JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Music-Genre Classification using Bidirectional Long Short-Term Memory and Mel-Frequency Cepstral Coefficients

Nantalira Niar Wijaya[1], De Rosal Ignatius Moses Setiadi[1,*], and Ahmad Rofiqul Muslikh[2]

[1] Faculty of Computer Science, Dian Nuswantoro University, Semarang, Central Java 50131, Indonesia; e-mail : nantalirawijaya@gmail.com, moses@dsn.dinus.ac.id
[2] Faculty of Information Technology, University of Merdeka, Malang, East Java 65146, Indonesia; e-mail : rofickachmad@unmer.ac.id
* Corresponding Author : De Rosal Ignatius Moses Setiadi

Abstract: Music genre classification is one part of the music recommendation process, which is a challenging job. This research proposes the classification of music genres using Bidirectional Long Short-Term Memory (BiLSTM) and Mel-Frequency Cepstral Coefficients (MFCC) extraction features. This method was tested on the GTZAN and ISMIR2004 datasets, specifically on the IS-MIR2004 dataset, a duration cutting operation was carried out, which was only taken from seconds 31 to 60 so that it had the same duration as GTZAN, namely 30 seconds. Preprocessing operations by removing silent parts and stretching are also performed at the preprocessing stage to obtain normalized input. Based on the test results, the performance of the proposed method is able to produce accuracy on testing data of 93.10% for GTZAN and 93.69% for the ISMIR2004 dataset.

Keywords: Music genre classification; Music recommendation; Music suggestion; Song classification; Song recommendation.

1. Introduction

Along with the development of digitalization and internet technology, various platforms have emerged that are used for streaming music to replace previous media such as cassettes, compact disks (CDs) and MP3 players. One of the largest streaming media is Spotify, which, based on data obtained from Statista in the third quarter of 2023, the music streaming platform Spotify has had 574 million monthly active users[1]. Apart from the convenience it offers in listening to music, this platform is popular because it has a recommendation feature for listening to music [2]. One part of this music recommendation is classifying music based on genre. Music genre was chosen as the object of this research because music genre is closely related to a person's personality and musical tastes[3]. This classification helps users get recommendations for music they often, rarely, or never listen to or play[4].

Features are input from the classification process. In this research, features can be obtained from music extraction. Several features are suitable for classifying music genres, for example, Mel Frequency Cepstral Coefficients (MFCC) and Linear Prediction Cepstral Coefficients (LPCC)[5]. MFCC represents the power spectrum of an audio signal at short time intervals in a non-linear frequency scale called the mel scale. Meanwhile, LPCC is based on linear predictive analysis of audio signals, which separates the source and filter from the signal. MFCC was chosen in this study because the mel scale simulates human hearing when listening to music[6].

Classification can be done using Machine Learning (ML) or Deep Learning (DL) approaches[7]. Some popular ML models for classification tasks include Random Forest[8], Multi-Layer Perceptron (MLP)[9], K-Nearest Neighbor (K-NN)[10], Support Vector Machine (SVM)[11], [12], and Naive Bayes (NB)[13]. Meanwhile, several DL models are also popular for carrying out classification tasks, such as Long Short-Term Memory (LSTM)[14], Convolution neural network (CNN)[15], and Gated Recurrent Unit (GRU)[16]. Both of these approaches have their respective advantages and disadvantages, so when using them, a deeper analysis of the objects to be classified is necessary. ML approaches tend to process smaller

Received: December, 9th 2023
Revised: January, 7th 2024
Accepted: January, 8th 2024
Published: January, 9th 2024

amounts of data and lower computing resources, while DL approaches are used to process large amounts of complex data with high accuracy. One study that uses ML is [17]. This research compares the Gaussian Mixture Model (GMM), K-NN, and SVM models. Based on experiments, each model produces an accuracy of 64.3%, 69.7%, and 61.2% for GMM, K-NN and SVM respectively. These results show that, in general, the classification method using ML still needs to be improved.

The number of songs released yearly is always increasing, but this will become a new problem because ML will be less reliable on larger datasets. This differs from the DL approach, which is usually carried out on larger datasets [18]. This approach is suitable for classifying large amounts of music data. Research [19] used a DL approach with the Masked Conditional Neural Network (MCLNN) model to classify music genres. This approach was proven to be superior to classification using the ML approach, with the MCLNN model producing an accuracy of 85.10%. The DL approach is increasingly being developed to classify music genres. The LSTM method is also proposed. LSTM has several advantages that are suitable for classifying music genres, such as the ability to process time series data[20]. LSTM can remember past information in memory and can process entire data sequences. Apart from that, LSTM also has various architectures that can be used, one of which is the Bidirectional LSTM (BiLSTM) architecture.

BiLSTM is a type of neural network architecture that allows the model to learn patterns from both directions, namely from past to future and from future to past[21]. In this architecture, data is fed to the LSTM from direct and reverse order, allowing the model to utilize contextual information from musical signals from past and future. This architecture can potentially produce more optimal accuracy than ordinary LSTM architectures. In research [22] also applied the LSTM method and the results were quite satisfactory, so in this study a comparison was also carried out with replication techniques.

A model that can learn more complex data is needed to get maximum accuracy. However, this also results in higher computational complexity of the DL model, which can allow overfitting to occur [23]. This happens because the DL model will be designed with more neurons in each layer and even the layer itself to make the computation complex[24]. The more neurons used, the greater capacity the model has to learn complex patterns and representations from the data, resulting in high accuracy. BiLSTM model customization needs to be implemented with techniques such as layer normalization to overcome overfitting. By applying this technique to the BiLSTM architecture, it is hoped that it can produce high accuracy and a low level of overfitting, so that it can outperform several previous methods. Based on this background, this research aims to:

1. Design a customized BiLSTM model and MFCC feature extraction so that it can work optimally and accurately for music genre classification.
2. Applying the BiLSTM model to the GTZAN and ISMIR2004 datasets to compare with the performance of other models such as LSTM and MCLNN in music genre classification based on the results of training data accuracy, test data accuracy, confusion matrix, and several metrics such as precision, recall, f1-score, and support for each class presented in the classification report.

