SN Computer Science (2026) 7:146
https://doi.org/10.1007/s42979-026-04747-6

ORIGINAL RESEARCH

Decoding Cultural Music Classification with Machine Learning and Segment Length Analysis

Mesfin Abebe[1] · Leta Endashew[2] · Jukka Heikkonen[3] · Rajeev Kanth[4] · Sudhir Kumar Mohapatra[3]

Received: 24 September 2025 / Accepted: 10 January 2026

Abstract

Music is a sound composed with rhythm, melody, or harmony, and it is evolving in time and with the culture of the society. The manual way of classifying and searching of the Ethiopian music is time consuming, and expensive. In this study, an Ethiopian cultural music classification models are proposed to simplify this task. A large number of Ethiopian cultural music are collected from YouTube and other online music database. This data set has 10 classes based on the music genre. Eight machine learning algorithms are employed to build the classification models: Logistic Regression, Naive Bayes, KNN, MLP, SVM, Decision Trees, Random Forest, and Adaboost algorithms. The models are optimized using manual hyperparameter tuning, randomized search, grid search, genetic algorithm (TPOT classifier), Bayesian (hyperopt), and Optuna optimization techniques. Based on the experiments, Random Forest outperforms the other algorithms with 80% accuracy. Statistical analysis using one-way ANOVA confirmed that optimization significantly improved classification performance ($p<0.05$). Confusion matrix analysis revealed that certain regional styles, such as Gojam and Somali, were more prone to misclassification due to overlapping rhythmic and tonal features. The results demonstrate that machine learning can effectively classify Ethiopian cultural music and provide a foundation for developing intelligent music retrieval and recommendation systems.

Keywords Music classification · Ethiopia music · Cultural style · Machine learning · Feature extraction · Model optimization

Introduction

Music is a universal form of expression that reflects a society's rich cultural identity, existing values, and proud traditions. It conveys personal, social, and religious emotions. It also preserves the historical essence of communities. Ethiopia, with its rich diversity of ethnic groups, exhibits distinct musical styles that represent regional lifestyles and heritage. Recognizing these cultural patterns enables listeners to associate a piece of music with its originating community.

The process of musical classification generally involves four key stages: data preparation, feature extraction, model training, and optimization. Audio features are derived from waveforms using Fourier and signal processing techniques, followed by preprocessing, algorithm selection, training, evaluation, and model refinement.

Music categorization plays significant part in defining the features of music. In general, many studies have been conducted on internationally known musical genre classifications [1–6]. In contrast, Ethiopian Music Classification [7, 8] do not specifically focus on how to use machine learning techniques to classify music based on cultural styles. The manual labelling process of cultural style-based music classification is time-consuming, expensive, and requires experts. Moreover, due to its subjective nature, manual labelling is inconsistent and impractical for large size data. Similarly, performers and composers may gravitate toward music from other cultural musical styles in order to create musical sounds that lie on the boundaries between genres, making manual labelling difficult. Many digital music

✉ Sudhir Kumar Mohapatra
skmoha@utu.fi

1 Adama Science and Technology University, Adama, Ethiopia

2 CPU Business and Information Technology College, Adama, Ethiopia

3 University of Turku, Turku, Finland

4 Savonia University of Applied Sciences, Kuopio, Finland

Published online: 30 January 2026

classification techniques have been introduced, but most of them have been implemented on standard Western music records. For Ethiopian music classification, there is no well-prepared dataset that significantly improves and simplifies modelling process.

For archiving or other related purposes, Ethiopian music classification has been done manually by music experts by listening to all or some segments of the music. This common method of classification reduces the sustainability of original Ethiopian music as it requires more experts to verify the accuracy of the classification. This difficulty can be reduced by using machine learning techniques to classify the Ethiopian music. Recently, some attempts have seen to develop a classification model for Ethiopian music based on four pentatonic scale functions. Yet, the Ethiopia's musical style is diverse and the numbers of classes (genres) are large which makes unfeasible to use these classification models without improving them.

Ethiopian cultural music represents a diverse and significant part of the nation's heritage, yet it has received little attention in computational music classification research. Most existing approaches rely on limited datasets, manual labelling, or focus primarily on Western music genres. The lack of a comprehensive Ethiopian music dataset and systematic evaluation of segment length and model optimization techniques delays the development of accurate and scalable classification models. This study addresses this gap by proposing a machine learning–based approach for classifying Ethiopian cultural music styles using extracted audio features and optimized learning algorithms.

The main objective of this study is the development of a machine learning framework capable of classifying Ethiopian cultural music styles. The remainder of this paper is organized as follows. Section 2 presents the related work and discusses previous studies on music classification and their limitations. Section 3 explains the research methodology adopted in this study, including data collection, preprocessing, and experimental setup. Section 4 describes the proposed approach and optimization framework used for Ethiopian cultural music classification. Section 5 reports and discusses the experimental results, including performance evaluation, error analysis, and comparative assessment with baseline methods. Finally, Sect. 6 concludes the study and outlines possible directions for future research.

Related Works

Automatic music classification has been an active research area for the past two decades. Early approaches relied on features such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma, and spectral descriptors for genre or instrument recognition. Tzanetakis and Cook [1] came out with one of the first systematic studies using statistical pattern recognition on audio signals. Subsequent works explored algorithms such as Support Vector Machines (SVM), Decision Trees, and Random Forests for feature-based classification [2, 3]. The models such as deep learning, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are now a days dominant in classification of music. Studies [4–6] suggest that proper feature selection and different segment length play a major role in the classification accuracy. This study found that limited research has experimented their models with cultural or regional music, like Indian or African traditional music. In the Ethiopian context, Terefe [7] and Selam [8] applied conventional machine learning and hybrid deep learning techniques, achieving accuracies of 85–86%. However, their datasets were relatively small, and optimization techniques were not explored. Sharma et al. [9] applied traditional machine learning approaches such as Support Vector Machines and Random Forests for music genre classification, demonstrating the effectiveness of handcrafted spectral and MFCC features but limited generalization across diverse cultural datasets. These models automatically learn hierarchical representations from spectrograms, outperforming traditional classifiers on benchmark datasets [10, 11]. In this regard, experiment by Choi et al. [10] and Pons et al. [11] demonstrated that CNN-based architectures achieve improved generalization across different music genres. In view of feature extraction, researchers have used Fourier analysis and time frequency domain transformations to represent the musical content in a computable format.

More recent studies such as Aggarwal et al. [12] and Raiaan et al. [13] emphasize the use of model optimization. Therefore, this study extends the existing body of work by focusing on Ethiopian cultural music classification using machine learning and optimization techniques. It investigates the influence of audio segment length, model choice, and optimization strategies on performance, providing an analytical foundation for computational ethnomusicology. Table 1 present a summery of the related work along with the limitation of the existing work.

