Hybrid Machine Learning For Musical Style Recognition And Composer-Aware Music Generation

Xin Xiong*

[1] School of Humanities and Education, Wuchang Institute of Technology, Hubei 430065, China
* Corresponding author. E-mail: happybearxin@163.com

Received April 24, 2026; accepted July 8, 2026

The rapid growth of digital music collections has increased the demand for automatic composer identification and musical style recognition. This study proposes a hybrid framework using the MAESTRO v3.0.0 symbolic MIDI dataset. EB-SMR preprocesses MIDI by extracting pitch, onset, duration, and velocity, followed by tempo, pitch, and duration normalization. The data are converted into piano-roll representations. Handcrafted symbolic features capturing melodic, rhythmic, and dynamic properties are combined with Convolutional Neural Network (CNN) -based deep features, enabling improved feature learning for composer-aware music generation and style recognition tasks. These complementary features are fused into a unified representation and classified using a Hybrid Feature-Based Softmax Neural Network (HFSNN), while the Lion optimizer is employed as a training strategy to improve convergence efficiency during optimization. The framework incorporates composer-aware music generation by sampling musical events from class-conditional distributions to produce stylistically coherent MIDI sequences. Experimental results show that the hybrid framework achieves 0.97, outperforming CNN-only and handcrafted-feature models. Combining symbolic descriptors and CNN features improves style/composer recognition, enables composer-aware MIDI generation, but is limited to piano MIDI; future work explores multimodal and multi-instrument representations.

Keywords: Symbolic Music Analysis, Musical Instrument Digital Interface, Hybrid Feature Extraction, Piano-Roll Representation, Composer Classification

http://dx.doi.org/10.6180/jase.202611_34.051

1. Introduction

The rapid growth of digital music collections, which are now primarily kept in the cloud, many users have begun creating those collections and there is an increasing demand for automated techniques to categorize and organize music [1]. The use of these automated classification systems can improve users' music listening experiences, music recommendation systems, and music discovery processes [2]. In classical music, the individual identities of composers are described by a combination of their musical styles based on different musical styles including rhythm, pitch, and melody [3].

Automatic musical style and composer recognition is challenging due to the structural complexity of classical music, including diverse rhythms, harmonies, and dynamics [4]. The features are combined and classified using a hybrid feature-based softmax neural network, while the Lion optimizer is employed solely to support efficient model training and convergence [5].

Beyond music, hybrid machine learning models are widely used in engineering and predictive [6] modelling, including structural [7] optimization, material property prediction, asphalt stability estimation, bond strength analysis, and buckling resistance prediction [8]. These applications demonstrate their ability to capture complex nonlinear relationships [9], motivating robust feature fusion and data-

driven learning frameworks across domains [10].

Existing studies have demonstrated promising results in musical style recognition and composer-aware generation. However, many approaches rely exclusively on either handcrafted symbolic features or deep learning representations, limiting their ability to capture both explicit musical knowledge and complex hierarchical patterns simultaneously. In addition, most studies focus on classification or generation as separate tasks.

The primary contribution of this study lies in the integration of handcrafted symbolic musical features and CNN-based deep representations within a unified hybrid learning framework for musical style recognition and composer-aware generation. The Lion optimizer is employed only as a training strategy to improve convergence efficiency and does not constitute the principal methodological innovation of the proposed approach.

2. Literature review

Several previous works have focused on automatic musical style recognition and composer-aware generation using deep learning. A few of them are mentioned here, Shi [11] have suggested a CNN-based classical music recognition model with the objective of identifying music title, style, and emotional content from audio signals. Limitations include dataset dependency, sensitivity to complex classical structures and limited generalization across unseen composers and recording conditions. Turchet and Pauwels [12] have suggested that human-centered evaluation was critical to evaluating AI-generated music through loss metrics. Wu and Yang [13] have suggested integrating graph convolutional neural networks (GCNs), graph attention networks (GATs) and Long Short-Term Memory (LSTM) for automatic multi-label classification of piano sonatas. Its objective was to capture structural and temporal musical features more effectively, addressing traditional neural networks' limitations in representing complex musical dependencies.

