ISSN 2278-3091
Volume 15, No.2, March – April 2026
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse031522026.pdf
https://doi.org/10.30534/ijatcse/2026/031522026

Regularized Deep Learning Models for Acoustic Event Classification

Fe B Yara[1,3], Dr. Bobby D. Gerardo[2]
[1]Technological Institute of the Philippines, Philippines, qfyara@tip.edu.ph
[2]Northern Iloilo State University, Philippines, bgerardo@nisu.edu.ph
[3]University of Mindanao, Philippines, fe_yara@umindanao.edu.ph

Received Date: February 14, 2026 Accepted Date: March 20, 2026 Published Date: April 06, 2026

ABSTRACT

Sound classification plays a vital role in various applications, including speech recognition, environmental monitoring, health, and multimedia analysis by enabling machines to interpret and categorize audio signals effectively. Despite its importance, many existing studies lack sufficient regularization techniques, leading to overfitting and limited generalization to unseen data. To address this gap, this study follows a structured approach that involves audio preprocessing, feature extraction, and splitting the data into training and test sets. The models are then trained and evaluated using four deep learning architectures: CNN, RNN, LSTM, and BiLSTM. To mitigate overfitting, several regularization techniques are applied, including dropout, batch normalization, early stopping, and k-fold cross-validation. The findings indicate that incorporating regularization improves generalization performance across all models, with accuracy achieved ranging from 61% to 87%, which is lower than the 91% to 99% reported in related studies, indicating a trade-off between improved generalization and peak accuracy.

Key words : Acoustic Event, Deep Learning, Regularized, Sound classification, Model performance comparison

1. INTRODUCTION

The classification of acoustic signals using machine learning has become essential across various application domains such as environmental sound analysis, speech recognition, and music classification due to its ability to learn complex patterns [1]. Acoustic sound like music is widely acknowledged for its positive impact on human health and performance. Existing studies have been shown to support mental well-being and enhance work efficiency among employees in various sectors [2], while also contributing to improved physical and emotional health [3], enhance concentration and reduce physiological markers of stress particularly the cortisol level when music genre preferences like classical, ambient music for calmness and lo-fi attentional support [4] and overall human well-being [5].

As digital audio continues to grow, machine learning-based audio classification has become an essential tool for improving accuracy, efficiency, and scalability in sound-related applications, thereby increasing the need for retrieving audio data based on attributes such as genre or mood from massive databases [6]. Despite progress with various machine learning classifiers, there remains a need to enhance the traditional machine learning classifiers [7], [8], [9].

Music has been a key medium of expression, and music genre classification is vital to its understanding and creative development [8]. It has become a core component of music information retrieval, serving as an effective approach for handling large volumes of music data [10]. Recent studies noted to be susceptible to overfitting on small datasets like GTZAN [4], without significant augmentation [11], lack of training data for non-Western music, found to have significant genre confusion due to overlapping melodic structures [12], inadequate regularization leading to overfitting [13], [14], and lack comparative insights [15].

This study aims to compare results across several studies that have cross-validated different machine learning models after feature extraction from a preprocessed audio dataset, GTZAN, for music genre classification, implement regularization techniques to mitigate overfitting due to insufficient regularization, and determine which models require further improvement or enhancement to address current limitations.

2. METHODOLOGY

This study presents four stages in the development of models for acoustic event classification. Figure 1 shows the data preprocessing, feature extraction, data splitting, deep learning model training and testing, regularization, and training optimization.

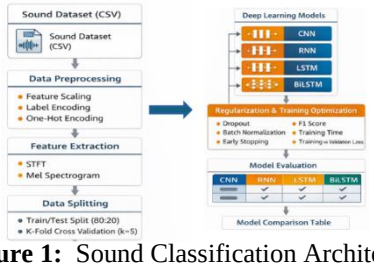


Figure 1: Sound Classification Architecture

2.1 Data Collection

The dataset employed in this study is from publicly available music database of Kaggle, the GTZAN genre collection. The dataset originally consisted of 1,000 audio files spanning ten musical genres such as blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock with 100 tracks per genre. Each audio file in its original forms runs 30 seconds at a sampling rate of 22,050 Hz in mono wav format, which was standardized to a fixed duration of 3 seconds and trimmed uniformly yielding approximately 10 non-overlapping segments. Table 1 shows the total 9,990 audio samples, effectively expanding the dataset tenfold while ensuring input uniformity across all deep learning models.