The rest of the paper is presented in four sections. In the second section, related work and research gaps are explained. The third section explains the proposed method and research stages from data collection to evaluation. The fourth section presents the results, comparisons, and analysis, and the last is the conclusion section.

2. Related Work

Research into the classification of music genres has been carried out by several previous studies, both using ML and DL. Research [25] discusses the classification of the GTZAN music dataset, where each piece of music from the dataset is divided into six parts with a duration of 5 seconds, from which an image is then generated from the extraction results with Mel-Spectrogram. The image will be classified using MusicRecNet, which is capable of producing an accuracy of 81.8%.

Research [26] also proposed classifying music genres by converting them into images and then classifying them with a CNN model. The proposed CNN model is a customized result consisting of convolutional and fully connected layers. This model is applied to 1880 music with genres: Pop, Country, Rap, Rock, Alternative, R&B Soul, Dance, Hiphop,

Classical, and Electronic. This research also converts music in (.mp3) format into a grayscale spectrogram image. These images were then augmented so that 15,000 spectrograms were obtained. Apart from being augmented, the number of images per genre is also balanced. As a result, this model obtained music genre classification accuracy of 85% on test data.

Study [19] applied the MCLNN model to several datasets, including GTZAN and ISMIR2004. In these two datasets, each piece of music will be extracted using the Mel-Spectogram and will be trained and tested using the 10-fold cross-validation technique. The GTZAN and ISMIR2004 datasets have different characteristics for each piece of music. GTZAN consists of music with a duration of 30 seconds, while ISMIR2004 consists of complete music with different durations. This causes the ISMIR2004 dataset to only take 30 seconds of music after the first 30 seconds of each piece before being extracted. The results taken in this study are the average of the 10-fold cross-validation results, where the GTZAN dataset produces an accuracy of 85.10% and the ISMIR2004 dataset produces an accuracy of 86.04%.

Another research [27] uses several ML models and one DL model which will be used on three datasets, including GTZAN, ISMIR2004, and Latin Music. The DL learning model gets the best results among the ML models, and the model is a Neural Network (NN) with a training algorithm, namely trainingdx. Before training and testing the model with 10-fold cross-validation, the dataset will be extracted using African Buffalo Optimization (ABO). The results obtained from the average accuracy on each dataset were 83.33% on the GTZAN dataset, 82.03% on the ISMIR2004 dataset, and 80.79% on the Latin Music dataset.

Based on several studies that have been carried out on Music Genre objects, using the DL approach gets better results than the ML approach. Therefore, this research uses the BiLSTM model, one of the DL approach models. This model will be applied to the GTZAN and ISMIR2004 datasets. The final results obtained from this research are training accuracy and testing accuracy, which can be compared with previous research.

3. Proposed Method

The method proposed in the research consists of several stages which are illustrated in Figure 1.

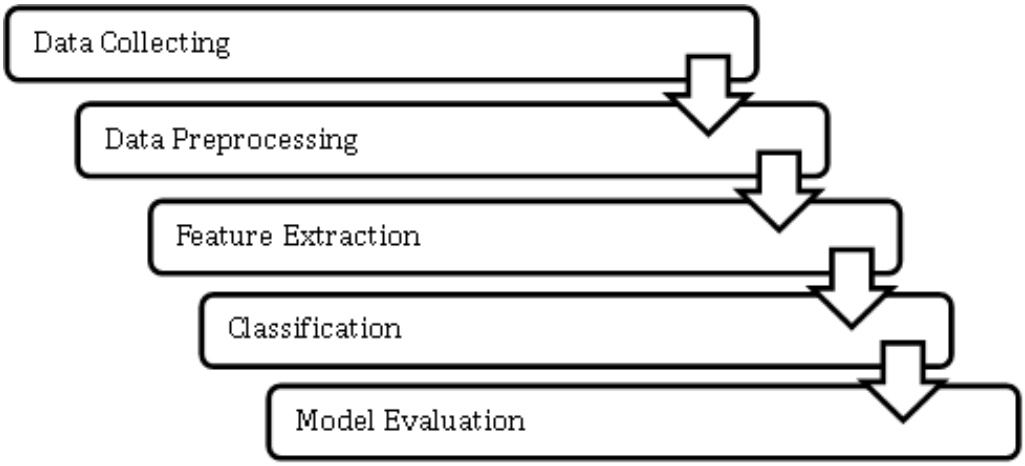


Figure 1. Method stages

3.1. Data Collection

This research uses two datasets, namely the GTZAN dataset [28] and ISMIR2004[29]. The GTZAN dataset is a dataset with balanced classes, containing 1000 music with ten genres including: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock, where each genre contains ten music with formats (.wav) and duration 30 seconds. Meanwhile, the ISMIR2004 dataset is an imbalance dataset containing 729 music with eight genres, and each genre contains complete music in format (.mp3) and different durations with a division into each genre as follows: 320 classical music, 115 electronic music, 26 jazz music, 29 music metal, six pop music, 16 punk music, 95 rock music, 122 world music. Therefore, data preprocessing is needed to facilitate classification by the BiLSTM model.

3.2. Preprocessing

Before carrying out the preprocessing stages, specifically for the ISMIR2004 dataset, it is necessary to change the format from (.mp3) to (.wav). This improves music quality and simplifies audio processing[30]. The ISMIR2004 dataset has varied music durations, so it needs to be standardized like the GTZAN dataset by cutting the duration to 30 seconds as in research[19]. Figure 2 is an example of one of the pieces of music in the ISMIR2004 dataset that was cut for duration. The sample dataset after cutting is presented in Figure 3.