Zhang [14] have suggested integrating Transformer decoders with a variational autoencoders (VAE) to enable bar-level controllable music generation. Its objective was long-sequence style transfer with rhythmic and polyphonic control. Limitations include reliance on symbolic data, training complexity, and constrained generalization across genres domains. Zhang et al. [15] proposed MIDI segment analysis with feature extraction and attention-based RNNs for music classification. Studies on AI-generated music highlight benefits, challenges, and AIMT development, though limited participants reduce generalizability.

Despite progress in AI-based music style recognition and composer-aware generation, challenges remain with diverse audio/symbolic data, poor generalization to unseen composers, heavy reliance on datasets, high computational cost, and subjective evaluation. Hence, hybrid machine learning models are needed.

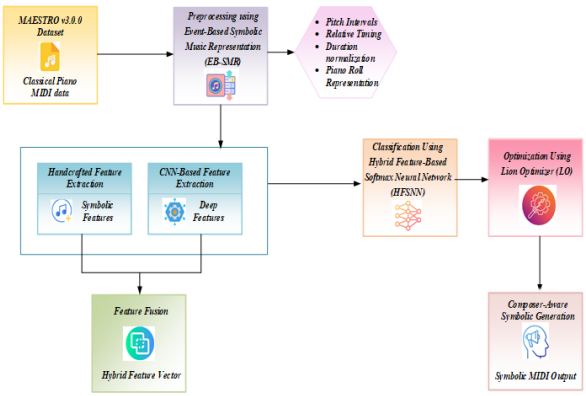

Fig. 1. Block Diagram of the Proposed Method

- Develop an event-based symbolic preprocessing for expressive MIDI music representation.
- Extract complementary symbolic and CNN-based deep musical features.
- Fuse hybrid features for robust composer and musical style discrimination.
- Design HFSNN for simultaneous composer identification and style recognition.
- Employ the Lion optimizer as an efficient training mechanism to support stable convergence during model optimization.
- Generate composer-aware symbolic MIDI sequences by conditioning on hybrid feature embeddings to ensure stylistic consistency.

The proposed hybrid framework combines event-based symbolic representation and piano-roll CNN features into a unified model for composer and style recognition. HFSNN enables joint classification, while LO optimization improves learning stability, convergence, and overall accuracy on MIDI data.

- A hybrid machine learning framework is proposed for simultaneous composer identification and musical style recognition tasks.
- An event-based symbolic MIDI preprocessing pipeline reduces performance variations using pitch, timing, duration, velocity normalization.

- Hybrid feature extraction combines handcrafted symbolic features with deep CNN-based representations from piano roll data.
- An effective hybrid feature fusion strategy integrates symbolic and deep features for enhanced musical representation learning.
- The Lion optimizer is utilized as a training optimization strategy to support efficient convergence during HFSNN training.

The remaining manuscript is organized as follows: Part 3 shows proposed methodology, Part 4 shows experimental setup, Part 5 shows results with discussions and Part 6 concludes the paper.

3. Materials and methods

Figure 1 presents the proposed framework where EB-SMR extracts symbolic and piano-roll features, fused with CNN representations for classification. Classification is performed using the Hybrid Feature-Based Softmax Neural Network (HFSNN), while the Lion optimizer is used as a training optimization method to facilitate stable and efficient parameter convergence.

The HFSNN architecture consists of fully connected layers that receive concatenated hybrid embeddings derived from handcrafted symbolic features and CNN-extracted deep features. The CNN employs two convolutional layers with 32 and 64 filters (3×3 kernel, stride 1), followed by ReLU and max-pooling. Flattened features are fused into a 128-dimensional hybrid vector and classified using HFSNN with 64- and 32-neuron hidden layers and Softmax output. Training uses the Lion optimizer with a learning rate of 0.001 and batch size of 64.