Table 1. Number of Datasets

Music genres	Number of Samples
Rock	998
Reggae	1000
Pop	1000
Metal	1000
Jazz	1000
Hip-hop	998
Blues	1000
Classical	998
Disco	999
Country	997
Total	9990

2.1 Exploratory Data Analysis

Figure 3 visualizes the raw audio signal waveform over time, to inspect the raw audio data, detect noise and understand signal patterns before feature extraction. Figure 2 shows a dense and continuous waveform, indicating that the audio contains consistent sound activity.

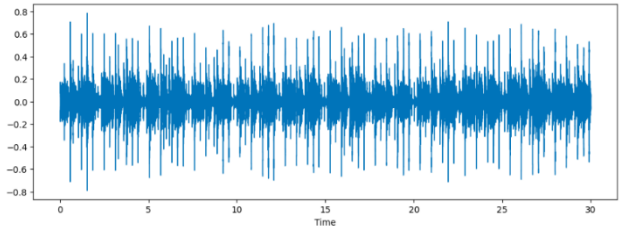


Figure 2: Waveform

2.2 Ethical Considerations

This study adheres to copyright and data privacy requirements: all audio samples are obtained from an open-source database platform, used strictly for academic research, and no personal identifiable information was collected or processed. Also, equal representation of samples across all music genres was used to prevent algorithm bias and ensure equitable model performance, showing the documented transparency of the preprocessing step, model architecture, and evaluation metrics.

2.3 Data Preparation

Following data collection, a structured preprocessing pipeline was applied to prepare the raw audio signals for model training. Figure 3 shows the pipeline, which consisted of three core states: the feature scaling, label encoding, and one-hot encoding prior to feature extraction using Short-Time Fourier Transform (STFT) and Mel Spectogram representations. The use of MFCCs and STFT accurate classification relies on extracting distinct features from audio signals that effectively represent the underlying linguistic content while filtering out noise and irrelevant components [17].

Figure 3: Preprocessing pipeline

A. Feature Scaling, Label Encoding, and One-hot Encoding

The raw time-series signals extracted from each audio segment were normalized using feature scaling to bring all input values within a consistent numerical range. Standardization ensures that no single feature dominates the learning process due to scale differences, which is particularly important for gradient-based optimization in deep learning models [18].

B. Feature Extraction

Two feature extraction methods were applied to the preprocessed time-series signals. The STFT was used to decompose the audio signal into its frequency components over time, producing a two-dimensional time frequency representation. This process utilizes `librosa.stft(y)` to load the sound and convert audio signal from the time domain to the time frequency components. Figure 4 shows the Mel Spectrogram, which further transforms this representation by mapping frequencies to the Mel scale, which approximates the non-linear frequency perception of the human auditory system in logarithmic decibels (`dB`).

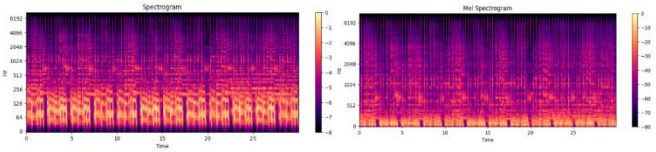


Figure 4: Feature Extraction

2.4 Data Splitting

The dataset was divided into training and test sets following an 80:20 ratio, allocating 80% of the 9,990 samples for training and the remaining 20% for held-out testing. To ensure generalized performance estimates, 5-Fold cross-validation (K=5) was additionally employed during model training. This approach partitions the training data into five equal folds, rotating the validation fold across all five iterations and averaging the metrics. K-fold cross-validation mitigates the risk of performance inflation due to a favorable random split, producing a more reliable estimate of each model's generalization capability compared to a single train-test split. The 70:30 ratio was also explored, 70% training, 15% validation, and 15% test in several epochs, 30, 50, and 100.

2.5 Model Construction

Four distinct deep learning architectures were constructed and evaluated in this study, the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM). These algorithms were selected due to their popularity in classifying music genres. Each model was designed to accept the preprocessed spectral features as input and output a probability distribution over the ten genre classes via a softmax activation function. The architecture diagram in Figure 1 summarizes the overall system design. The overall process was conducted using Google Colab.

CNN

The CNN model was constructed using two convolutional blocks, each consisting of two `Conv1D` layers with `ReLU` activation and same padding, followed by Batch Normalization, MaxPooling1D, and Dropout. Figure 5 shows the use of double convolutional blocks per stage follows the VGG-style design principle, which has been shown to extract richer local features before the dense output layers to reduce overfitting by spatially summarizing learned feature maps rather than exposing all individual feature weights to the classifier.