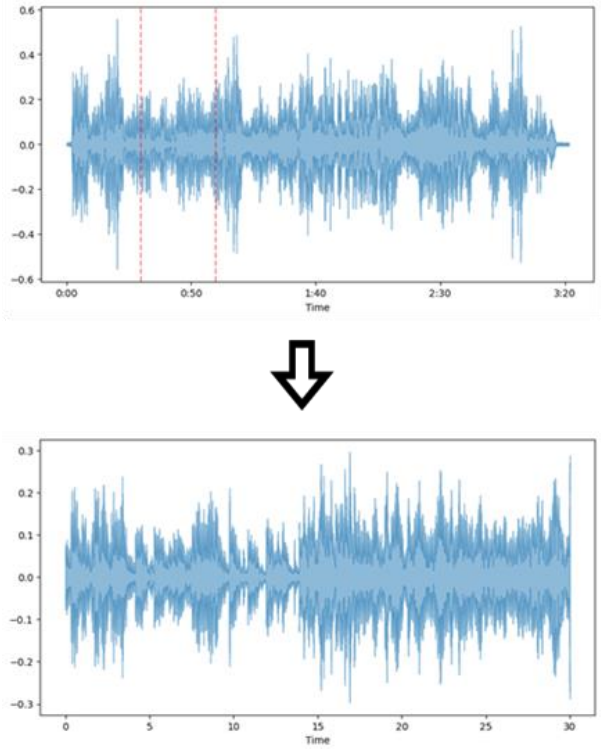
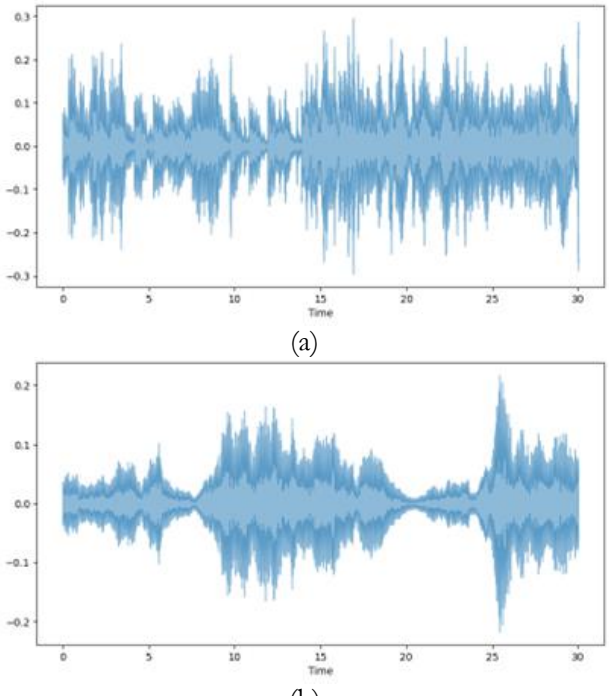


Figure 2. ISMIR2004 Dataset Cutting Sample.



(a)

(b)

Figure 3. Sample dataset after cutting (**a**) ISMIR2004; (**b**) GTZAN.

Next, the silent part of each piece of music is deleted, whereas if there is sound in the music part with a noise level below -50dB, it will be deleted. The decibel unit must be converted into numeric as an amplitude threshold with the formula $10^{(db/20)}$resulting in an amplitude threshold ≈ of 0.00316. Music audio that has had the silent part removed can be seen in Figure 4.

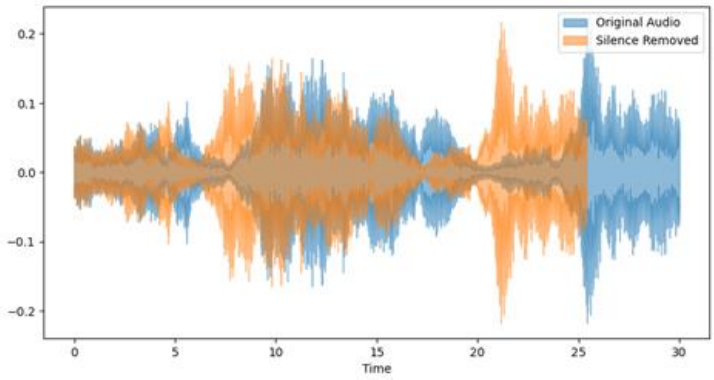


Figure 4. Silent Part Removal

To maintain the length of the audio duration, an audio stretching technique is used using time_stretch. This technique requires a parameter in the form of a rate, which is obtained from the formula $\frac{\text{number of audio samples}}{\text{sample rate} \times \text{duration}}$. Figure 5 presents an example of visualization of the audio stretching results that have had the silent part removed.

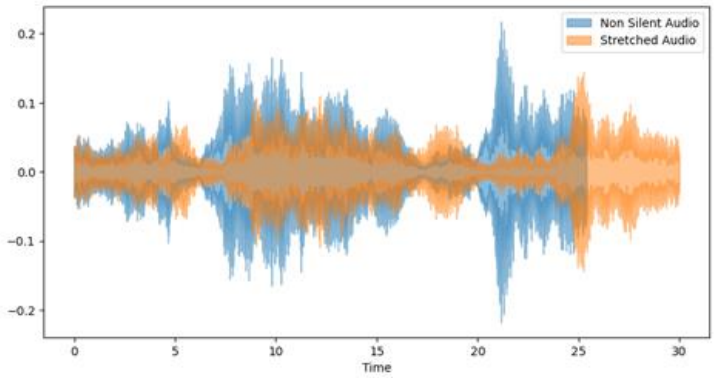


Figure 5. Streching Audio

The audio data was then cut into 10 parts to increase the data. This cutting section can be visualized in Figure 6. Thus, quantitatively the total records increase 10 times, for more clarity you can see Table 1.