Research Methodology

The performance of a machine learning model is dependent on the dataset quality that is used to build it. In this study, various data points of the Ethiopian cultural music are systematically collected from the Internet. These data sources are online music stores such as YouTube and oromp3. The data points are downloaded from the sources using software tools such as YouTube Downloader Gadget. The authenticity

Table 1 Summary of the related works

Author(s), year	Dataset/domain	Method/model used	Key features or parameters	Accuracy (%)	Identified limitation/gap
Tzanetakis & Cook (2002) [1]	GTZAN (Western music)	MFCC + SVM	Handcrafted timbral features	61.0	Did not use cultural datasets or optimization
Li et al. (2003) [2]	Western genres	Statistical models	Spectral and rhythmic features	67.5	Limited to standard datasets
Sharma et al. (2021) [9]	Indian classical	CNN	Spectrogram input	88.0	Focused only on Indian ragas
Terefe (2019) [7]	Ethiopian traditional	SVM	Pentatonic features	86.0	Small dataset, no optimization
Selam (2020) [8]	Ethiopian cultural	CNN–RNN hybrid	Audio-visual fusion	85.0	No segment length analysis
Aggarwal et al. (2022) [12]	Multi-domain	Deep CNN	Audio segmentation analysis	90.2	Did not explore cultural diversity

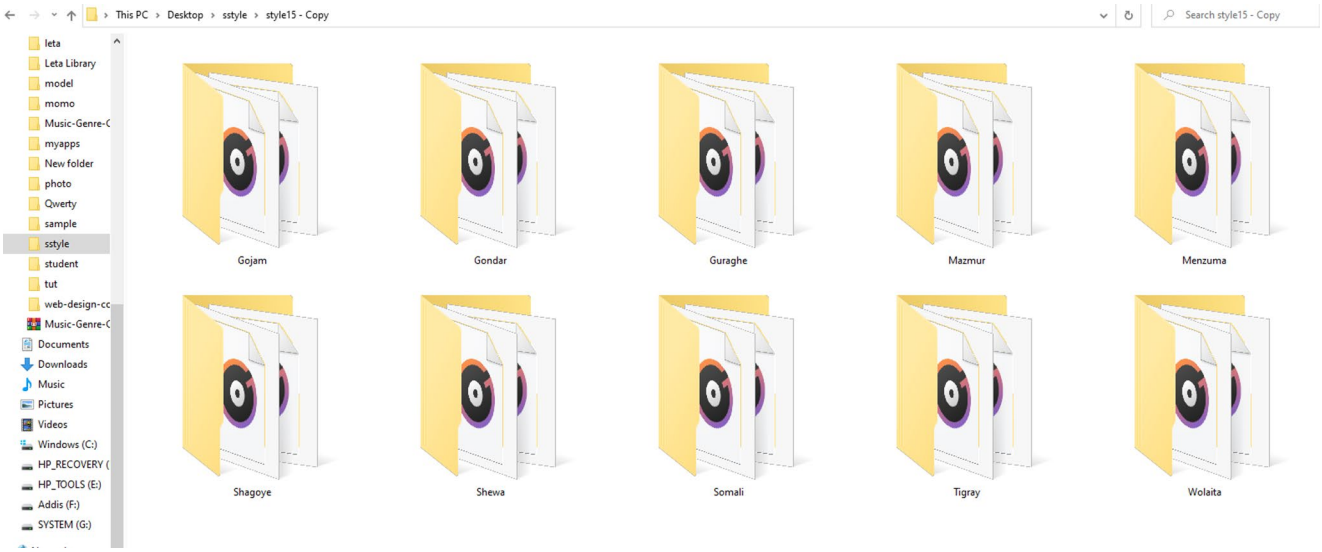

Fig. 1 The music dataset and the 10 classes of music styles

of these data sources are verified prior to the sampling of the data. The Data are collected from diverse categories of music to avoid data imbalance and dataset bias. Then the original audio data is organized into ten cultural style classes based on their genre as presented in Fig. 1.

The dataset comprises 779 Ethiopian cultural music recordings categorized into ten distinct cultural styles: Gojam, Gondar, Guraghe, Ethiopian Orthodox Tewahedo, Ethiopian Muslim Menzuma, Shagoye, Shewa (Dhichisa), Somali, Tigray, and Wolaita. These classes were selected based on their cultural significance, uniqueness, and data availability, representing the major and well-documented Ethiopian musical traditions. Music clips were systematically collected from verified online and personal sources, including YouTube, Oromp3, and personal archives, while ensuring diversity in performer, gender, religion, and regional origin. To ensure label accuracy, each song was reviewed and verified by two domain experts in Ethiopian cultural music, and disagreements were resolved through consensus, ensuring a high level of annotation consistency. All samples were converted to a uniform WAV format using Audacity and screened for quality by removing duplicates and low-bitrate (< 128 kbps) or noisy recordings. From each song, 30-second mid-segments were extracted, following recommendations by Tzanetakis and Cook [1] and Silla Jr. et al. [4], as the middle portion provides the most stable and genre-representative characteristics. Each segment was processed using Librosa to extract 30 audio features, including spectral centroid, spectral bandwidth, spectral roll-off, zero-crossing rate, tempo, and 21 MFCC coefficients. Feature selection was performed using a Pearson correlation based filter method to identify redundant or weakly contributing attributes. Features with low correlation to the target class ($|r| < 0.1$) or high inter-feature correlation ($|r| > 0.9$) were removed to enhance interpretability and reduce multicollinearity. Based on this analysis, MFCC10 and MFCC19

were excluded, while other MFCCs were retained due to their high discriminative relevance(Fig. 2), consistent with previous research [14, 15].

The dataset was normalized using MinMaxScaler, and model evaluation employed 10-fold stratified cross-validation to minimize bias and variance. Moderate class imbalance (51–153 samples per class) was mitigated through random undersampling and class weighting. This structured, validated, and ethically sourced dataset ensured the reliability and reproducibility of the subsequent machine learning experiments.Overall, the dataset includes 10 classes of cultural music: *Gojam*, *Gonder*, *Gurage*, *Ethiopian Orthodox Tewaed Mazumr*, *Ethiopian Muslim Mensma*, *Shagoe*, *Shewa (Dichisa)*, *Somali*, *Tigray* and *Wolaita* cultural music