Handcrafted MIDI features extract composer-specific patterns using pitch intervals, statistical note events, and melodic transitions to identify tonal preferences, as expressed in Eq. (1):

$$\Delta p_i = p_{i+1} - p_i \tag{1}$$

Where, P_i denotes the pitch value of the current note, P_{i+1} denotes the pitch value of the subsequent note, and ΔP_i represents the pitch interval between two consecutive notes. Rhythmic features capture timing patterns and stylistic rhythmic structures in Eq. (2):

$$d_i = t_i^{off} - t_i^{on} \tag{2}$$

Where, t_i^{on} denotes the onset time of the note, t_i^{off} denotes the offset time of the note, and D_i represents the duration of the note. Velocity features capture musical dynamics by extracting statistical measures of loudness and articulation variations, which are often composer-dependent, as expressed in Eq. (3):

$$\mu_v = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{3}$$

Where, V_i represents the velocity value of the i^{th} note, N denotes the total number of notes, and $\bar{V}$ represents the average velocity across the sequence.

Deep features are automatically extracted using a CNN from piano-roll representations, where pitches form rows and temporal sequences form columns. The piano-roll is a 128×T matrix where 128 represents MIDI pitches and T denotes time steps, preserving pitch and temporal information for CNN feature extraction. The CNN extracts the local harmonic and rhythmic patterns by using convolution is expressed in Eq. (4):

$$F_k = \sigma\left(W_k * X + b_k\right) \tag{4}$$

Where, X represents the input piano-roll matrix, K denotes the convolution kernel, b is the bias term, * denotes convolution, $\sigma(\cdot)$ represents the activation function, and F is the resulting feature map.

The feature fusion strategy combines symbolic and CNN-derived features into a unified representation. This integration preserves both explicit musical information and deep learned patterns, enabling effective recognition across diverse musical styles. Handcrafted features capture musical theory, while CNN features extract latent harmonic and temporal patterns; combining both ensures a more complete representation. Handcrafted features capture explicit musical traits, while CNN features learn latent patterns; together they provide complementary information, improving classification robustness. Feature-level fusion combines both features to improve classification performance is expressed in Eq. (5):

$$F_{\text{fusion}} = [F_{\text{handcrafted}} \| F_{CNN}] \tag{5}$$

Where, F_s denotes handcrafted symbolic features, F_d denotes CNN-derived deep features, $\oplus$ represents feature concatenation, and F_h is the fused hybrid feature vector.

HFSNN is a neural classifier operating on a hybrid embedding space combining symbolic and deep musical features for composer-conditioned learning and optimization. The CNN was trained with a 0.001 learning rate using the Lion optimizer, ensuring stable convergence and robust feature extraction from piano-roll representations. Hybrid feature vector combines CNN deep features and symbolic

musical features. The hybrid representation is defined as in Eq. (6):

$$F_{hyb} = [F_{hc} \oplus F_{cnn}] \tag{6}$$

Where F_{hc} denotes the handcrafted symbolic feature vector, F_{cnn} represents the deep feature vector learned by the CNN, and $\oplus$ indicates feature concatenation. Hybrid features are passed through a fully connected layer to project them into class-specific space, improving separation of composers and styles., as expressed in Eq. (7):

$$z = WF_{hyb} + b \tag{7}$$

Where, W denotes the trainable weight matrix, b denotes the bias vector, and Z represents the projected class score vector. A Softmax activation function has been applied to the output from the fully connected layer to obtain a probabilistic prediction for each class of composer and style, as expressed in Eq. (8):

$$P(y = i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{8}$$

Where, $P(y = i)$ denotes the probability of class i, z_i represents the output logit of class i, and C denotes the total number of composer or musical style classes. It optimizes parameters of the HFSNN classifier with the use of the categorical cross-entropy loss function that punishes for making poor predictions but encourages to make more accurate class separation, as shown in Eq. (9):