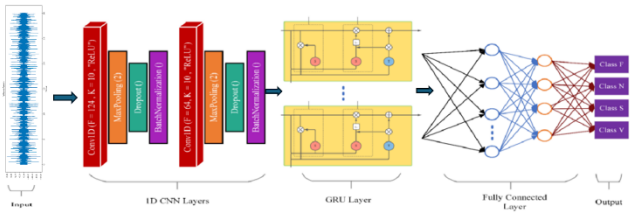
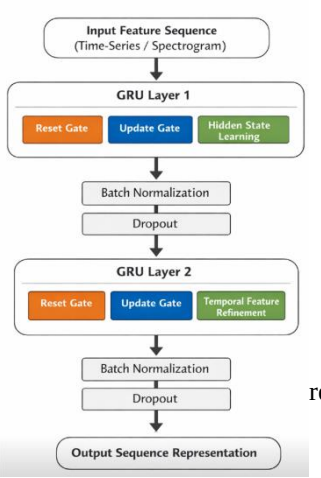


Figure 5: CNN Architecture

RNN

The RNN model replaced the conventional SimpleRNN units with Gated Recurrent Units (GRU), which incorporate reset and update gates to selectively retain or discard information across time steps. This model is specifically built to handle sequential data by preserving a hidden state that retains information from earlier inputs [19]. This architectural decision addresses the well-documented vanishing gradient problem associated with vanilla RNNs on longer sequences [20]. Two stacked GRU layers were used, with Batch Normalization and Dropout applied after each layer to stabilize training and reduce overfitting.

Figure 6 visualizes the architecture of RNN showing the two (2) stacked GRU layers, after each GRU layer, Batch Normalization was applied to stabilize the learning process and accelerate convergence, while Dropout regularization was incorporated to mitigate overfitting by randomly deactivating a fraction of neurons during training.



re

LSTM

The LSTM model is developed to overcome the vanishing gradient issue found in traditional RNN by utilizing specialized memory units called cells, which consist of three key gates such as input, forget and output gates [21]. This model was structured with two (2) stacked LSTM layers, each incorporating L2 kernel regularization, recurrent regularization, and recurrent dropout. Batch Normalization was inserted after each LSTM layer. The addition of recurrent dropout, which applies dropout to the recurrent connections within the LSTM cell, is considered among the most effective techniques for regularizing recurrent network [22]. Figure 7 shows the LSTM model architecture, indicating the two stacked LSTM layers.

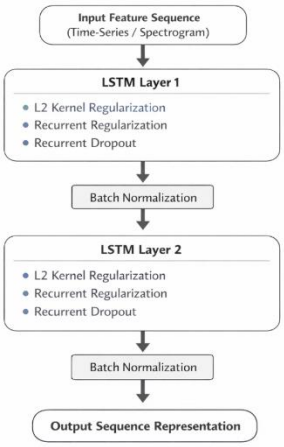


Figure 7: LSTM Architecture

BiLSTM

This model mirrors the LSTM architecture but wraps each LSTM layer with Bidirectional wrapper, enabling the model to learn both forward and backward temporal dependencies simultaneously in varying lengths [23]. This is particularly valuable in audio classification tasks where genre-relevant patterns such as rhythmic structure, chord transitions, and timbal texture can manifest across multiple temporal directions within a segment [24]. Figure 8 presents the BiLSTM architecture, with the same regularization techniques applied to the LSTM, retained in the BiLSTM.

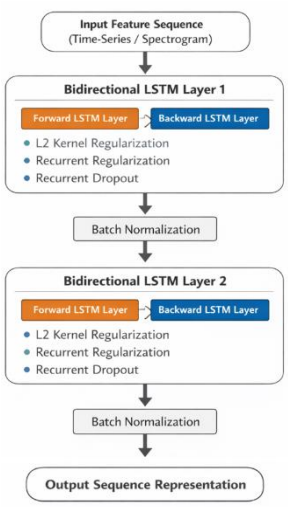


Figure 8: BiLSTM Architecture

2.6 Regularization and Training Optimization

To address the generalization limitations commonly reported in music genre classification studies, particularly overfitting on the GTZAN dataset, this study systematically applied three training optimization strategies, such as Dropout, Batch Normalization, and Early Stopping across all four models.

Dropout

Dropout was applied at multiple points within each model architecture, with rates ranging from 0.2 to 0.3. In each of the four (4) models, recurrent dropout of 0.1 was additionally applied directly to the recurrent connections within each cell. Dropout prevents co-adaptation of neurons by randomly deactivating a proportion of units during each training step, forcing the network to learn more distributed and generalizable representations [25]. Similarly, dropout was applied in the CNN-based sound classification system published in Sensors (MDPI), without batch normalization or early stopping, limits the effectiveness of regularization in the pipeline [26]. Further, a probabilistic dropout regularization method can enhance robustness and generalization [27].