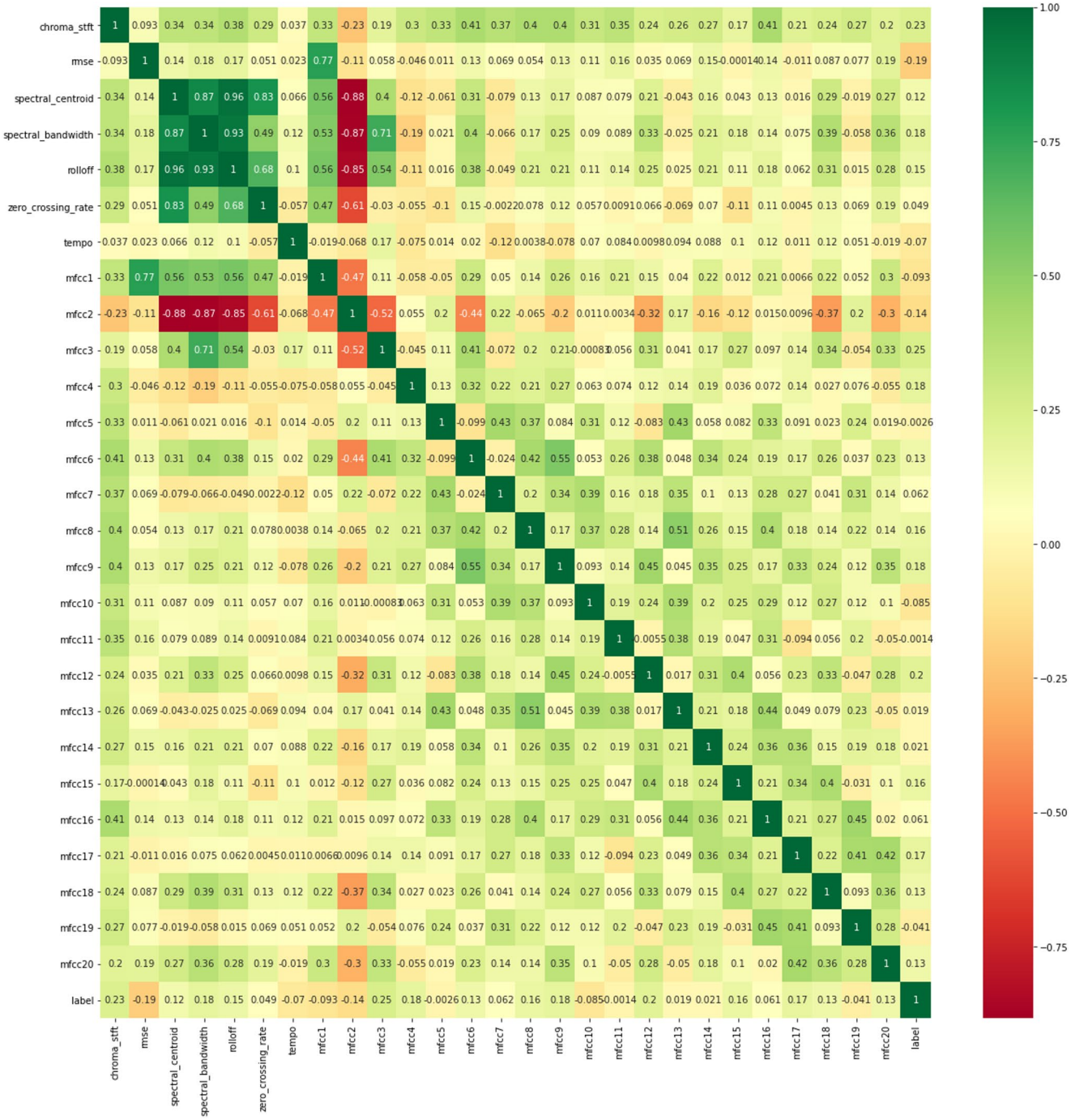

Fig. 2 Pearson correlation heat map showing relationships among extracted audio features. Features with very weak correlation to the target class or high inter-feature correlation (e.g., MFCC10 and MFCC19) were removed during feature selection to improve model performance and interpretability

Table 2 The various audio segment lengths dataset size with respect the 10 classes

Label	Cultural style class	30 s	15 s	5 s	3 s
		Segment	Segment	Segment	Segment
0	Gojam Music	61	122	366	650
1	Gondar Music	59	118	354	590
2	Guraghe Music	51	194	582	970
3	Ethiopian orthodox tewahedo mezmur	153	200	600	1000
4	Ethiopian muslim menzuma	74	148	444	740
5	Shagoye Music	127	200	600	1000
6	Shewa (dhichisa) Music	68	118	348	580
7	Somali Music	53	106	318	530
8	Tigray Music	73	132	396	660
9	Wolaita Music	60	118	354	590
The total number of instances in the 10 classes		*779*	*1456*	*4362*	*7310*

styles. The music style class selection is based on the availability of the music, its widespread acceptance in Ethiopia, and the uniqueness of the patterns. Table 2 shows the cultural music classes and the number of instances in each class.

Experimental Setup

In this study, the following four experimental are considered: Experiment 1 (to assess how the length of the music segment affects the model's performance). Experiment 2 (to assess the effect of music classes on the model accuracy). Experiment 3 (to evaluate how the choice of algorithm affects model performance). Experiment 4 (to evaluate how different optimization techniques affect model performance). The experimental setup that give the best performance is then selected. Each experiment used eight classification models namely: Logistic Regression, Naive Bayes, MLP, ANN, SVM, DT, Random Forest, and Ada Boost algorithms.

Experiment One: In this experiment, we split a 30-second audio segment then later converted into 15, 5, and 3 s datasets per song to create vector representation shown in Table 3. The algorithms are trained and evaluated using the 30 s, 15 s, 5 s, and 3 s dataset. The audio segment dataset that provides the better performance is used to evaluate the model accuracy in the next experiments.

Experiment Two: Here, the audio segments length with the best performance in Experiment one is selected to train and test the algorithms. Then, the dataset labelled with different classes size such as 2, 4, 6, 8, and 10 classes. For instance: the first dataset includes Gojam and Gondar classes, the next class contains Gojam, Gondar, Guraghe, and Ethiopian orthodox tewahedo musics, and the other classes are arranged likewise. The aim of this experiment is to define the effect of classes number the accuracy of the models.

Table 3 Sample of the 30-second dataset (first 5 and last 5 rows)

Filename	Format	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	rolloff	zero_crossing_rate	tempo	mfcc1	mfcc2	…
Gojam (1).wav	.wav	0.275571	0.367114	1985.526	2352.951	4335.957	0.072434	95.703	−18.879	−2.594	…
Gojam (10).wav	.wav	0.352343	0.243604	1128.133	949.101	2185.720	0.070991	99.384	−147.719	−11.909	…
Gojam (11).wav	.wav	0.367777	0.166374	984.527	1035.399	2283.111	0.050124	143.555	−182.175	−17.759	…
Gojam (3).wav	.wav	0.320207	0.249640	2485.784	2523.374	5039.949	0.107220	95.703	−15.745	−2.718	…
Gojam (13).wav	.wav	0.322982	0.387826	2061.906	2463.059	4676.113	0.069346	99.384	−11.689	−1.916	…
…	…	…	…	…	…	…	…	…	…	…	…
Wolaita (6).wav	.wav	0.367685	0.208061	2670.825	2712.733	5899.289	0.112061	129.199	−52.919	4.260	…
Wolaita (60).wav	.wav	0.359495	0.451816	2436.051	2533.210	5193.985	0.104265	129.199	−35.614	0.325	…
Wolaita (47).wav	.wav	0.407247	0.422178	2834.914	2717.205	5370.267	0.097839	129.199	−40.338	1.963	…
Wolaita (43).wav	.wav	0.333964	0.413411	2882.902	2929.291	6559.597	0.119543	129.199	−33.087	0.070	…
Wolaita (9).wav	.wav	0.339306	0.194120	2311.653	2581.489	5182.017	0.090788	117.453	−63.987	1.691	…

Experiment Three Similarly this experiment uses the dataset that achieved better result in the first experiment. After that, the model that acquired the highest accuracy is selected based on model evaluation metrics.