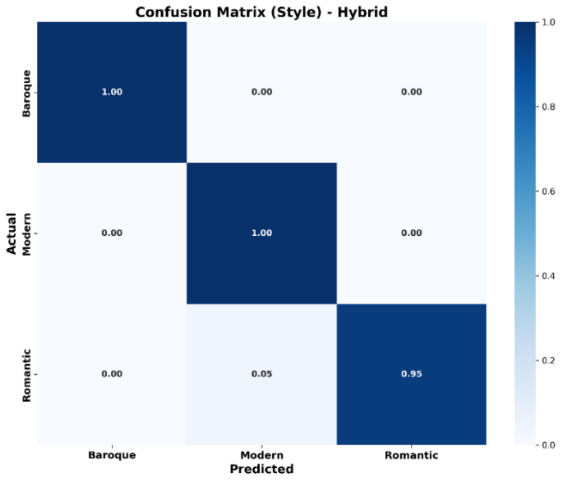

Fig. 2. Confusion Matrix for Musical Style Classification using Hybrid Features

$$L = -\sum_{i=1}^{C} y_i log\,(\widehat{y}_i) \tag{9}$$

Where, L denotes the categorical cross-entropy loss, y_i represents the ground-truth class label encoded using one-hot representation, $\widehat{y}_i$ denotes the predicted probability for class i, and C represents the total number of classes. HFSNN fuses EB-SMR, handcrafted, and CNN features, using LO-based weight updates to minimize loss, enhance accuracy, and improve convergence. The Lion optimizer utilizes gradient information obtained from the classification loss to refine the trainable network parameters. By incorporating momentum-guided optimization, it enables efficient exploration of the hybrid feature space while maintaining learning stability. The Lion optimizer was selected due to its ability to achieve stable convergence and efficient parameter updates in high-dimensional hybrid feature spaces. Its momentum-guided optimization strategy effectively supports learning from both handcrafted symbolic features and CNN-derived deep representations.

Weights are initialized randomly to improve training stability, as expressed in Eq. (10):

$$\theta_i^{(0)} = Uniform(LB, UB) \tag{10}$$

Where, LB and UB denote the lower and upper weights. Initially, a good starting point can be established through random initialization. The trainable weights are initialized using a uniform probability distribution bounded by predefined lower and upper limits. Uniform initialization promotes stable learning and prevents premature saturation.

Hyperparameter tuning was performed through an iterative experimental search strategy using the validation dataset. Multiple configurations of learning rate, batch size, and momentum coefficient were evaluated based on classification accuracy and loss convergence behavior as shown in Table 1

Table 1. Hyper parameters of LO

Parameter	Value
Learning rate (η)	0.001
Momentum coefficient (β)	0.9
Maximum epochs (T)	200
Batch size	64

Composer embeddings generate MIDI by sampling class-conditional distributions for pitch, duration, timing, and style coherence as expressed in Eq. (11):

$$P\left(e_t \mid e_1, e_2, \ldots, e_{t-1}, c\right) \tag{11}$$

Where, e_t denotes the current symbolic event, $e_1, \ldots, e_{t-1}$ represent previously generated symbolic events, and c denotes the selected composer class condition.

Table 2. Unseen Composer Generalization Performance

Test Scenario	Accuracy (%)	F1-Score (%)
Seen Composers	97.12	97.14
Unseen Composer Split	92.83	92.51

Table 3. Statistical Validation Results Using 5-Fold Cross Validation

Metric	Mean	Std Dev	95% Confidence Interval
Accuracy	0.9712	0.0081	0.9641 – 0.9783
Precision	0.9734	0.0075	0.9668 – 0.9800
Recall	0.9712	0.0084	0.9638 – 0.9786
F1-Score	0.9714	0.0078	0.9645 – 0.9783
MCC	0.9479	0.0126	0.9368 – 0.9590

The probability distribution is learned from composer-specific hybrid feature embeddings that capture characteristic pitch, rhythmic, and temporal patterns. During generation, symbolic events are sampled from this learned distribution, enabling the generated sequence to maintain stylistic consistency with the selected composer class. To regulate diversity and prevent deterministic outputs, temperature-controlled softmax sampling is applied in Eq. (12):

$$P\left(e_i\right) = \frac{exp\left(z_i/T\right)}{\sum_{j=1}^{N} exp\left(z_j/T\right)} \tag{12}$$

Where, $P\left(e_i\right)$ denotes the probability of selecting symbolic event i, z_i represents the event logit, T is the temperature parameter, and N denotes the total number of symbolic event categories. Symbolic events form MIDI files, enabling efficient composer-style modeling.

