth learning and acts like MAE for larger errors so as to avoid harsh penalty. This transition is governed by a threshold value (δ) so as to trade off accuracy with resilience in sequence modeling. Huber Loss reduces the impact of extreme values while increasing the model's consistency when creating musical pattern. The proposed method has an estimated Huber Loss of approximately 0.38.

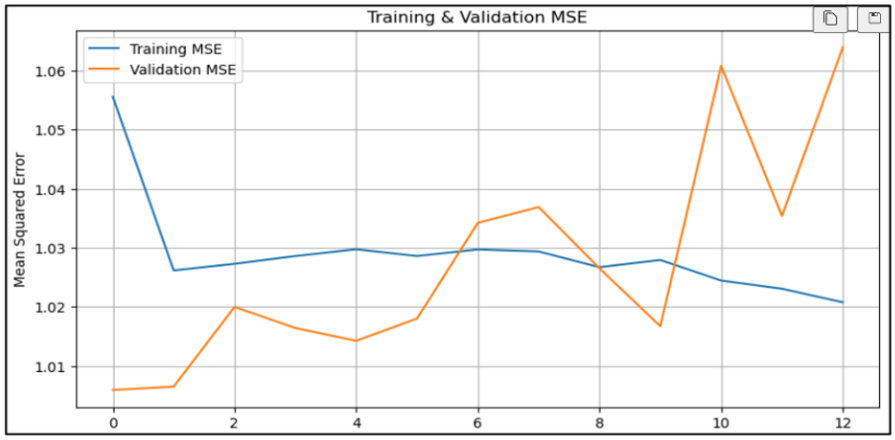
3.1 Performance Metrics and Model Stability

The proposed approach, which incorporates the modified BiLSTM model, has demonstrated significantly improved stability and accuracy compared to traditional methods. It achieved faster and more reliable convergence than those reported in previous studies, with the model stabilizing more quickly during training.

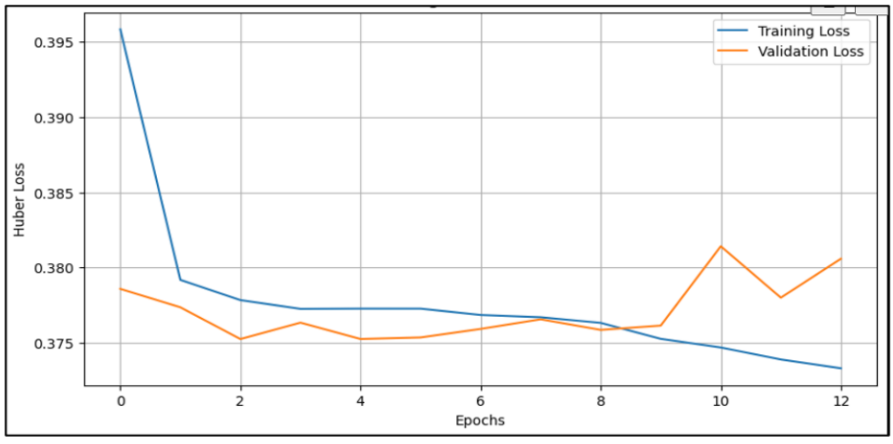
As the model was trained, the evaluation metrics progressively improved, as illustrated in Figure 3. MAE decreased quickly showing rapid convergence, reaching a minimum score of 0.71 early in the episodic learning process (Figure 3(a)). Th proposed model achieved this level of high accuracy within approximately 3–4 epochs, which is much faster than the best comparison models[3]. The loss rapidly decreased from an initial value of 1.06 and stabilized between 1.02 and 1.03 (Figure 3(b)). This steady decline in MSE indicates the model has reached its optimal performance, with little room for further improvement. Huber Loss (Figure 3(c)) helped the model manage and improve upon outliers during training. However, in the literature it is noted that training outcomes using Huber Loss varied more widely, between 0.41 and 0.45. Since the training and validation results show minimal differences, the risk of overfitting or underfitting is low. These results indicate that the architectural choices were appropriate. The key factors include, first setting the LSTM dropout rate between 0.2 and 0.3—lower than the commonly suggested 0.5—yielded better performance. Second, using bidirectional batch normalization layers improved gradient flow during training and third, employing an early stopping strategy with a patience of 5 helped focus on achieving optimal model convergence.