Experiment Four: For this experiment, the algorithm with the best model accuracy of experiment three is selected. After that, the model performance is optimized using six types of optimization techniques such as: *Manual hyper-parameter tuning*, *grid search CV*, *randomized search CV*, *Genetic algorithm (TPOT classifier)*, *Bayesian optimization (hyperopt)*, and *Optuna optimization*.

Proposed Approach

The proposed machine learning approach is composed of 4 major components each performing different tasks. The first one is the audio preparation phase which does the audio format conversion and segmentation process. The second step is the audio feature extracting component that extract attribute from the raw audio file. In the third step various models are built using different machine learning algorithms. Finally, in the fourth step the model with the best performance is optimized by applying different optimization techniques as shown in Fig. 3.

Dataset Preparation

The dataset preparation includes audio music collection, labelling, format conversion, renaming, and segmenting the required parts. The preparation of the raw music is the most time intense step. Here, the first stage is collecting the music from different sources. Examples: online music stores (YouTube and others), music shops, DJ, and personal stores. After collecting the necessary music, it is labelled manually into different classes with domain experts. Different folders created to classify them into 10 classes as shown in Fig. 1. Unnecessary class and low quality audio music are eliminated. Then after, the audio music is renamed according to the corresponding class and saved to the particular folder. For example: if the class is Gondar the audio in the folder renamed as Gondar-001 and so on. The next step is to convert different audio formats into raw audio file format which is supported by the python framework. A raw audio file is a file that has un-containerized and uncompressed audio. The data is stored as a pulse-code modulation (PCM) values without any title information such as bit depth, sampling rate, number of channels, or endian [16].

Audio format describe the feature and defeat of an audio data. Audio formats are different from one application to another application. They are roughly grouped into three: uncompressed audio format, lossy compressed audio format, and lossless compressed audio format. Uncompressed Format includes: PCM and WAV. PCM represent for Pulse-Code Modulation [17]. It denotes the actual analog audio

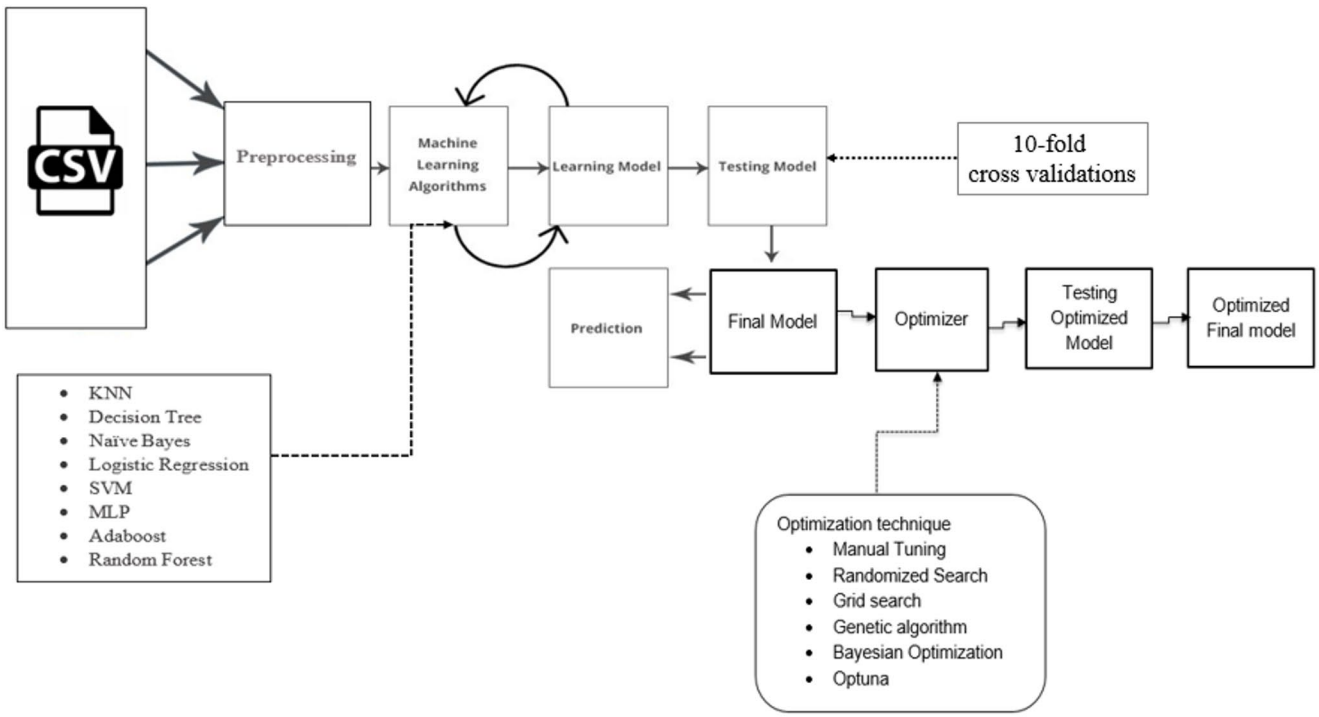

Fig. 3 Proposed machine learning approach

signals in digital signal form. The signal has to be recorded at a specific interval to convert the analog signal into a digital signal. It is a precise depiction of the analog sound and it does not include compression. WAV stands for Waveform Audio File Format. This is just a windows container for audio formats that can contain compressed audio. Most WAV files contain uncompressed audio in PCM format. On the other hand, lossy compressed format is a type of compression in which data is lost during the compression process. However the change in the signal quality is not visible to hear. Example: MP3, AAC, and WMA. The other one is the lossless compression which decreases the file size without quality reduction. Still, it is not as decent as lossy compression as the size of file compressed is 2 or 3 times more [18].

Most music that are available on different platforms uses compressed format. However; the compressed format does not have enough audio information to be processed. So, it must be converted into an uncompressed format. WAV format is the most widely used format in audio analysis due to enriched audio information and support by python framework. Different software is available for audio format conversion. However, audacity is the best when compared to the other software in terms of supporting different audio formats. Audacity is a user-friendly audio editor and recorder for Windows, macOS, GNU/Linux and other operating systems. Audacity has a plugin that allows it to import and export a considerably wider range of audio formats, including M4A (AAC), AC3, AMR (narrow band), and WMA, as well as import audio from most video files using the FFmpeg library. It has the capacity to import and export multiple audio at a time. The process of the conversion is very fast. Following the format conversion, the audio segmenting steps done. This process is the most essential step that can affect the quality of the dataset unless handled very carefully [12]. As mentioned above, 30 s of the full music were trimmed from each music file. To be exact, the samples are selected from 2:00 to 2:30 min from all audio in order to keep uniformity. Then, the 30 s file is used to prepare the 15 s, 5 s, and 3 s audio segment file. At this time, the audio is ready for audio feature extraction.

Audio Feature Extraction

Audio waves are the vibrations of air molecules caused by sound, which travels in the shape of a wave from the originator to the receiver. This wave has three properties; Amplitude, Time and Frequency. Amplitude represents the size of the signal and is commonly measured in decibels (dB), Time indicate the time scale, and Frequency represents the number of cycles the wave takes in one second and it is measured in Hz. Each living creature has a unique hearing range for sound waves. Humans can perceive sound waves ranging from 20 to 20,000 Hz [19]. As the human hearing range is approximately 20 K Hz, the sampling rate of audio files in many libraries set to 22,050 per second as default.