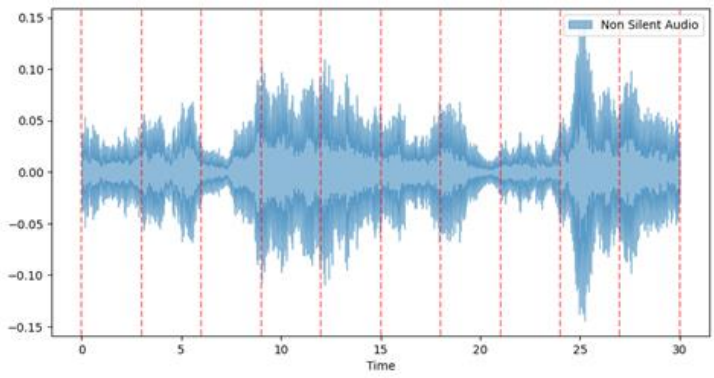


Figure 6. Visualization of an example of music cut into 10 parts

Table 1. Dataset Details Before and After Preprocessing

Genre	GTZAN		ISMIR2004	
	Before	After	Before	After
classical	100	1000	320	3200
jazz	100	1000	26	260
metal	100	1000	29	290
pop	100	1000	6	60
rock	100	1000	95	950
blues	100	1000	-	-
country	100	1000	-	-
disco	100	1000	-	-
hiphop	100	1000	-	-
reggae	100	1000	-	-
electronic	-	-	115	1150
punk	-	-	16	160
world	-	-	122	1220
Total	1000	10000	729	7290

The audio preprocessing results will be divided into training data, validation data, and testing data, each with a sequential ratio of 80:10:10. This division is carried out for each genre so that the resulting data has the same portion. The data is then collected into dictionary-type variables.

3.3. Feature Extraction

The feature extraction used is MFCC with the Librosa library. This function will carry out the steps in MFCC extraction such as Frame Blocking, Windowing, FFT calculation, implementation of Mel Filter Bank, and DCT. The results obtained from the MFCC feature extraction process are presented in Figure 7.

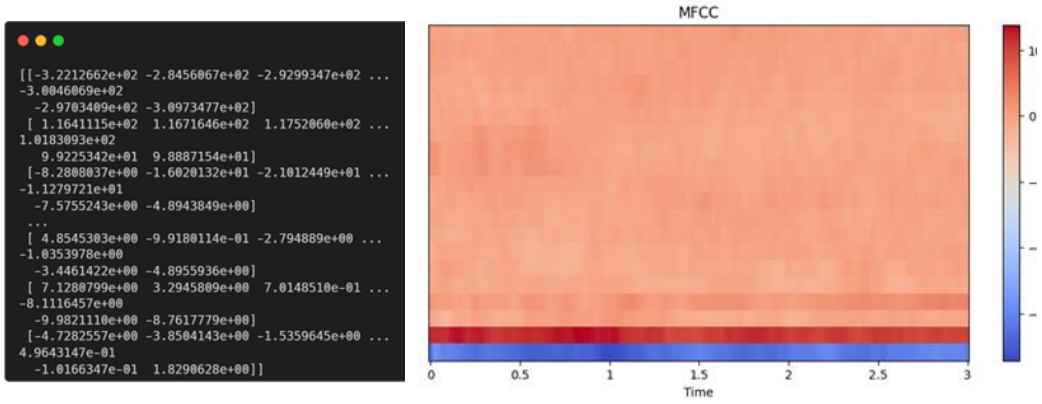


Figure 7. MFCC Feature Extraction Results (left) and its Visualization (right)

The x-axis of the visualization presented in Figure 7 is time, and the y-axis is the number of MFCC coefficients. The MFCC coefficient is the main parameter which is usually selected from 13 to 20 coefficients [31]. In this study, 20 MFCC coefficients were used, and delta and delta^2 were added. Delta functions to measure cepstral changes between adjacent time frames, providing information about the speed of change in the sound spectrum. While delta^2 The second derivative of the MFCC coefficient, measuring the acceleration of cepstral changes, provides information about acceleration or deceleration. Delta and delta2 each also have 20 coefficients. Combining the three, of course, increases the information in music data, making it easier to represent dynamic information in sound signals. An illustration of delta, delta2, and their combination with MFCC is presented in Figure 8.

The results obtained from feature extraction with MFCC are (X0, X1, X2). X0 is the amount of music data extracted, X1 is the number of MFCC vectors generated for each music data, and X2 is the number of MFCC coefficients extracted at each time step. The data that will be used as input shapes in the BiLSTM model are X1 and X2, with the numbers (130,

60). The 60 MFCC coefficients are obtained from a vertical combination of 20 MFCC coefficients, 20 delta coefficients, and 20 delta^2 coefficients, which form a more complete set of features.

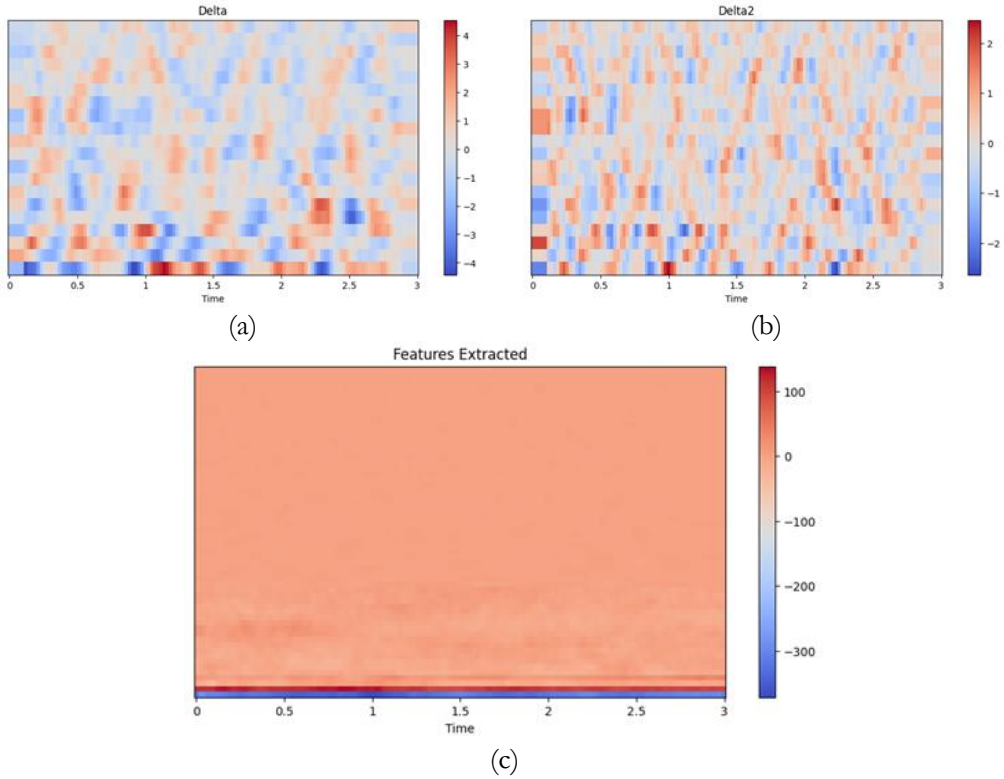


Figure 8. (**a**) Delta; (**b**) Delta2; (**c**) Combined MFCC, Delta, and Delta2.