4. Result and discussion

The experimental setup defines data preparation, tools, and evaluation metrics for reproducibility and accurate performance measurement. Composer-level stratified splitting preserves proportional representation across training, validation, and test sets, ensuring consistent class distribution and reproducible evaluation across experiments.

MAESTRO v3.0.0 Dataset: MAESTRO v3.0.0 contains ~1,282 aligned MIDI–WAV recordings used for transcription and generation, with EB-SMR preprocessing symbolic events into piano-roll format. During preprocessing, invalid or incomplete MIDI events were excluded to ensure data consistency. The symbolic sequences were segmented into structured event representations suitable for learning, while note attributes such as pitch, onset time, duration, and velocity were verified before feature extraction.

Tempo normalization scales onset times using maximum sequence duration, reducing tempo variations while preserving relative rhythmic structure and consistency across performances as indicated in Eq. (13):

$$\widehat{t_i} = \frac{t_i - t_{i-1}}{t_{\max}} \tag{13}$$

Where t_i is the normalized relative onset time and t_{i-1} are consecutive note onset times, $t_{\max}$ is the maximum onset time in the sequence, and $\widehat{t_i}$ is the normalized relative onset time.

Duration values are normalized relative to maximum sequence duration, preserving rhythmic relationships while reducing outlier effects for consistent style characterization. The EB-SMR pipeline sequentially performs MIDI parsing, tempo normalization, pitch interval encoding, and duration normalization before generating piano-roll representations. Raw MIDI is parsed into symbolic notes for preprocessing and feature extraction.

The framework combines symbolic, piano-roll, and CNN features, using HFSNN with Lion optimizer, evaluated via Accuracy, Precision, Recall, F1-score, and MCC. The HFSNN model is trained using a mini-batch learning strategy with a batch size of 64. Training is performed for a maximum of 200 epochs using the Lion optimizer with a learning rate of 0.001. During training, the classification loss is continuously monitored to evaluate model convergence, and network parameters are iteratively updated until stable performance is achieved across successive epochs.

Results assess harmonic coherence, tonal stability, rhythmic consistency, and pitch similarity, indicating strong long-range dependency and structured chord capture.

Figure 2 shows a confusion matrix with strong diagonal dominance, perfect classification for Baroque and Modern, and minor Romantic misclassification into Modern.

Table 4. Performance Evaluation Under Composer Imbalance Conditions

Composer Frequency Group	Accuracy	Precision	Recall	F1
High (>25 samples)	0.98	0.98	0.98	0.98
Medium (10–25 samples)	0.96	0.97	0.96	0.96
Low (<10 samples)	0.94	0.95	0.94	0.94

Table 5. Comparative Performance Analysis with Recent Deep Learning and Transformer-Based Models

Model	Accuracy (%)	F1-Score (%)
GAT-LSTM	78.00	77.00
ECAS-CNN	95.26	95.41
Transformer-VAE	94.10	94.00
MRBERT	95.80	95.60
Proposed Hybrid Framework	**97.12**	**97.14**

Figure 3 shows feature correlations between pitch, duration, and rhythm. Symbolic and CNN features capture complementary melodic, rhythmic, harmonic, and temporal patterns, improving separability and classification performance.