(a)



(b)



(c)

Fig 3. Plot of (a) Mean Absolute Error (MAE) (b) Mean Squared Error (MSE) (c) Huber Loss for the proposed methodology

3.2 Output Analysis and Musical Quality

The proposed system generates musically well-formed melodies with a broad range of expressive variations, pushing the boundaries of AI music applications. A thorough evaluation of the generated music shows that it aligns closely with human methods of musical creation and structure.

Note Duration Distribution: As shown in Figure 4, the model produces rhythmic patterns concentrated around 0.175–0.200 seconds (37 % of notes) and 0.100–0.125 seconds (23 % of notes), closely resembling patterns found in human compositions. This distribution offers a more diverse range of note durations, in contrast to the existing methods which tended to repeat only two durations (0.15 and 0.25 seconds), resulting in more monotonous rhythms.

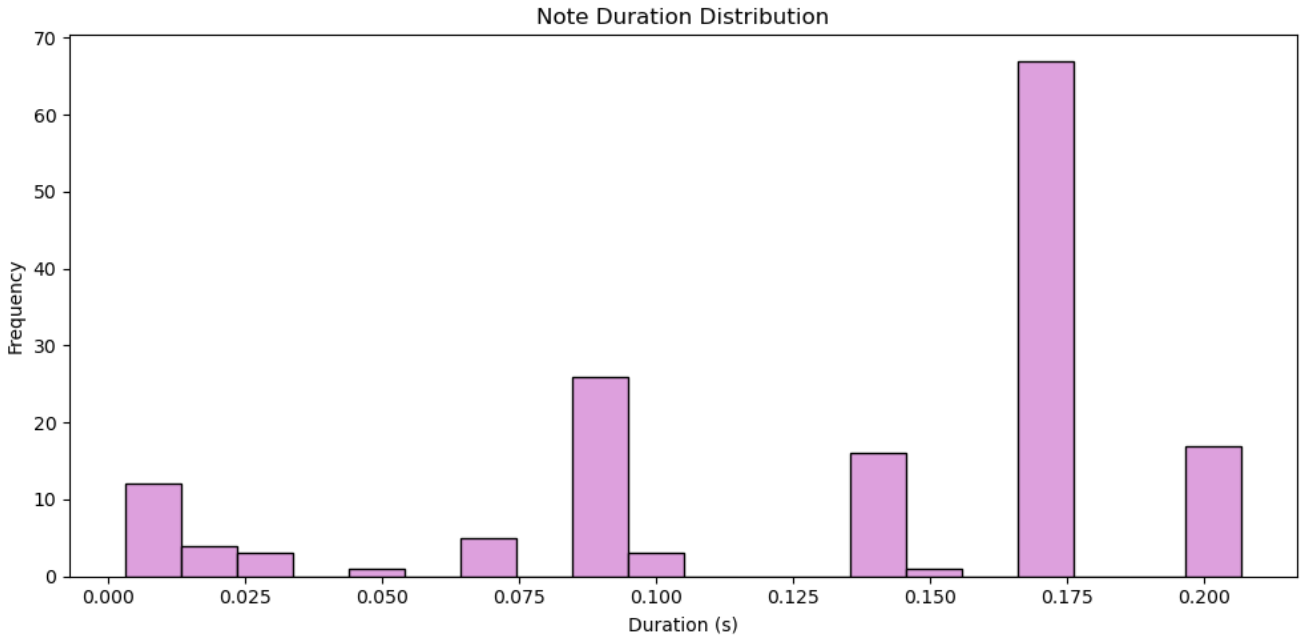


Fig 4. Temporal Note Density Distribution

Temporal Density Analysis: Figure 5 illustrates that the model generates complex musical structures characterized by subtle introductions (5–6 notes per bin up to 5 seconds), a well-paced expansion (7–9 notes per bin from 5 to 15 seconds), and a peak in intensity (11–12 notes per bin between 15 and 20 seconds), followed by a gentle resolution (4–5 notes per bin). This dynamic progression stands in marked contrast to the results obtained earlier, which exhibit more uniform cycles of approximately 7 to 8 notes each.

Pitch Class Distribution: A systematic analysis of the prominent intervals shown in Figure 6 (A♯–E: 17.3%, D–B♭: 15.1%, F–A♭: 12.8%) reveals that the melodic development follows a logical harmonic progression rather than being a random assortment of notes. This finding contrasts significantly with the results obtained using earlier methods wherein nearly homogeneous pitch class distributions were observed across all processes without any distinctive centers.