3.4. Classification

The BiLSTM model was created using a sequential model from the Keras library. The model has several layers, such as Bidirectional LSTM and a 1D Global Average Pooling layer. The normalization layer is also applied to this model to reduce over-fitting. The last layer applied is the output layer with the Softmax activation function. An overview of the detailed BiLSTM model can be presented in Tabel 2.

Table 2. Detail of BiLSTM Model.

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional) return_sequence=True	(None, 130, 1024)	2347008
layer_normalization (Layer Normalization)	(None, 130, 1024)	2048
global_average_pooling1d (GlodbalAveragePooling1D)	(None, 1024)	0
dense (Dense)	(None, 10)	10250
Total params: 2359306 (9.00 MB), Trainable params: 2359306 (9.00 MB), and Non-trainable params: 0 (0.00 Byte)		

The BiLSTM model will then be compiled using functions from the Keras library. The function requires several parameters, such as an optimizer, loss function, and metrics that will be used to evaluate the model as it is trained and validated. Adam is the chosen optimizer, while the loss function used is sparse_categorical_crossentropy, and the accuracy metric will be used to evaluate performance at each epoch.

The model will be trained using train data and validated with validation data. In addition, the fit function in the Keras library requires several more parameters to train the model, such as batch size and epochs. The batch size and epochs applied this time are 64 and 30,

respectively. One more optional parameter can be applied to this function, namely callback. This parameter implements early stopping and checkpoints when training the model. The model will stop training when it sees val_loss not decreasing for several epochs, and the model that has the best weight will be saved in .h5 format. Each epoch performed during model training will be saved into a variable to display the model training history in a line diagram, for more details, see Tabel 3.

Table 3. Hyperparameter and Tuning Parameter for BiLSTM

Parameter Name	Values
Optimizer	Adam (Learning rate = 0.001)
Loss funtion	sparse_categorical_crossentropy
Metrics	accuracy
Callback	Early Stopping (monitor = val_loss, patience = 10, restore best weights) Model Checkpoint (save best only)
Batch size	128
Epoch	30

3.5. Evaluasi Model

Apart from evaluating the model by looking at line diagrams, test data has also been prepared to test the model against new data in this training. In the sklearn.metric library, there are classification reports, and confusion matrix functions to evaluate the model more clearly. The classification report presents several metrics, such as each class's precision, recall, and accuracy. Meanwhile, the confusion matrix shows how good or bad the model is in making predictions for each class. By applying evaluation to the model with several techniques, it is hoped that it can be used as a benchmark for the performance of the model that will be implemented.

4. Results and Discussion

This research was implemented using the Python language and Jupyter Notebook as an editor. In particular, the preprocessing of the ISMIR2004 dataset was handled using Visual Studio Code on a local computer because the dataset is large. Because the DL model training process requires quite a lot of computing, this research will use Google Colab. So, the preprocessing results on the ISMIR2004 dataset will be exported using pickle (.pkl) format. The training and validation process of the proposed BiLSTM model is carried out in epoch 30where the training and validation accuracy graphs are presented in Figure 9 and Figure 10, while the loss graphs are in Figure 11 and Figure 12, respectively, for the GTZAN dataset and ISMIR2004 dataset. Then, the accuracy and loss in the last epoch are summarized in Table 4.