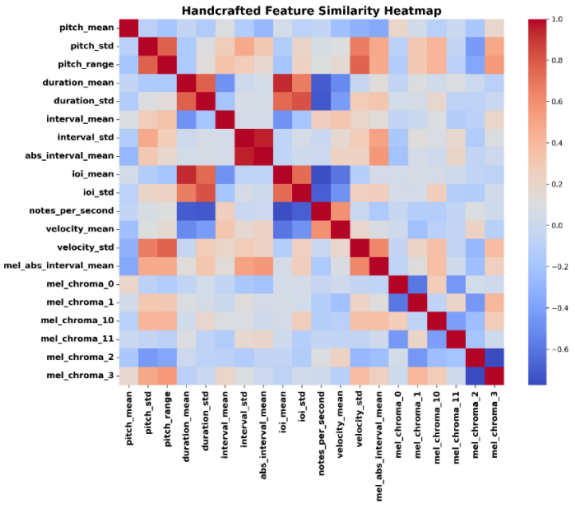

Fig. 3. Correlation Heatmap of Handcrafted Symbolic Music Features

Figure 4 t-SNE shows clear clusters of composers and styles, demonstrating hybrid features effectively capture discriminative stylistic characteristics. The observed cluster separation reflects distinctive composer-specific musical characteristics. Composers differ in their use of harmony, melody, rhythm, and dynamics, contributing to the clear separation observed in the learned feature space.

Table 2 shows strong performance on seen and unseen composers, demonstrating good generalization with high accuracy and F1-scores.

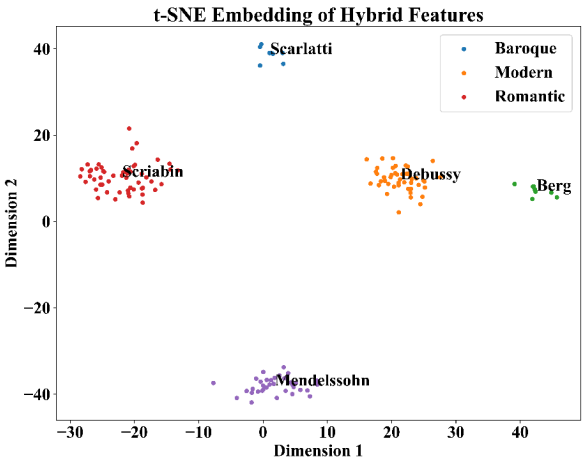

Fig. 4. t-SNE Embedding of Hybrid Features by Musical Style

Figure 5 compares generated and real piano rolls; generated ones are compact, while classical rolls are denser with wider time–pitch range, and exhibits greater musical complexity.

Table 3 shows 5-fold cross-validation results with low variance and narrow confidence intervals, indicating stable and reliable performance.

Table 4 shows high accuracy and F1-scores across composer groups, demonstrating robustness against imbalance and varying dataset distributions.

Table 5 shows the proposed hybrid framework outperforms recent deep learning and transformer models, achieving highest accuracy and F1-score.

Table 6 shows consistent performance across genres, demonstrating the model's applicability beyond classical music.