Batch Normalization

Normalizing data is an important step when preparing data for machine learning models [28]. This regularization was applied after each major layer in all four (4) architectures. By normalizing the inputs to each layer across a mini-batch, Batch Normalization reduces internal covariate shift, accelerates convergence, and allows higher learning rates without divergence [29]. This is particularly important for LSTM and BiLSTM models, where the outputs of recurrent layers can exhibit high variance across time steps. Despite widespread use, the systematic impact of batch normalization on lightweight deep learning model accuracy remains underexplored, which serves as a gap of their study [30].

Early Stopping and Learning Rate Scheduling

Early stopping was configured with a patience of 15 epochs, monitoring validation loss and restoring the best model weight upon termination. This prevents unnecessary overfitting during prolonged training without requiring a fixed epoch count. `ReducLROnPlateau` was additionally employed, having the learning rate when validation loss plateaued for 5 consecutive epochs, with a minimum learning rate floor of 1e-6. The combination of early stopping and adaptive learning rate scheduling has been shown to improve final model accuracy by ensuring convergence to a better local minimum rather than a plateau [31], and incorporating stacked dropout layers, early stopping, and model checkpointing prevents overfitting [12].

2.7 Model Testing

Each model was evaluated on the held-out 20% test set following training with K-fold cross-validation. Performance was measured across five metrics, such as Accuracy, Precision, Recall, F1 Score, and Training Time. Table 2 presents the consolidated results for all four models.

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score	Training Time (s)
CNN	0.8754	0.8748	0.8754	0.8748	186.27
RNN	0.7903	0.7914	0.7903	0.7895	281.34
LSTM	0.6111	0.6095	0.6111	0.6058	516.15
BiLSTM	0.7192	0.7197	0.7192	0.7181	894.90

Accuracy

In this study, accuracy measures the overall correctness of the model across all music genres and is defined as the ratio of correctly classified instances to the total number of classified objects [32], which represents the key indicator of a machine learning model's performance. The accuracy is obtained using the equation:

$$Accuracy = \frac{(TP + TN)}{(TP+FP+TN+FN)}$$

Where TP represents True Positives, FP denotes False Positives, TN stands for True Negatives, and FN indicates False Negatives. A True Positive (TP) occurs when the model correctly predicts the genre, a True Negative (TN) occurs when the model correctly identifies samples not belonging to a given genre, while the False Positives (FP) and False Negative (FN) represent incorrect predictions In this study, accuracy_score() compares the predicted genre labels (y_pred) with the true labels (y_true).

Precision

This metric evaluates how many of the samples predicted as a specific genre are correct. Scikit-learn computes precision per genre using the TP and FP values from the confusion matrix.

$$Precision = \frac{TP}{(TP + FP)}$$

Recall

This metric evaluates how many of the samples predicted as a specific genre

$$Recall = \frac{TP}{(TP + FN)}$$

F1 Score

F1 score combines precision and recall into a single metric, providing a balanced measure of model performance. In this study, precision and recall are first computed per genre. The mean value of precision and recall is then calculated. Weighted averaging ensures that genres with more samples contribute proportionally to the final scores. Overall, this metric is particularly useful when there is a trade-off between precision and recall.

Training Time

Training Time records the total wall-clock duration in seconds required to complete model training.

3. RESULTS AND DISCUSSION

The results presented in Table 1 reveal meaningful differences in performance across four (4) architectures. The following subsections discuss each model's results in detail, contextualized against findings from related scholarly studies.

CNN

The CNN model achieved the highest accuracy among all four (4) architectures at 87.54%, with precision, recall, and F1 score values all closely aligned approximately 0.875. This consistency across metrics indicated that the model classifies genres with relatively balanced performance across all ten classes rather than excelling on a subset while failing on others. The training time of 186.27 seconds was also the shortest among all models, demonstrating CNN's computational efficiency on time-series audio data.

This result is consistent with broader literature. Wenyi Xu (2024) reported 91.9% accuracy for CNN model using short-time Fourier spectrum inputs on the GTZAN dataset, while Mumtahina Ahmed et al., (2024) achieved 92.7% with a modified CNN evaluated on GTZAN and ISMIR2004. The Approximately 4-5% gap between those figures and the present study result of 87.54% is attributable primarily to methodological differences. The present study employed K-Fold cross-validation and strict train-test separation, whereas the comparative studies simpler hold-out splits that are more susceptible to data leakage and favorable partitioning. Under a controlled evaluation protocol, 87.54% is a credible result.