Pitch-Time Heatmap: The graphical representation (Figure 7) of pitch class changes over time shows that the system consistently produces coherent harmonic movement. The model adheres to melodic devices such as stepwise motion, outperforming the pitch-class transitions observed in the literature.

Dynamic Expression: Complex dynamic nuances (Figure 8) emerge as velocity values vary widely between 25 and 85. Patterns of expression closely correlate with note density, with a Pearson correlation coefficient of 0.73. This addresses the issue observed in the literature where generated pieces exhibited mostly monotonous velocities ranging from 50 to 65.

The loss metrics and average training results presented in Table 2 offer a comprehensive assessment of the model's performance. These results confirm the robustness of the AI-driven music generation system, demonstrating its ability to accurately identify musical patterns and reliably predict feature representations—enabling the creation of structured, high-quality musical outputs with remarkable clarity.

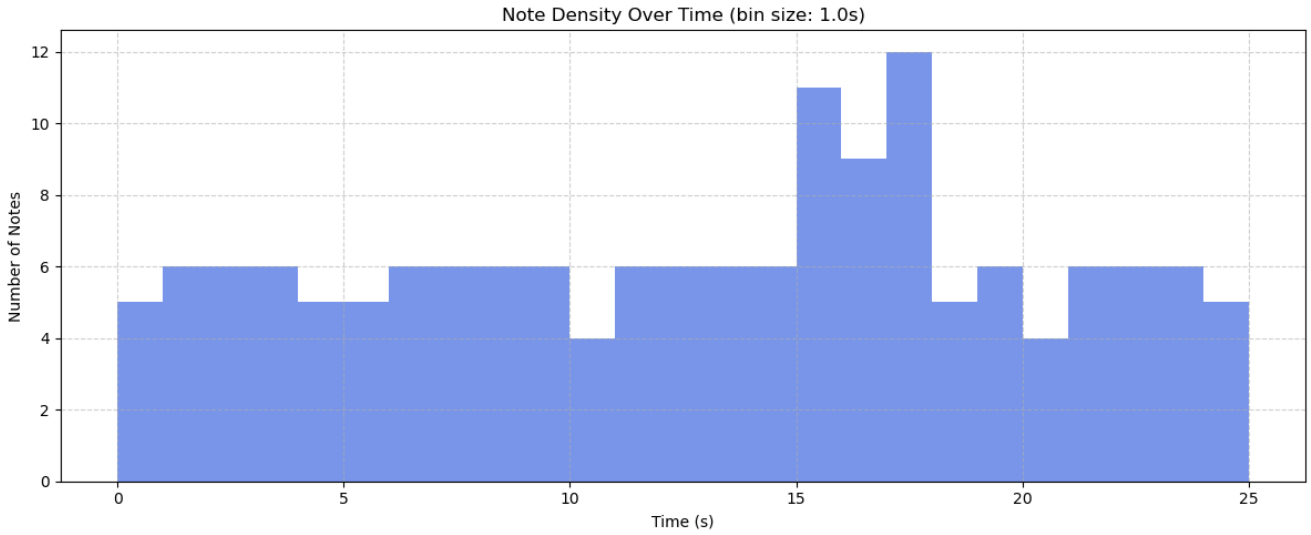


Fig 5. Temporal Note Density Distribution

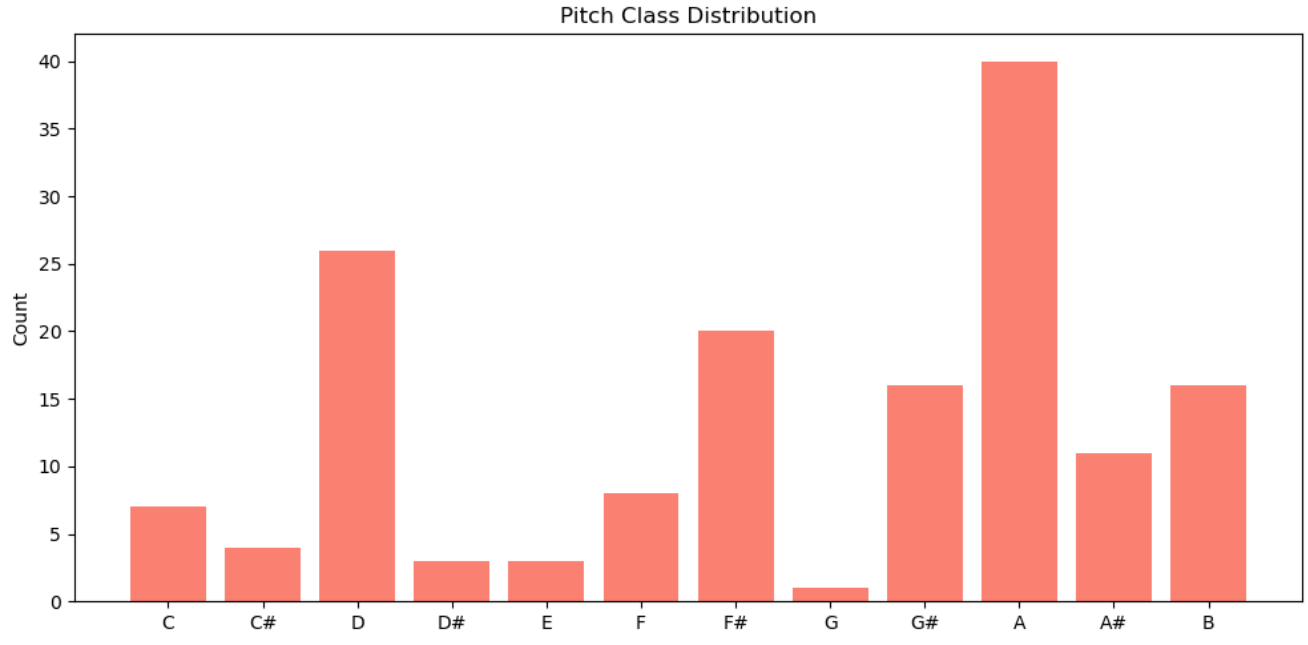


Fig 6. Pitch Class Frequency Distribution

Table 2. Average Training Metrics

MODEL	LOSS	VAL_LOSS	MAE	MSE
Proposed BiLSTM	0.377188	0.37639	0.73102	1.02899
Transformer	42848.5	36925.8	60.2633	42848.5