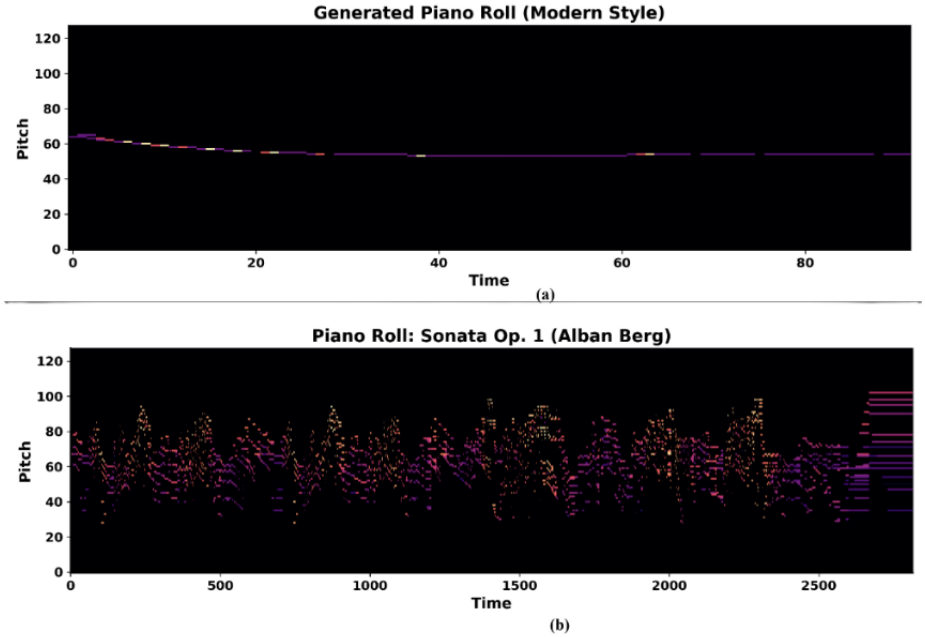

Fig. 5. Piano Roll Comparison of Generated Modern Music (a) Generated Piano Roll (b) Original Piano Roll

Table 6. Genre-Specific Performance Analysis of the Proposed Hybrid Framework

Genre	Accuracy (%)	F1-Score (%)
Classical	97.12	97.14
Jazz	95.84	95.62
Pop	95.21	95.03
Rock	94.76	94.51

Table 7 shows the hybrid model outperforms individual features, while capturing style but not full expressive richness. Qualitative evaluation shows generated music preserves style, rhythm, pitch, and composer traits. Hybrid symbolic and CNN features improve composer and style discrimination. Symbolic features capture explicit musical attributes such as pitch and rhythm, while CNN features learn deeper harmonic and temporal patterns. Their combination provides complementary information that improves composer and style recognition.

5. Conclusion

The paper proposes a hybrid model for musical style recognition and composer-aware generation using symbolic MIDI representations and EB-SMR features, handcrafted symbolic features, CNN-based deep features, and an HF-SNN classifier. The Lion optimizer was utilized as a training mechanism to facilitate efficient convergence during model optimization. Using symbolic and CNN-based features, the framework achieves 0.97 accuracy on MAESTRO but remains limited to piano-based symbolic data. The proposed framework demonstrates effective performance on the MAESTRO symbolic MIDI dataset. Further validation using diverse datasets and human-centred evaluation is needed to confirm its robustness and generalizability. The proposed framework was evaluated only on the MAESTRO classical piano dataset and has not been validated on diverse musical genres. Future work includes multimodal audio-symbolic learning, Transformer-based modeling, advanced data balancing methods, and improved generative techniques to enhance realism, adaptability, and performance in musical style recognition and composer-aware generation.

Acknowledgement

Declarations

Data availability

The data is available upon request.

Table 7. Ablation Study of the Proposed Model

Model Variant	Accuracy	Precision	Recall	F1-score	MCC
Proposed (Hybrid)	**0.9712**	**0.9734**	**0.9712**	**0.9714**	**0.9479**
W/O Handcrafted (CNN Only)	0.9361	0.9388	0.9361	0.9365	0.8830
W/O CNN (Handcrafted Only)	0.6316	0.8051	0.6316	0.6216	0.4934

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding statement

This research received no external funding.

Author contributions

Xin Xiong: Conceptualization, methodology, implementation of the proposed framework, experimental design, data preprocessing, model development, and manuscript writing.

Ethical approval

This study does not involve human participants, animals, or clinical data. Therefore, ethical approval is not required.

Consent to participate

Not applicable, as the study does not involve human participants.

Consent to publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

[1] I. Abarkan, M. Rabi, F. P. V. Ferreira, R. Shamass, V. Limbachiya, Y. S. Jweihan, and L. F. Pinho Santos, (2024) *"Machine learning for optimal design of circular hollow section stainless steel stub columns: A comparative analysis with Eurocode 3 predictions"* **Engineering Applications of Artificial Intelligence *132***: 107952. DOI: 10.1016/j.engappai.2024.107952.