The strong performance of CNN on 3-second clips is explained by its architectural affinity for local feature extraction. Convolutional filters learn localized spectral patterns such as timbral textures, harmonic content, and rhythmic density, which are present and discriminative even within short audio windows. The use of GlobalAveragePooling1D in place of Flatten further contributed to generalization by reducing the parameter count in the classifier head, a technique supported by Lin et al., (2014) in their work on network-in-network architecture.

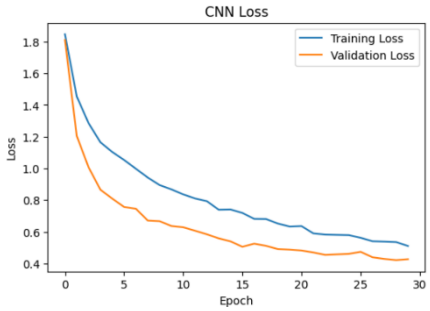


Figure 9: CNN Training Loss vs Validation Loss

Figure 9 shows a line graph of CNN loss over training epochs, comparing training loss and validation loss. Both curves decrease steadily as epochs increase, indicating effective parameter optimization and learning progression. The validation loss remains slightly lower than the training loss throughout training, showing a good generalization performance with minimal overfitting. The smooth convergence pattern further implies stable gradient updates and efficient feature extraction.

RNN

The RNN model, implemented using Gated Recurrent Units (GRU), achieved 79.03% accuracy with closely aligned precision (0.7914), recall (0.7903), and F1 score (.07895). The training time of 281.34 seconds was moderate, reflecting the sequential nature of recurrent computation compared to the parallelizable convolutions of CNN.

RNN's lower accuracy relative to CNN aligns with the comparative findings of Xu (2024), who also reported that RNN underperformed CNN in music genre classification tasks. The present result of 79.03% is defensible given that 3-second clips provide sequential context. GRU layers were specifically chosen over SimpleRNN to mitigate the vanishing gradient problem, and the addition of Batch Normalization after each GRU layer helped stabilize training. Despite these improvements, recurrent models are inherently constrained by short sequence lengths, as the gating mechanisms of GRU derive their benefit from learning to

retain information across many time steps, a condition that 3-second clips only partially satisfy.

The obtained result is aligned with the findings of Kidanue, (2025), wherein the RNN model demonstrated comparable performance metrics, achieving an accuracy of 87.33, an F1 score of 87.27, a precision of 88.12% and a recall result of 87.33%, indicating consistent model efficiency across similar machine learning-driven frameworks.

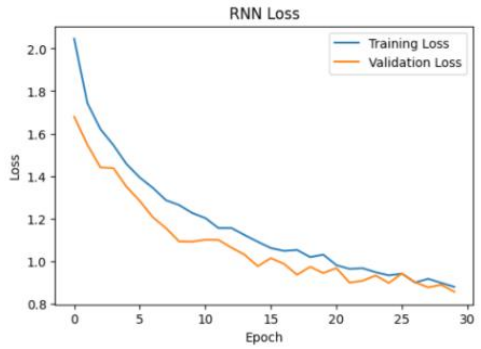


Figure 10: RNN Training Loss vs Validation Loss

Figure 10 demonstrates the RNN loss curve over training epochs, showing both training and validation loss decreasing over successive epochs. The two (2) curves follow a similar downward trend, indicating stable learning and convergence of the model. The close alignment between training and validation loss suggests balanced bias-variance tradeoff with minimal overfitting but still of good generalization performance.

LSTM

The LSTM model recorded the lowest accuracy at 61.11%, with precision at 0.6095, recall at 0.6111, and F1 score at 0.6058. The training time of 516.15 seconds was substantially longer than both CNN and RNN, reflecting the greater computational cost of LSTM's gating mechanism across all time steps in the sequence.

LSTM's underperformance relative to its theoretical capability is a finding that warrants careful interpretation. Despite applying recurrent dropout, Batch Normalization, L2 regularization, and Early Stopping, the LSTM cells are designed to selectively preserve long-range dependencies across many time steps. When sequences are short, these gating mechanisms have limited temporal history to process, reducing the LSTM to approximating a simpler feed-forward network while incurring far greater computational overhead.