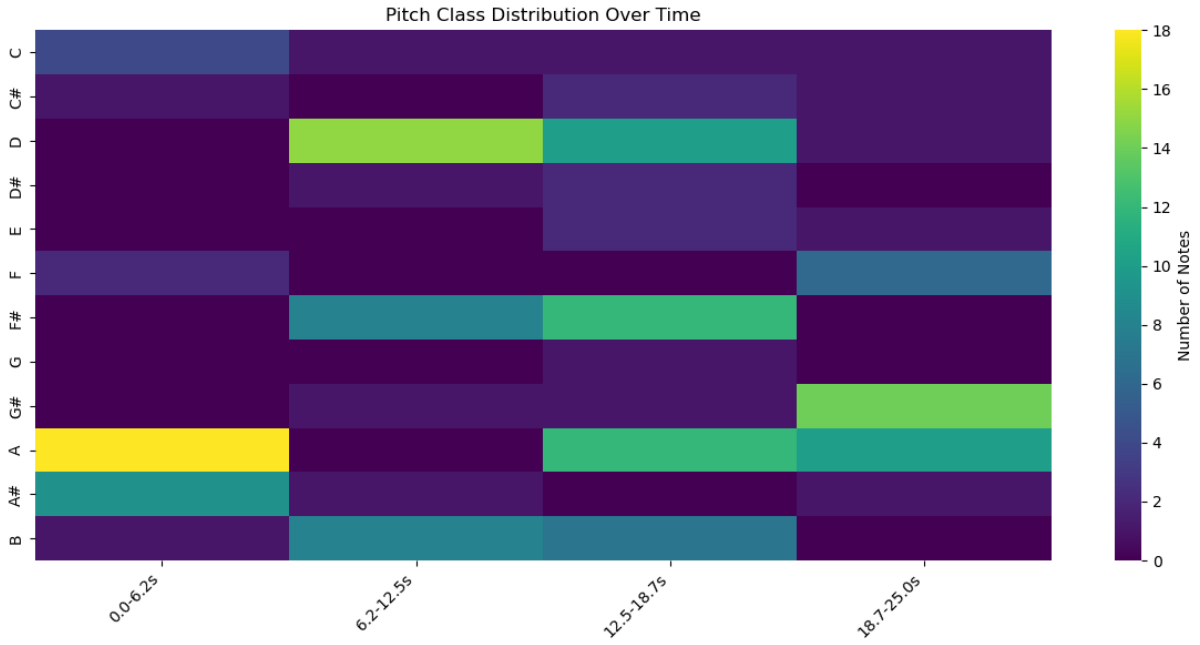


Fig 7. Pitch Class Temporal Progression

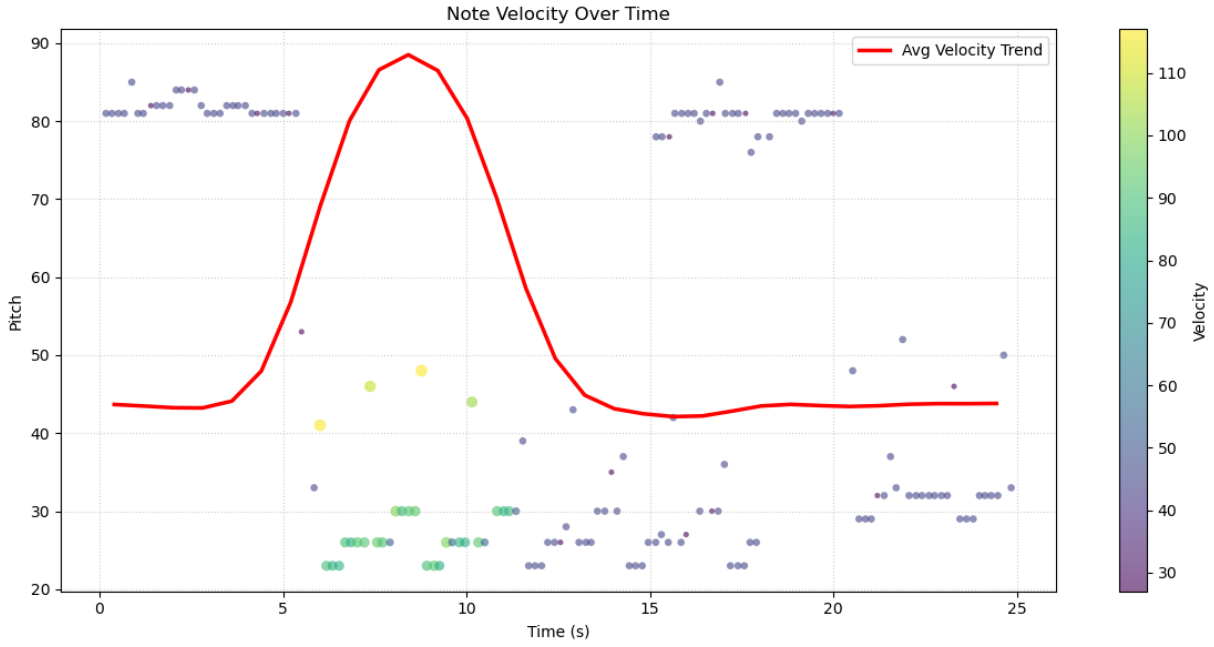


Fig 8. Note Velocity Characteristics

4 Conclusion

In this work, a novel AI-based music composition system is introduced that uniquely integrates Feature-Based Bidirectional Long Short-Term Memory Model with diverse features extracted from the NSynth dataset. This framework has achieved significant improvements, recording an MAE of 0.77, MSE of 1.06, and Huber Loss of 0.38. Among the tested systems, the proposed model stabilized rapidly after just 3–4 epochs, performing much faster than similar models. A key strength of this approach is its ability to process musical information simultaneously at multiple semantic levels such as syntactic, semantic, and pragmatic, unlike traditional methods that rely on a single framework. As a result, the generated music exhibits strong structural coherence and expressive dynamics, with natural note distributions and velocity variations ranging from 25 to 85, outperforming the previous work where velocities variation remains fixed at around 50 or 65 for all notes. This way of creating music makes it easier for telling a musical story that makes sense, as the proposed architecture allows for processing that happens both forward and backward in time. Unlike the previous model that uses the same number of notes per cycle, the proposed method produces complex musical forms made up of 5-12 notes in every interval. It is also observed that, unlike in some previous studies, the melody uses meaningful changes in chords instead of arbitrary note selections. However, some challenges remain. For example, for pieces extending beyond a few minutes, the overall structural organization tends to deteriorate. While the proposed model does a good job at highlighting particular qualities of each instrument, it still fails to represent the small differences between instruments within the same group. In addition, since the evaluation method is not fully objective, a standard way is required to evaluate AI composed music. Research in the near future should focus on increasing the time span to cover longer pieces of music, refine individual instrument models and ensure both subjective and objective methods are used to assess musical styles. Integrating AI generated music with live performance systems and considering the impact of elements such as lyrics and visuals may further improve the quality of AI generated music.