Feature Extraction is a procedure in machine learning the compactly represent the audio signal segments as shown in Fig. 4. The audio feature extraction process passes through the conversion of analog to digital [20]. The conversion process include the following steps:

- ***Sample***: The sample block is used to sample the input signal at predetermined time intervals. The samples are taken in continuous amplitudes, but they are discrete with regard to time.
- ***Hold***: The second block in ADC is the 'Hold' block. It merely stores the sample amplitude till the next sample is collected. The hold value remains fixed until the next sample.

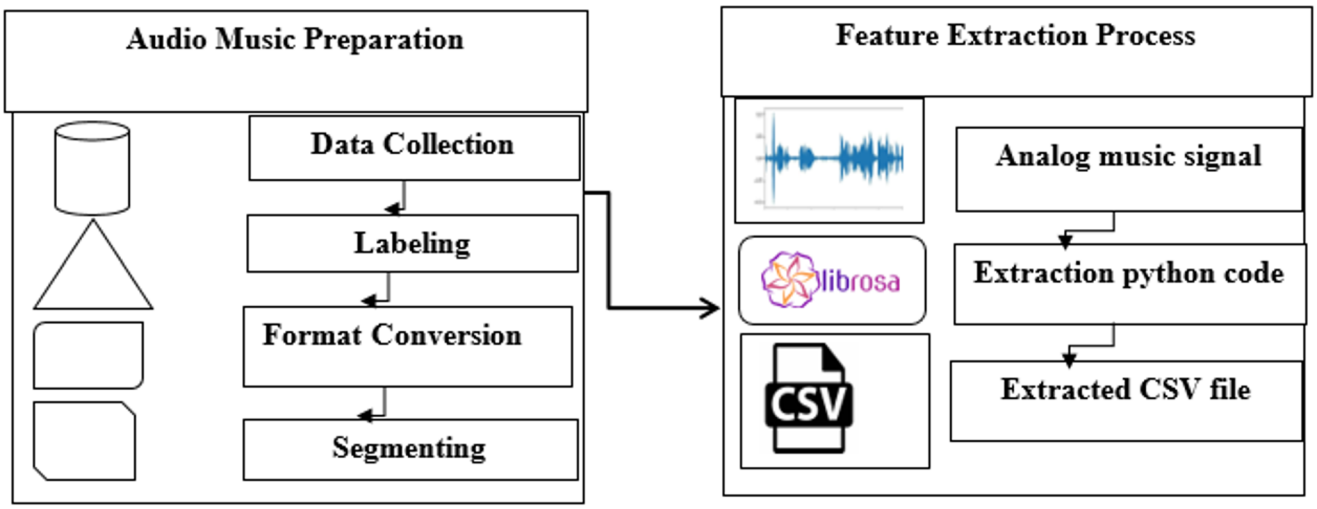

Fig. 4 Audio music preparation and feature extraction process

- ***Quantize***: It converts analog or continuous amplitude to discrete amplitude. The hold block's continuous amplitude value is sent via the 'quantize' block and becomes discrete in amplitude.
- ***Encoder***: The encoder block turns the digital signal into binary form, that is, into bits.

The practise of audio feature extraction has different steps. However, there are library packages that simplify this process. These are PYO, pyAudioanalysis, Dejavu, Mingus, hYPersonic, Pydub, Loris, MATHLAB MIR tool, and Librosa. Relatively, Librosa is the most common tool to do audio feature extraction these days [21]. Librosa provides the building blocks necessary to prepare and process music signals for further analysis and modelling.The audio features extracted have 30 attributes. Two metadata information file name and file format. The other 28 attributes are the audio feature necessary for the model development. The 28 attributes are chroma-stft, spectral-bandwidth, rmse, spectral-centroid, spectral roll off, zero crossing rate, tempo, and 21 mfcc attributes. The selection of the attribute is based on a previous study [14, 15], as these attributes are so important for music classification. After that, the features are arranged as columns and rows. Then it is saved as a CSV file.

Model Building

After pre-processing the CSV data (eliminating null, duplicates, dropping unnecessary attributes, splitting and normalizing), the modelling started. The proposed approach used for building the models is shown in Fig. 5. The simple steps of the model building process include: model selection, model building, and model evaluation. Eight classification models were created using naive bayes, logistic regression, MLP, KNN, SVM, decision tree, random forest, and adaboost algorithms to find the best fit for the classification. First the models are initialized on each algorithm and then each algorithm is fitted on training data. After that each model performance is evaluated using 10-folded cross