The finding is further supported by a comprehensive survey of deep learning for audio classification in IEEE Access, which observed that LSTM models frequently underperform on shorter audio segments and smaller datasets [16]. The interaction between dataset size, segment duration, and LSTM performance represents a recognized challenge in the field that the present study empirically confirms under a rigorous evaluation protocol. Future work should investigate LSTM performance on longer audio segments or hierarchical architectures that aggregate information across multiple short windows.

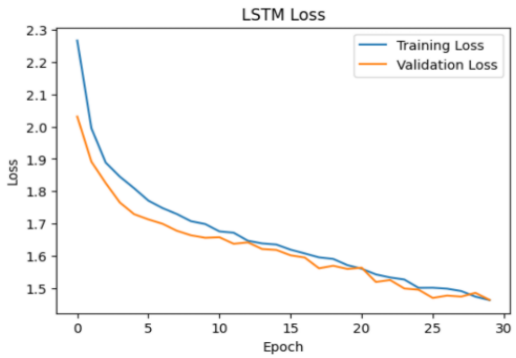


Figure 11: LSTM Training Loss vs Validation Loss

Figure 11 illustrates the training and validation loss curves of an LSTM model across epochs. Both losses exhibit a steady downward trend, indicating effective optimization and gradual convergence of the model during training. The proximity and parallel behavior of the training and validation curves suggest stable learning dynamics with minimal overfitting and good generalization capability.

BiLSTM

The BiLSTM model achieved 71.92% accuracy, outperforming LSTM by approximately 10.8 percentage points, with precision (0.7197), recall (0.7192), and F1 score (0.7181) all tightly clustered. However, the BiLSTM required the longest training time of all models at 894.90 seconds, reflecting the doubled computational burden of processing sequences in both forward and backward directions.

The BiLSTM's improvement over LSTM is theoretically expected and empirically confirmed based on the results. By processing the sequence in both temporal directions, BiLSTM gains access to future context when making predictions at each time step, providing a richer feature representation than a unidirectional LSTM.

Recent studies have explored the use of bidirectional models. Graves and Schmidhuber (2005) established that bidirectional architectures consistently outperform their unidirectional counterparts on sequence classification tasks. The present results are consistent with these literatures, Ghosh, et al., (2023), in their study on music genre classification published in the International Journal of Emerging Business and Social Sciences, reported that 88.64% accuracy using a BiLSTM model with basic dropout regularization on the full 30-second GTZAN clips, and 80% accuracy was achieved with overfitting issues [23]. The longer clip duration used in that study likely accounts for the performance advantage of recurrent models in their results compared to the present study.

Nevertheless, the BiLSTM still falls 15.62 percentage points below CNN. This gap reinforces the conclusion that for short audio segments on a dataset of the GTZAN's scale, convolutional feature extraction is more effective than sequential modeling. The significant training time of 894.90 seconds is nearly five times that of the CNN, which also raises practical considerations about computational efficiency when deploying recurrent models in resource-constrained environments.

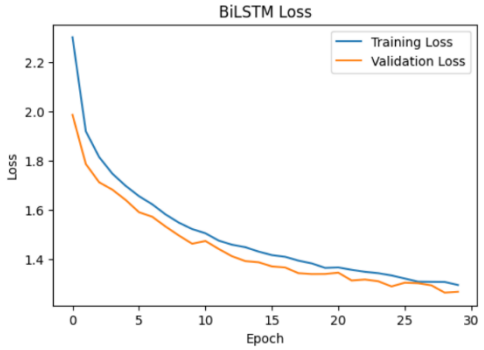


Figure 12: BiLSTM Training Loss vs Validation Loss

Figure 12 depicts the training and validation loss trajectories of a BiLSTM model over successive epochs. Both loss curves demonstrate a consistent downward trend, reflecting effective gradient-based optimization and progressive model convergence.

Notably, the validation loss closely tracks, and at times remains slightly lower than the training loss, indicating strong generalization capability and the absence of significant overfitting. The smooth and gradually flattening curves further suggest that the model is approaching convergence, with diminishing performance gains in later epochs.

4. CONCLUSION AND RECOMMENDATION

Across all four (4) models, the results consistently demonstrate that CNN is the most suitable architecture for music genre classification on the GTZAN dataset under the conditions of this study, the 3-second audio segments, K-Fold cross-validation, and a comprehensive regularization pipeline. The performance ranking CNN>RNN>BiLSTM>LSTM reflects a clear pattern: architecture that extract local spectral features (CNN) outperforms those that require long sequential context (LSTM, BiLSTM) when input sequences are short.