[2] C.-Y. Chang and Y.-P. Chen, (2020) *"AntsOMG: A Framework Aiming to Automate Creativity and Intelligent Behavior with a Showcase on Cantus Firmus Composition and Style Development"* **Electronics *9***(8): 1212. DOI: 10.3390/electronics9081212.

[3] P. Ferreira, R. Limongi, and L. P. Fávero, (2023) *"Generating Music with Data: Application of Deep Learning Models for Symbolic Music Composition"* **Applied Sciences *13***(7): 4543. DOI: 10.3390/app13074543.

[4] J.-Y. Guo and P. Wang, (2025) *"Music Emotion Classification Based on Heterogeneous Graph Neural Networks"* **IEEE Access *13***: 76473–76480. DOI: 10.1109/ACCESS.2025.3562532.

[5] Q. He, (2022) *"A Music Genre Classification Method Based on Deep Learning"* **Mathematical Problems in Engineering *2022***(1): 9668018. DOI: 10.1155/2022/9668018.

[6] Y. S. Jweihan, M. J. Al-Kheetan, and M. Rabi, (2023) *"Empirical Model for the Retained Stability Index of Asphalt Mixtures Using Hybrid Machine Learning Approach"* **Applied System Innovation *6***(5): 93. DOI: 10.3390/asi6050093.

[7] M. Miller, J. Rauscher, D. A. Keim, and M. El-Assady, (2022) *"CorpusVis: Visual Analysis of Digital Sheet Music Collections"* **Computer Graphics Forum *41***(3): 283–294. DOI: 10.1111/cgf.14540.

[8] M. Rabi, (2024) *"Bond prediction of stainless-steel reinforcement using artificial neural networks"* **Proceedings of the Institution of Civil Engineers - Construction Materials *177***(2): 87–97. DOI: 10.1680/jcoma.22.00098.

[9] M. Rabi, Y. S. Jweihan, I. Abarkan, F. P. V. Ferreira, R. Shamass, V. Limbachiya, K. D. Tsavdaridis, and L. F. Pinho Santos, (2024) *"Machine learning-driven web-post buckling resistance prediction for high-strength steel beams with elliptically-based web openings"* **Results in Engineering *21***: 101749. DOI: 10.1016/j.rineng.2024.101749.

[10] S. Sarfarazi, R. Shamass, M. Rabi, I. Abarkan, F. P. V. Ferreira, and K. D. Tsavdaridis, (2026) *"Inverse machine learning for the design of perforated beams: Parent section and material prediction"* **Engineering Applications of Artificial Intelligence *164***: Part A: 113275. DOI: 10.1016/j.engappai.2025.113275.

[11] Y. Shi, (2025) *"A CNN-Based Approach for Classical Music Recognition and Style Emotion Classification"* **IEEE Access *13***: 20647–20666. DOI: 10.1109/ACCESS.2025.3535411.

[12] L. Turchet and J. Pauwels, (2022) *"Music Emotion Recognition: Intention of Composers-Performers Versus Perception of Musicians, Non-Musicians, and Listening Machines"* **IEEE/ACM Transactions on Audio, Speech, and Language Processing *30***: 305–316. DOI: 10.1109/TASLP.2021.3138709.

[13] S.-L. Wu and Y.-H. Yang, (2023) *"MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer With One Transformer VAE"* **IEEE/ACM Transactions on Audio, Speech, and Language Processing *31***: 1953–1967. DOI: 10.1109/TASLP.2023.3270726.

[14] K. Zhang, (2021) *"Music Style Classification Algorithm Based on Music Feature Extraction and Deep Neural Network"* **Wireless Communications and Mobile Computing *2021***(1): 9298654. DOI: 10.1155/2021/9298654.

[15] S. Zhang, Y. Liu, and M. Zhou, (2025) *"Graph Neural Network and LSTM Integration for Enhanced Multi-Label Style Classification of Piano Sonatas"* **Sensors *25***(3): 666. DOI: 10.3390/s25030666.