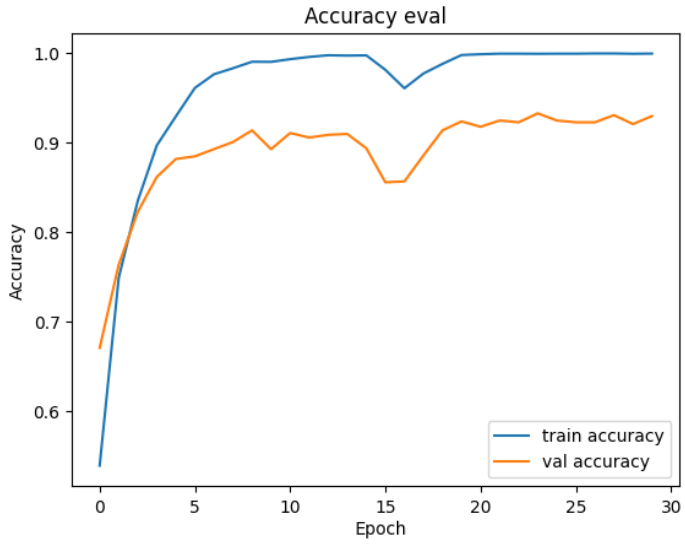


Figure 9. Accuracy Plot on the GTZAN dataset

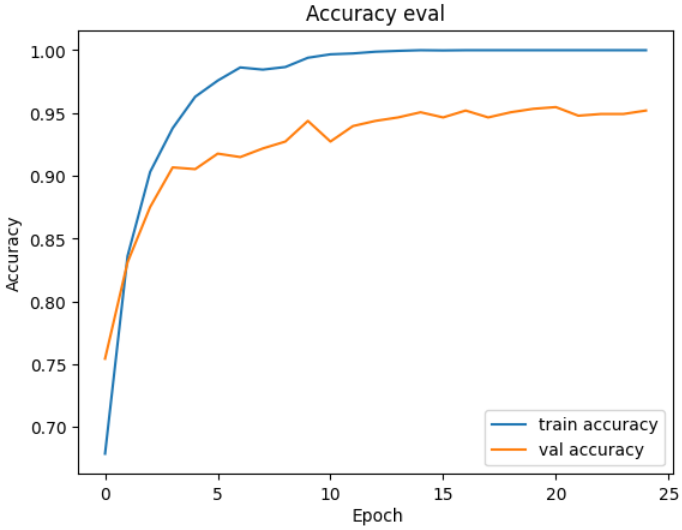


Figure 10. Accuracy Plot on the ISMIR2004 dataset

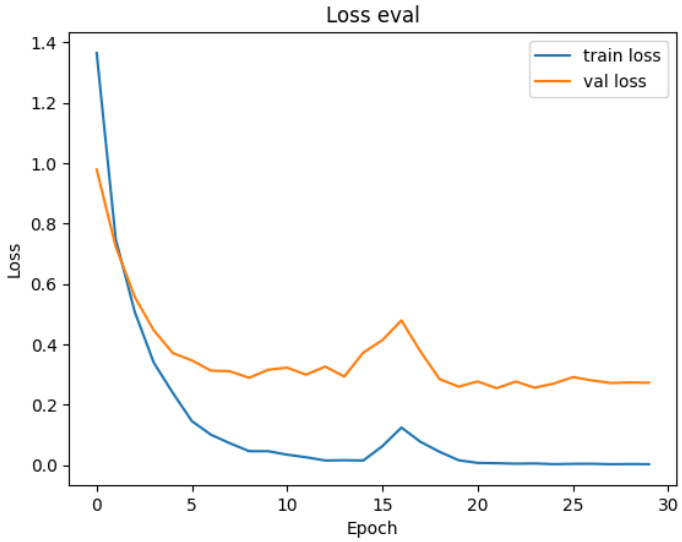


Figure 11. Loss Plot on the GTZAN dataset

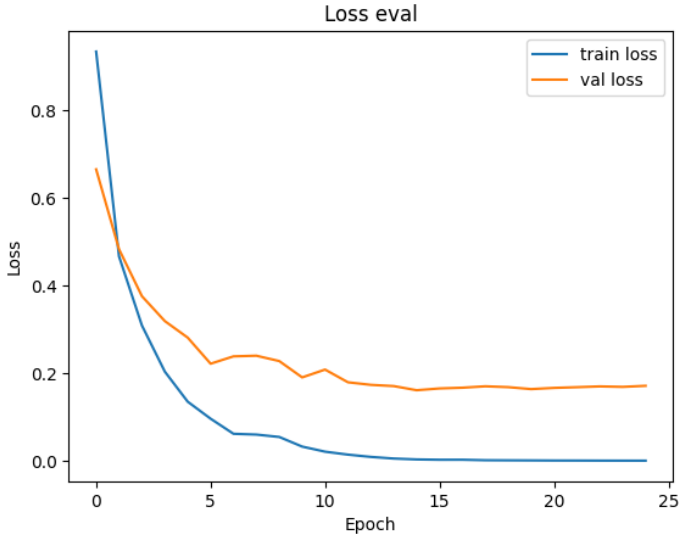


Figure 12. Loss Plot on the ISMIR2004 dataset

Based on accuracy measurements, both have high accuracy, namely above 90% for validation data. However, the comparison is needed to validate the performance of the proposed BiLSTM. Table 5 compares the proposed method's performance with an LSTM model replicated from the method [22] but with the same preprocessing approach as the proposed method. A more detailed design of the replicated LSTM model is presented in Figure 13. Apart from that, the results are also compared with research[19]. It appears that the proposed method is superior in validation accuracy. This confirms the theory that BiLSTM bidirectional analysis can improve accuracy performance.

Table 4. Accuracy and Loss Training and Validation from Last Epoch

	GTZAN	**ISMIR2004**
Data Training Accuracy	99.87%	100.00%
Data Validation Accuracy	92.90%	95.20%
Data Training Loss	0.35%	0.05%
Data Validation Loss	27.32%	17.13%

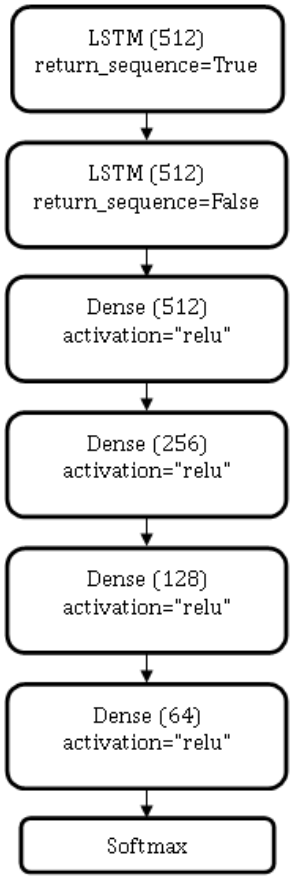


Figure 13. Replicated LSTM Model Details

Table 5. Validation Accuracy Comparison with Prior-art

Dataset	**LSTM**	**MCLNN[19]**	**BiLSTM**
GTZAN	84.00%	85,10%	92.90%
ISMIR2004	89.99%	86,04%	95.20%

The final test in this research is on test data. The measuring tools used on the test data are the confusion matrix and classification report from the scikit-learn library. The measurements are very complete, consisting of precision, recall, f1-score, and accuracy. The results of the confusion matrix measurements are presented in Figures 14 and 15 for GTZAN and ISMIR2004, respectively, while the classification report is presented in Figures 16 and 17.