A critical observation is that the present study's results are systematically lower than several accuracy values reported in the literature. This is not indicative of methodological failure but rather of methodological rigor. Prior studies reporting 91-99% accuracy frequently relied on simpler train-test splits without cross-validation, did not apply comprehensive regularization, and in some cases used the full 30-second clips on a dataset known to contain artist repetition bias. The use of K-Fold cross-validation in the present study provides a more conservative and honest estimate of generalization performance.

The systematic application of dropout, batch normalization, and early stopping across all models, a combination identified as underexplored in the literature by Md Dzahir and Chia (2026) did not eliminate the performance gaps between architecture but did produce stable, reproducible results with minimal overfitting. The training versus validation loss curves confirmed that early stopping successfully prevented prolonged overfitting in all four models.

These findings collectively contribute to the understanding of deep learning model behavior on standardized audio datasets and provide an empirically grounded benchmark for future studies employing similar preprocessing pipelines. The results suggest several directions for future work like evaluating models on longer audio segments to better leverage recurrent architectures, exploring attention mechanisms or transformer-based models or hybrid models, applying dataset augmentation strategies to address GTZAN's known biases, and testing on more diverse, culturally representative audio collections.

REFERENCES

[1] F. Klarer, J. Werner, M. Klaiber, F. Gerschner, and M. Rossle, "Monitoring Applications with Sound Data: A Systematic Literature Review on Sound Classification with Transfer Learning," ScienceDirect, vol. 246, pp. 2032–2041, Nov. 2024, doi: https://doi.org/10.1016/j.procs.2024.09.661.

[2] M. Nan, "The Role of Music Therapy in the Emotional Regulation and Psychological Stress Relief of Employees in the Workplace," Journal of Healthcare Engineering, Jan. 2022, doi: 10.1155/2022/4260904.

[3] M. Dr Mahajan and A. Dhalia, "Harmony and Healing: A Multidisciplinary Examination of Music's Role in Stress Management and Holistic Well-being," National Journal of Commerce and Management, vol. 12, no. 01, Jun. 2025, doi: https://njcm.pratibha-spandan.org/wp-content/uploads/v12i01a04.pdf.

[4] R. Akhtar and A. Mishra, "Music Genre Classification Using Machine Learning Techniques," Artificial Intelligence and Data Science, Sep. 2025, doi: https://doi.org/10.48550/arXiv.2509.01762.

[5] R. de Filippis and A. A. Foysal, "Associations between Music Listening Habitsand Mental Health: A Cross-Sectional Analysis," Open Access Library Journal, vol. 12, Apr. 2025, doi: https://doi.org/10.4236/oalib.1113196.

[6] M. Ahmed, U. Rozario, M. M. Kabir, Z. Aung, J. Shin, and A. M. F. Mridha, "Musical Genre Classification Using Advanced Audio Analysis and Deep Learning Techniques," IEEE Journal of the Computer Society, vol. 5, Jul. 2024, [Online]. Available:

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10605044

[7] R. Kidanue, "Machine Learning-Driven Music Genre Recognition," Cornerstone: A Collection of Scholarly and Creative Works for Minnesota State University, Mankato, May 2025.

[8] R. Oladejo, A. Abayomi-Alli, O. Arogundade, A. Adeyanju, and M. Lawrence, "Artificial Intelligence in Music Genre Classification: A Systematic Review of Techniques, Applications, and Emerging Trends," Cureus Journal of Computer Science, Jul. 2025, doi: https://doi.org/10.7759/s44389-025-04209-9.

[9] W. Xu, "Music genre classification using deep learning: a comparative analysis of CNNs and RNNs," Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1, Nov. 2024, doi: https://doi.org/10.2478/amns-2024-3309.

[10] Y. Singh and A. Biswas, "Robustness of musical features on deep learning models for music genre classification," Expert Systems with Applications, vol. 199, Aug. 2022, doi: https://doi.org/10.1016/j.eswa.2022.116879.

[11] Thanh Chu Ba, Thuy Dao Thi Le, and Loan Trinh Van, "Music genre classification using deep neural networks and data augmentation," Entertainment Computing, Feb. 2025, doi: DOI: 10.1016/j.entcom.2025.100929.

[12] Eshete Derib Emiru and Estifanos Tadele Bogale, "Ethiopian music genre classification using deep learning," Applied Computing and Intelligence, vol. 5, no. 1, Apr. 2025, [Online]. Available: https://www.aimspress.com/article/doi/10.3934/aci.2025007

[13] Mengjin Xue, "Deep learning model using squeezenet and promoted ideal gas molecular motion for music genre classification from audio spectrograms," Scientific Reports, Sep. 2025, doi: https://doi.org/10.1038/s41598-025-16499-z.