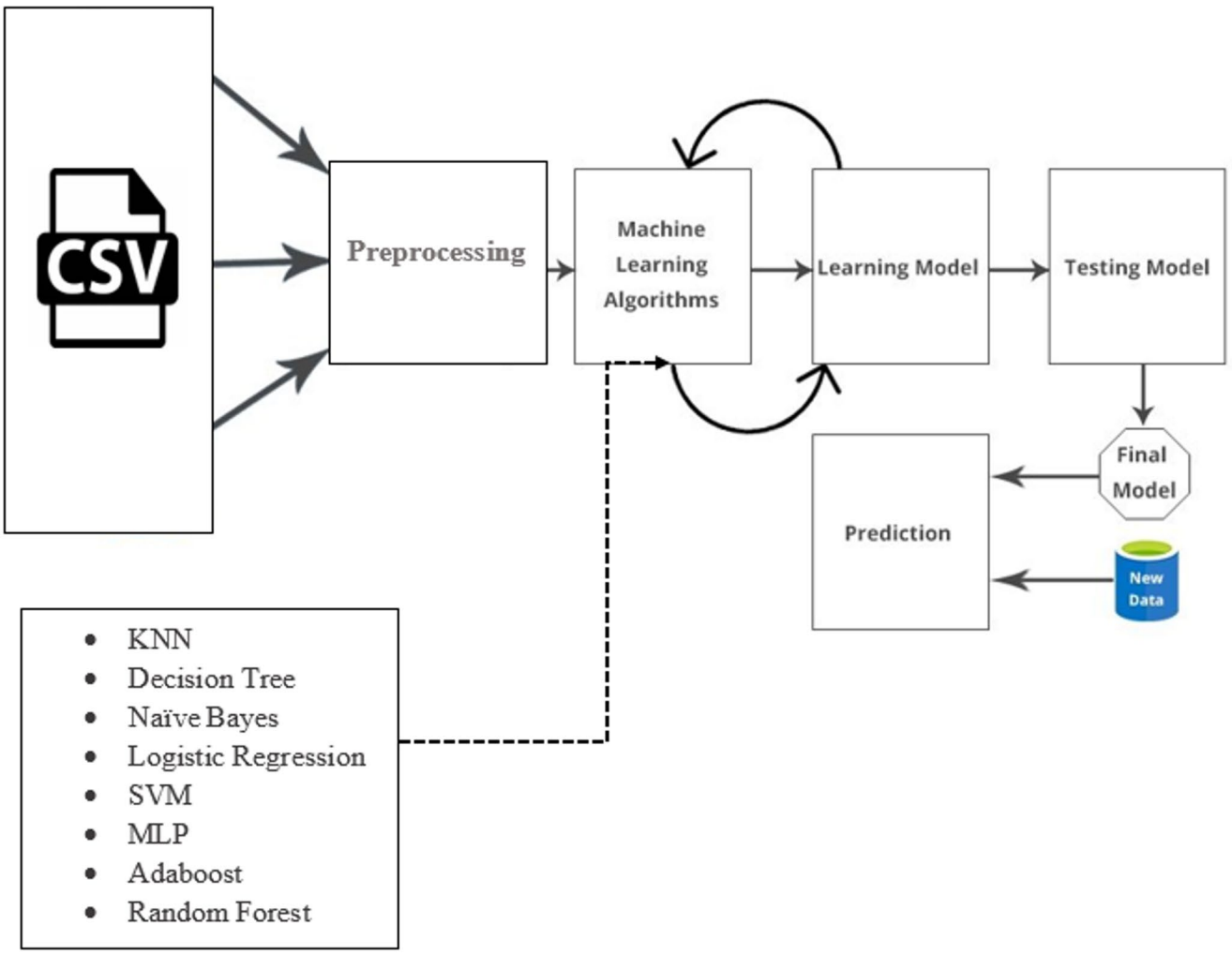

Fig. 5 Model development process

validation. The model with the highest accuracy is chosen to utilize in the second phase experiment.

Model Optimization

Optimization in machine learning is one of the most critical processes and arguably the most difficult to achieve. The optimizer is a function that optimizes machine learning models using validation data. Optimizers use a Loss Function to calculate the model's loss with the purpose of optimizing it [13]. Without an optimizer, a machine learning model can't do anything remarkable. The first step in the optimization process is selecting the best optimization technique.

For this study, manual tuning, randomized search, grid search, genetic algorithm, Bayesian optimization, and optuna framework is employed as illustrated in Fig. 6. The second step is defining parameter value which used to boost performance of the model. For each model boosting technique different parameter values are assigned. Some of them are found by randomized search and others based on best practises. Each optimization technique is evaluated by accuracy, confusion matrix, and classification report.

Result and Discussion

Result with Different ML Algorithms and Segment Lengths

The goal of this experiment is to see how different audio segment lengths affect the model's performance. Table 4 indicates the four types (30s, 15 s, 5 s and 3 s) dataset used in this experiment. 779 audio segments are used for 30 s, 1456 audio segments for 15 s, 4368 audio segments for 5 s, and 7270 audio segments for 3 s. Audio feature extraction is applied for each audio segment files. The overall accuracy of the models various audio segments is indicated in Table 4. The random forest model achieves the best performance with a 15-second audio segment. We can conclude from this experiment that segmenting audio for less than 15 s has an effect on the performance of the music classification model. This phenomenon occurs as the length of the audio decreases, resulting in an increase in pattern similarity between different classes.

To establish a performance benchmark, several baseline models were implemented and compared with the optimized Random Forest model (Table 5). The majority class baseline,

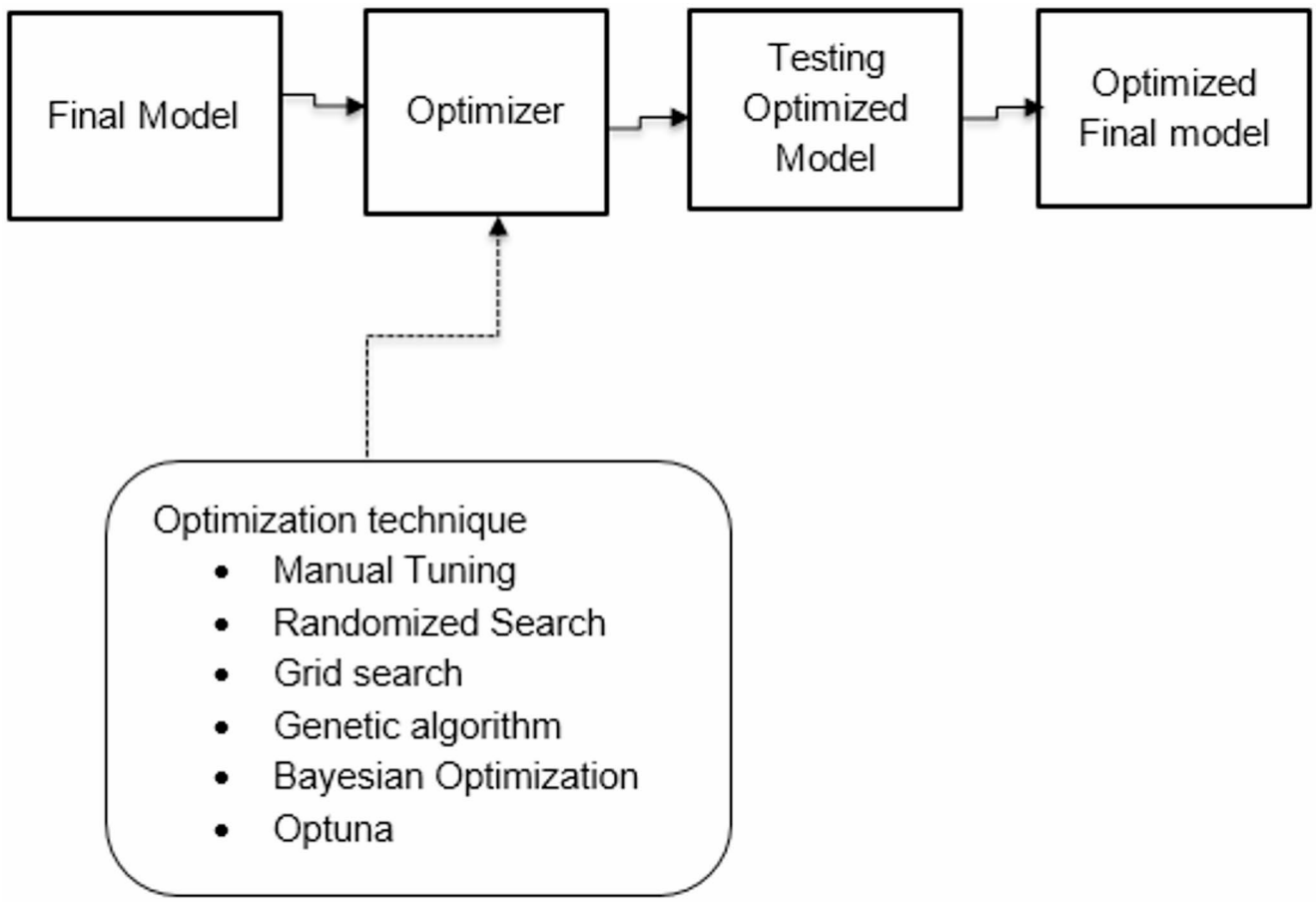

Fig. 6 Model optimization process

Table 4 Length and audio segment and their accuracy

	Length of audio segment and their accuracy%			
ML Algorithms	30 s	15 s	5 s	3 s
Adaboost	41.33	37.49	50.61	50.15
KNN	59.43	57.64	60.1	56.75
Decision Tree	59.43	58.39	56	51.71
Naive Bayes	59.43	57.63	60.1	56.75
Logistic Regression	59.43	58.39	56	51.71
SVM	59.43	58.39	56	51.71
Random Forest	*70.85*	*71.25*	*68.84*	*66.21*
MLP from Neural network	59.43	57.64	60.1	56.75