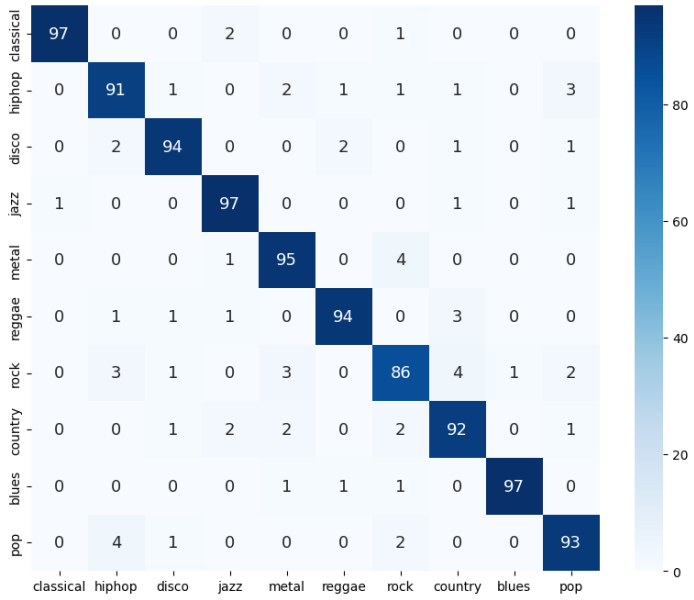


Figure 14. Confusion Matrix GTZAN

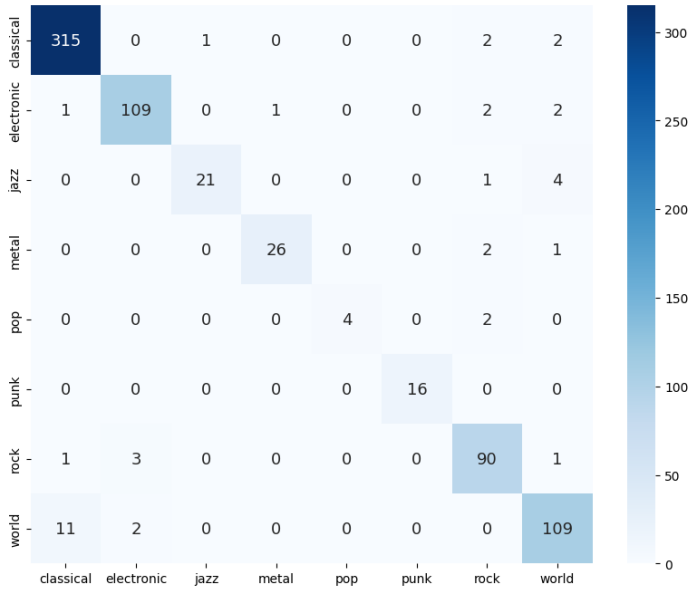


Figure 15. Confusion Matrix ISMIR2004

Based on the test data measurement results for the balanced GTZAN dataset, it appears that precision, recall, f1-score, and accuracy are all 94%. In more detail, the precision level per class, the minimum is 89% and a maximum of 99%, the minimum recall is 86% and a maximum of 97%, and f1-score is a minimum of 87% and a maximum of 98%. Meanwhile, in the imbalanced ISMIR2004 dataset, the average precision, recall, f1-score, and accuracy values were 95%. More detail produces a minimum precision of 91% and a maximum of 100%. For minimum recall of 67% and maximum of 100% and f1-score minimum of 80% and maximum of 100%. Especially the pop class is something that needs to be highlighted because the precision is 100%, the recall is 67%, and the f1-score is 80%. This means that even though the model is very good at classifying positive instances of the "pop" class, the model tends to miss some positive instances that should be classified, this may be due to the very small sample size, namely only six records, so that the model has difficulty accurately generalizing the class pattern. But overall, it appears that the proposed method can perform very well on two datasets with different characteristics.

Classification Report

	precision	recall	f1-score	support
classical	0.99	0.97	0.98	100
hiphop	0.90	0.91	0.91	100
disco	0.95	0.94	0.94	100
jazz	0.94	0.97	0.96	100
metal	0.92	0.95	0.94	100
reggae	0.96	0.94	0.95	100
rock	0.89	0.86	0.87	100
country	0.90	0.92	0.91	100
blues	0.99	0.97	0.98	100
pop	0.92	0.93	0.93	100
accuracy			0.94	1000
macro avg	0.94	0.94	0.94	1000
weighted avg	0.94	0.94	0.94	1000

Figure 16. Classification Report GTZAN

Classification Report

	precision	recall	f1-score	support
classical	0.96	0.98	0.97	320
electronic	0.96	0.95	0.95	115
jazz	0.95	0.81	0.88	26
metal	0.96	0.90	0.93	29
pop	1.00	0.67	0.80	6
punk	1.00	1.00	1.00	16
rock	0.91	0.95	0.93	95
world	0.92	0.89	0.90	122
accuracy			0.95	729
macro avg	0.96	0.89	0.92	729
weighted avg	0.95	0.95	0.95	729

Figure 17. Classification Report ISMIR2004

5. Conclusion

Based on the results achieved, this research proves that the BiLSTM model performs satisfactorily in classifying music genres. With training data accuracy results of 99.87% and 94.60% test data accuracy on the GTZAN dataset, as well as 100.00% training data accuracy and 94.65% test data accuracy on ISMIR2004, it proves that the method can work quite stably on both balance and imbalance datasets. In addition, this research produces good accuracy compared to the accuracy in review studies and model replication.

Research on the classification of music genres using DL is complex research. Apart from setting up DL models as a tool for classification, feature extraction in music also has an important role. Therefore, in future research, we can apply feature extraction methods other than MFCC. Apart from that, in future research, you can try the latest dataset, considering that as time goes by, music genres become more varied, so trying the latest dataset it will make the DL model more relevant to the times.