[14] Shuang Zhang, Zhiyong Sun, and Hasan Jafari, "A novel approach for music genre identification using ZFNet, ELM, and modified electric eel foraging optimizer," Scientific Reports, Apr. 2025.

[15] Arda Deniz, Bilal Saoud, Ibraheem Shayea, Shambulov Ulykbek, Abilkair Imanberdi, and Fuad Abdulgaleel Abdoh Ghaleb, "Audio Data Analysis and Music Genre Classification with Various Machine Learning Techniques," IEEE/ACIS 29th International Conference on Software Engineering, pp. 219–224, 2025, doi: 10.1109/SNPD65828.2025.11254808.

[16] Khalid Zaman, Melike Sah, Cem Direkoglu, and Masashi Unoki, "A Survey of Audio Classification Using Deep Learning," Open Access Journal, vol. 11, pp. 106620–106649, Sep. 2023, doi: 10.1109/ACCESS.2023.3318015.

[17] Mahendra Kumar Gourisaria, Rakshit Agrawal, Manoj Sahni, and Pradeep Kumar Singh, "Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques," Discover Internet of Things, Jan. 2024.

[18] Yann Lecun, Leon Bottou, Genevieve Orr, and Klaus-Rober Muller, "Efficient BackProp", [Online]. Available: https://www.researchgate.net/publication/2811922_Efficient_BackProp

[19] Ibomoiye Domor Mienye, Theo G. Swart, and George Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," Open Access Journal, vol. 15, no. 9, p. 517, Aug. 2024, doi: https://doi.org/10.3390/info15090517.

[20] Kyunghyun Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," vol. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, Oct. 2014, doi: 10.3115/v1/D14-1179.

[21] Hamed Alizadegan, Behzad Rashidi Malki, Mohsen Asghari Ilani, Arian Radmehr, and Hossein Karimi, "Comparative study of long short-term memory (LSTM), bidirectional LSTM, and traditional machine learning approaches for energy consumption prediction," Sage Journal, vol. 43, no. 1, Aug. 2024, doi: https://doi.org/10.1177/0144598724126941.

[22] Yarin Gal and Zoubin Ghahramani, "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks," Oct. 2016, doi: https://doi.org/10.48550/arXiv.1512.05287.

[23] Pei-Chun Lin and Nureize Arbaiy, "BiLSTM-Based Approach for Music Genre Classification," International Conference on Industry Sciences and Computer Science Innovation, vol. 263, pp. 619–626, 2025, doi: https://doi.org/10.1016/j.procs.2025.07.074.

[24] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5–6, pp. 602–610, Aug. 2005, doi: DOI: 10.1016/j.neunet.2005.06.042.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014.

[26] Hung-Chi Chu, Young-Lin Zhang, and Hao-Chu Chiang, "A CNN Sound Classification Mechanism Using Data Augmentation," Department of Information and Communication Engineering, Aug. 2023, doi: https://doi.org/10.3390/s23156972.

[27] Osita Miracle Nwakeze, "Regularization-Based Solutions to Overfitting in Modern Predictive Models: A Review," Methods in Science and Technology Studies, vol. 1, no. 2, Jan. 2026, doi: https://doi.org/10.64539/msts.v1i2.2025.370.

[28] Jan Benedikt Ruhland, Iraj Masoudian, and Dominik Heider, "Enhancing deep neural network training through learnable adaptive normalization," Knowledge-Based Systems, vol. 326, no. 113968, Jul. 2025, doi: https://doi.org/10.1016/j.knosys.2025.113968.

[29] Sergey Ioffe and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR, vol. 37, 2015.

[30] M. A. S. Md Dzahir and K. S. Chia, "Batch Normalization and Dropout Regularization in Developing Lightweight Deep Learning Models for Electronic Nose Gas Classification," International Journal of Engineering, vol. 39, no. 09, pp. 2336–2345, Dec. 2025, doi: doi: 10.5829/ije.2026.39.09c.19.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning," Genetic Programming and Evolvable Machines, vol. 19, pp. 305–307, Oct. 2017.

[32] Reban Cliff A. Fajardo, Fe B. Yara, Randy F. Ardeña, Michael Kevin L. Hernandez, and Jan Carlo T. Arroyo, "A Data-Driven Approach in Predicting Scholarship Grants of a Local Government Unit in the Philippines Using Machine Learning," International Journal of Engineering Trends and Technology, vol. 72, no. 6, pp. 74–81, Jun. 2024, doi: https://doi.org/10.14445/22315381/IJETT-V72I6P108.