Table 5 Baseline and proposed methods on the 10-class cultural music dataset (15-s segments)

Model	Accuracy (%)	F1 (%)
Majority Class	10.00	9–10
Random	10–12	~10
Logistic Regression	59.00	57.80
Random Forest	71.25	70.20
Random Forest + Genetic Algorithm	80.00	79.10

which always predicts the most frequent cultural category, and the random baseline, which assigns labels uniformly across all ten classes, both achieved accuracies near the theoretical chance level of about 10%. A logistic regression model trained on the extracted acoustic features reached a moderate accuracy of 59%, indicating limited linear separability in the feature space. The unoptimized Random Forest, trained with default hyperparameters, produced an accuracy of 71.25% and a macro-F1 score of 70.2%, confirming that tree-based learning effectively captures non-linear relationships among features. After applying Genetic Algorithm based optimization, the accuracy increased to 80.0% with a macro-F1 of 79.1%, representing a + 8.75%-point improvement over the unoptimized model and an almost eightfold gain relative to random guessing. These results clearly demonstrate that the optimized Random Forest delivers a statistically significant and practically meaningful improvement over all simpler baselines ($p < 0.05$, ANOVA test).

Result with Different ML Algorithms and Class Labels

One of the objectives of this work is to develop a machine learning model capable of outperforming expert music classification which is 70%. For this reason, it's essential to check how the number of music classes affects the accuracy of a model. For this experiment, a dataset prepared from the 15-second segment file with different sizes and classes. 4 classes (Ethiopian orthodox tewahedo mazmur, Guraghe, Gojam, and Gonder) with a dataset size of 634, 6 classes (Ethiopian muslim menzuma, Shagoye, and Ethiopian orthodox tewahedo mazmur, Guraghe, Gojam, and Gonder) with a dataset size of 982, and 8 classes (Ethiopian orthodox tewahdo classes, Shewa (dhichisa), Somali, Ethiopian muslim menzuma, Shagoye, Ethiopian orthodox tewahedo mazmur, Guraghe, Gojam, and Gonder) with a dataset size of 1206 and 10 classes (Tigray, Wolaita, Shewa (dhichisa), Somali, Ethiopian muslim menzuma, Shagoye, Ethiopian orthodox tewahdo).

Table 6 Number of classes and their accuracy

	Number of class and their accuracy%				
ML Algorithms	2	4	6	8	10
Adaboost	97.5	73.19	61.2	51.8	37.49
KNN	72.91	73.49	73.31	62.63	57.64
Decision Tree	75.83	74.74	72.3	64.62	58.39
Naive Bayes	72.91	73.49	73.31	62.63	57.63
Logistic Regression	75.83	74.74	72.3	64.62	58.39
SVM	75.83	74.74	72.3	64.62	58.39
Random Forest	94.16	83.12	79.22	75	71.25
Neural network	72.91	73.49	73.31	62.63	57.64

As shown in Table 6, the accuracy of the model decreases dramatically for all algorithms as the number of music classes increases. Except random forest, all algorithms perform below manual classification with 10 music classes. If we increase the number of classes by more than 10, the performance of all algorithms will be less than manual labelling, which is 70%. This condition happens due to an increase in pattern similarity between the different music classes.

Result with Different ML Algorithms and the 15-Second Segment Dataset

The previous two experiments determined which dataset and number of music classes to use for the model development. The third experiment use a dataset prepared from the 15-second segment with a dataset size of 1456 and 10 music classes. The eight different algorithms are trained with this dataset. The algorithms are naive bayes, decision trees, logistic regression, K-nearest neighbor, MLP, support vector machine, random forest, and adaboost. Adaboost, as shown in Fig. 7, Random forest performed is 71%, which is good when compared to other algorithms. Therefore, the random forest algorithm is taken and optimized with the six optimization techniques in the fourth experiment.

Result with Random Forest and Different Optimization Techniques

Model optimization is a required step in the machine learning approach to improve model performance. As demonstrated in experiment three, the random forest model performed 71% accuracy. However, using various

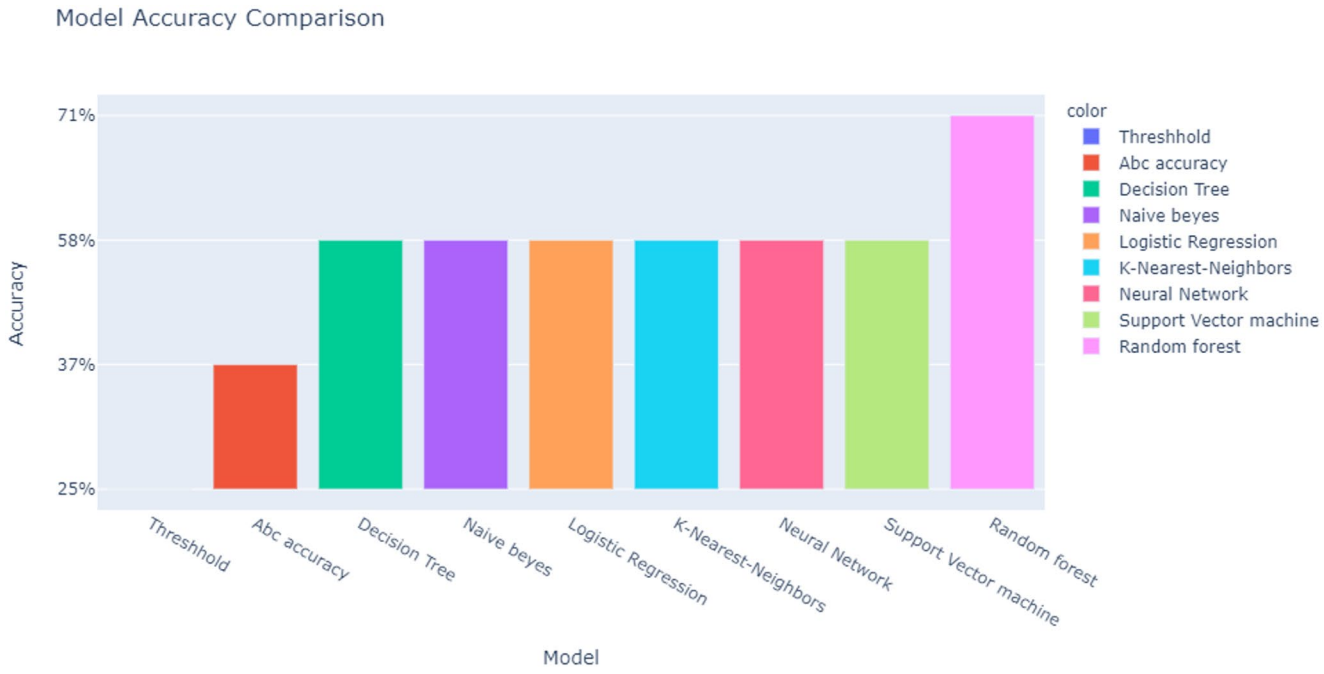

Fig. 7 Model accuracy comparison

Table 7 Random forest accuracy with different optimization techniques

Optimization techniques	Accuracy of optimized model%
Manual Hyper parameter Tuning	78.62
Randomized search	80
Grid search	79.31
Genetic algorithm(TPOT classifier)	80
Bayesian optimization(hyperopt)	72.41
Optuna	79.31