Author Contributions: Conceptualization: N.N.W. and D.R.I.M.S.; methodology, N.N.W. and D.R.I.M.S.; software: N.N.W.; validation: N.N.W. and D.R.I.M.S.; formal analysis: N.N.W; investigation: N.N.W. and D.R.I.M.S.; resources: N.N.W; data curation: N.N.W; writing—original draft preparation: N.N.W; writing—review and editing: N.N.W, D.R.I.M.S, and A.R.M.; visualization: N.N.W; supervision: D.R.I.M.S.; project administration: D.R.I.M.S. and A.R.M.; funding acquisition: all.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in this research is GTZAN which can be downloaded at the URL: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification; and ISMIR2004 can be downloaded at the URL:: https://zenodo.org/records/1302992.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] G. Marie, "Number of Spotify monthly active users (MAUs) worldwide from 1st quarter 2015 to 3rd quarter 2023tle," 2023.
[2] J.-W. Chang *et al.*, "Music recommender using deep embedding-based features and behavior-based reinforcement learning," *Multimed. Tools Appl.*, vol. 80, no. 26–27, pp. 34037–34064, Nov. 2021, doi: 10.1007/s11042-019-08356-9.
[3] R. Brisson and R. Bianchi, "On the relevance of music genre-based analysis in research on musical tastes," *Psychol. Music*, vol. 48, no. 6, pp. 777–794, Nov. 2020, doi: 10.1177/0305735619828810.
[4] Y. Liang and M. C. Willemsen, "Personalized Recommendations for Music Genre Exploration," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, Jun. 2019, pp. 276–284. doi: 10.1145/3320435.3320455.
[5] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Appl. Acoust.*, vol. 158, p. 107020, 2020, doi: 10.1016/j.apacoust.2019.107020.
[6] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Convolutional Neural Networks Approach for Music Genre Classification," in *2020 International Symposium on Computer, Consumer and Control (IS3C)*, Nov. 2020, pp. 399–403. doi: 10.1109/IS3C50286.2020.00109.
[7] J. Ramírez and M. J. Flores, "Machine learning for music genre: multifaceted review and experimentation with audioset," *J. Intell. Inf. Syst.*, vol. 55, no. 3, pp. 469–499, Dec. 2020, doi: 10.1007/s10844-019-00582-9.
[8] A. Parmar, R. Katariya, and V. Patel, "A Review on Random Forest: An Ensemble Classifier," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 26, 2019, pp. 758–763. doi: 10.1007/978-3-030-03146-6_86.
[9] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *Advances in Computers*, 1st ed., vol. 117, no. 1, Elsevier Inc., 2020, pp. 339–368. doi: 10.1016/bs.adcom.2019.09.007.
[10] P. Cunningham and S. J. Delany, "k-Nearest Neighbour Classifiers - A Tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, Jul. 2022, doi: 10.1145/3459665.
[11] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, no. xxxx, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.
[12] D. R. Ignatius Moses Setiadi *et al.*, "Effect of Feature Selection on The Accuracy of Music Genre Classification using SVM Classifier," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2020, pp. 7–11. doi: 10.1109/iSemantic50169.2020.9234222.
[13] D. R. Ignatius Moses Setiadi *et al.*, "Comparison of SVM, KNN, and NB Classifier for Genre Music Classification based on Metadata," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2020, pp. 12–16. doi: 10.1109/iSemantic50169.2020.9234199.
[14] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," pp. 1–42, Sep. 2019.
[15] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
[16] N. Elsayed, A. S. Maida, and M. Bayoumi, "Gated Recurrent Neural Networks Empirical Utilization for Time Series Classification," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Jul. 2019, pp. 1207–1210. doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00202.
[17] M. Jakubec and M. Chmulik, "Automatic music genre recognition for in-car infotainment," in *Transportation Research Procedia*, 2019, vol. 40, pp. 1364–1371. doi: 10.1016/j.trpro.2019.07.189.
[18] S. Christin, É. Hervet, and N. Lecomte, "Applications for deep learning in ecology," *Methods Ecol. Evol.*, vol. 10, no. 10, pp. 1632–1644, Oct. 2019, doi: 10.1111/2041-210X.13256.
[19] F. Medhat, D. Chesmore, and J. Robinson, "Masked Conditional Neural Networks for sound classification," *Appl. Soft Comput.*, vol. 90, p. 106073, May 2020, doi: 10.1016/j.asoc.2020.106073.
[20] F. Karim, S. Majumdar, and H. Darabi, "Insights Into LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 7, pp. 67718–67725, 2019, doi: 10.1109/ACCESS.2019.2916828.
[21] D. Utebayeva, A. Almagambetov, M. Alduraibi, Y. Temirgaliyev, L. Ilipbayeva, and S. Marxuly, "Multi-label UAV sound classification using Stacked Bidirectional LSTM," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, Nov. 2020, pp. 453–458. doi: 10.1109/IRC.2020.00086.
[22] S. Deepak and B. G. Prasad, "Music Classification based on Genre using LSTM," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Jul. 2020, pp. 985–991. doi: 10.1109/ICIRCA48905.2020.9182850.
[23] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, "Research on Overfitting of Deep Learning," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2019, pp. 78–81. doi: 10.1109/CIS.2019.00025.

[24] Z. Wang, M. Yan, J. Chen, S. Liu, and D. Zhang, "Deep learning library testing via effective model generation," in Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Nov. 2020, pp. 788–799. doi: 10.1145/3368089.3409761.
[25] A. Elbir and N. Aydin, "Music genre classification and music recommendation by using deep learning," *Electron. Lett.*, vol. 56, no. 12, pp. 627–629, Jun. 2020, doi: 10.1049/el.2019.4202.
[26] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," *Can. J. Electr. Comput. Eng.*, vol. 43, no. 3, pp. 170–173, 2020, doi: 10.1109/CJECE.2020.2970144.
[27] B. Jaishankar, R. Anitha, F. Daniel Shadrach, M. Sivarathinabala, and V. Balamurugan, "Music Genre Classification Using African Buffalo Optimization," *Comput. Syst. Sci. Eng.*, vol. 44, no. 2, pp. 1823–1836, 2023, doi: 10.32604/csse.2023.022938.
[28] Andrada, "GTZAN Dataset - Music Genre Classification," *kaggle*, 2019.
[29] P. Cano, N. Wack, and P. Herrera, "ISMIR04 Genre Identification task dataset (1.0) [Data set]," *Zenodo*, 2018.
[30] T. Hidayat, M. H. Zakaria, and N. Che Pee, "Comparison of Lossless Compression Schemes for WAV Audio Data 16-Bit Between Huffman and Coding Arithmetic," *Int. J. Simul. Syst. Sci. Technol.*, vol. 19, no. 6, pp. 36.1-36.7, Feb. 2019, doi: 10.5013/IJSSST.a.19.06.36.
[31] M. Ashraf *et al.*, "A Hybrid CNN and RNN Variant Model for Music Classification," *Appl. Sci.*, vol. 13, no. 3, p. 1476, Jan. 2023, doi: 10.3390/app13031476.