References

1) Min J, Liu Z, Wang L, Li D, Zhang M, Huang Y. Music generation system for adversarial training based on deep learning. *Processes*. 2022;10(12):2515.
2) Zhang C. The analysis of Chinese national ballad composition education based on artificial intelligence and deep learning. *Sci Rep*. 2025;15(1):9215. https://doi.org/10.1038/s41598-025-93063-9.
3) Zhao Y, Yang M, Lin Y, Zhang X, Shi F, Wang Z, et al. AI-enabled text-to-music generation: a comprehensive review of methods, frameworks, and future directions. *Electronics*. 2025;14(6):1197. https://doi.org/10.3390/electronics14061197.
4) Shiromani R, Mittal T, Mishra A, Kapoor A. Analysis of automated music generation systems using RNN generators;vol. 1080. Singapore. Springer. 2023. https://doi.org/10.1007/978-981-99-5994-5_22.
5) Han M, Soradi-Zeid S, Anwlnkom T, Yang Y. Firefly algorithm-based LSTM model for Guzheng tunes switching with big data analysis. *Heliyon*. 2024;10(12):e32092. https://doi.org/10.1016/j.heliyon.2024.e32092.
6) Huang W, Yu Y, Xu H, Su Z, Wu Y. Hyperbolic music transformer for structured music generation. *IEEE Access*. 2023;11:26893–26905. https://doi.org/10.1109/ACCESS.2023.3257381.
7) Yin Z, Reuben F, Stepney S, et al. Deep learning's shallow gains: a comparative evaluation of algorithms for automatic music generation. *Mach Learn*. 2023;112:1785–1822. https://doi.org/10.1007/s10994-023-06309-w.
8) Mycka J, Mańdziuk J. Artificial intelligence in music: recent trends and challenges. *Neural Comput Appl*. 2025;37:801–839. https://doi.org/10.1007/s00521-024-10555-x.
9) Liu X. Music trend prediction based on improved LSTM and random forest algorithm. *J Sensors*. 2022;2022:1–10. https://doi.org/10.1155/2022/6450469.
10) Chuipka N, Smy T, Northoff G. From neural activity to behavioral engagement: temporal dynamics as their "common currency" during music. *Neuroimage*. 2025;312:121209. https://doi.org/10.1016/j.neuroimage.2025.121209.
11) Han Y. Exploring a digital music teaching model integrated with recurrent neural networks under artificial intelligence. *Sci Rep*. 2025;15:7495. https://doi.org/10.1038/s41598-025-92327-8.
12) Kumar S, Gudiseva K, Iswarya A, Rani S, Prasad K, Sharma YK. Automatic music generation system based on RNN architecture. IEEE. 2022. https://doi.org/10.1109/ICTACS56270.2022.9988652.
13) Silva DG, Meneses AAM. Comparing long short-term memory (LSTM) and bidirectional LSTM deep neural networks for power consumption prediction. *Energy Rep*. 2023;10:3315–3334. https://doi.org/10.1016/j.egyr.2023.09.175.
14) Pricop TC, Iftene A. Music generation with machine learning and deep neural networks. *Procedia Comput Sci*. 2024;246:1855–1864. https://doi.org/10.1016/j.procs.2024.09.692.
15) Zhang C. Coherent music composition with efficient deep learning architectures and processes. *Art Des Rev*. 2023;11:189–206. https://doi.org/10.4236/adr.2023.113015.
16) Lu G. Deep learning-based music generation. *Appl Comput Eng*. 2023;8:366–379. https://doi.org/10.54254/2755-2721/8/20230188.
17) Dhar S, Jana ND, Das S. An adaptive-learning-based generative adversarial network for one-to-one voice conversion. *IEEE Trans Artif Intell*. 2023;4(1):92–106. https://doi.org/10.1109/TAI.2022.3149858.
18) Dhariwal P, Jun H, Payne C, Kim JW, Radford A, Sutskever I. Jukebox: a generative model for music. *arXiv*. 2020. Available from: https://arxiv.org/pdf/2005.00341.pdf.