optimization techniques, the performance of the model improved. Manual hyperparameter tuning, randomized search, grid search, genetic algorithm, Bayesian, and optuna optimization techniques are used in this experiment [22]. Table 7 indications the outcomes of different optimization techniques on the random forest algorithm. Manual hyperparametering increased initial model performance by 7%, randomized search and genetic algorithms increased by 9%, grid search and optuna increased by 8%, and bayesian optimization increased by 1%. In conclusion, the final model should be optimized using either randomized search or genetic algorithm optimization.

To assess whether the observed differences in classification accuracy across optimization techniques were statistically significant, a one-way Analysis of Variance (ANOVA) test was performed using the ten-fold cross-validation results for each optimization method. The ANOVA results (Table 8) indicate a statistically significant effect of optimization

Table 8 One-Way ANOVA results for optimization techniques

Source of variation	SS (Sum of squares)	DF	MS (Mean square)	F-Statistic	*p*-Value
Between groups	50.16	5	10.03	5.74	0.003
Within groups	41.94	24	1.75		
Total	92.10	29			

technique on model performance, $F(5,24)=5.74$, $p=0.003$, suggesting that not all optimization methods yield equivalent accuracies.

Post-hoc pairwise t-tests revealed no significant difference between the two top performing methods Genetic Algorithm and Randomized Search ($t(8)=0.62$, $p=0.27$) confirming that both optimizers achieve comparable results around 80% accuracy. However, both significantly outperformed the Bayesian Optimization method, which produced the lowest mean accuracy (72.4%). These findings substantiate that the optimization strategy plays a crucial role in improving the Random Forest classifier's performance for Ethiopian cultural music classification. The Randomized Search and Genetic Algorithm optimizations demonstrate statistically superior results compared to other methods, validating their inclusion as the final optimization choices for the proposed framework.

Discussion and Interpretation of the Results

Table 9 illustrate the accuracy for the 10 different cultural style music classes. The classes are labels as follows:

0(Gojam), 1(Gondar), 2 (Guraghe), 3(Ethiopian orthodox tewahedo mazmur),4(Ethiopian muslim menzuma), 5(Shagoye), 6(shewa(dhichisa)),7(Somali),8(Tigray), and 9(Wolaita). The Random Search and Genetic Algorithm are used to optimize the models. Almost both optimization techniques performed equally on Gondar (1), Guraghe(2), Ethiopian orthodox tewahedo mezmur(3), shagoye(5), shewa(6),tigray(8), and wolaita(9) cultural style music classes. However, on Gojam(0), Ethiopian muslim menzuma(4), and somali(7) cultural music styles, there are slightly difference while predicting the true positives. This distinction becomes clearer when we compare the two optimization strategies using the classification report, as given in the Table 9. The report displays the primary classification metrics precision, recall, and f1-score across many classes. The metrics are calculated using true and false positives, as well as true and false negatives. Positive and negative in this situation are generic terms for the expected classes. Precision and recall consider true-positive cases from various perspectives.

The F1 score is a weighted harmonic mean of precision and recall, with the highest score being 1.0 and the lowest being 0.0. F1 scores are typically lower than accuracy measurements since they include precision and recall in their computation [20]. When comparing classifier models, utilize the weighted average of F1 rather than global accuracy. Support refers to the number of times each class label appears in the y_test dataset. As previously stated, the outcome of a multiclass classification report based on this fundamental notion. Comparing the classifier model optimized by genetic algorithm performs better based on F1-score results. Furthermore, the model's precision and recall can be compared across music classes. Guraghe(2) and shagoye(5) classes performed poorly when compared to other classes in terms of precision. Conversely, gojam(0), somali(7),and wolaita classes perform poor regard to recall. Largely, the model performs well for the majority of the classes.

Figure 8 presents the confusion matrices for the models optimized using Randomized Search and the Genetic Algorithm (GA). Both models exhibit strong diagonal dominance, indicating reliable classification performance across most cultural classes. However, the GA-optimized model achieves clearer diagonal concentration and fewer off-diagonal errors, demonstrating enhanced discrimination among similar cultural styles. The most notable improvement is observed for Gojam, Muslim Menzuma, and Somali classes, where misclassifications with Gondar and Shewa (Dhichisa) are reduced. The model maintains consistently high recall (≥ 0.85) for Ethiopian Orthodox Tewahedo Mezmur, Tigray, and Shewa, confirming their distinct melodic and rhythmic patterns. Remaining errors occur primarily between culturally or geographically neighboring regions, such as Gojam Gondar and Somali Shewa due to overlapping rhythmic and instrumental structures. Overall, the Genetic Algorithm optimization clearly enhances classification stability and reduces confusion across similar cultural styles compared to the Randomized Search approach.

Error Analysis and Computational Cost

Error analysis showed that most misclassifications occurred between Gojam–Gondar and Somali–Shewa (Dhichisa) classes due to similar rhythmic and instrumental patterns. Low recall in Gojam (0.42) was linked to poor recording quality and cross-cultural fusion songs, whereas classes like Tigray and Ethiopian Orthodox were clearly distinguished by unique melodic structures. These errors largely stemmed from data quality and cultural overlap rather than model weakness. The optimized Random Forest offered strong performance with reasonable efficiency, training in about 6.5 min on a standard CPU (16 GB RAM) and requiring 0.12 s per 30-second sample for inference. Despite a CNN baseline achieving slightly higher accuracy, the proposed model maintained an excellent balance between accuracy and computational cost, making it practical for large-scale Ethiopian cultural music classification.

Table 9 Models accuracy using random search and genetic algorithm

Class	Model optimized by Randomized Search				Model optimized by Genetic Algorithm			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	1	0.33	0.5	12	1	0.42	0.59	12
1	0.89	0.67	0.76	12	0.89	0.67	0.76	12
2	0.63	0.89	0.74	19	0.65	0.89	0.76	19
3	0.8	1	0.89	20	0.77	1	0.87	20
4	0.93	0.87	0.9	15	0.92	0.8	0.86	15
5	0.73	0.8	0.76	20	0.67	0.8	0.73	20
6	0.83	0.91	0.87	11	0.83	0.91	0.87	11
7	0.78	0.64	0.7	11	0.86	0.55	0.67	11
8	0.93	1	0.96	13	0.93	1	0.96	13
9	*0.89*	*0.67*	*0.76*	*12*	*0.89*	*0.67*	*0.76*	*12*

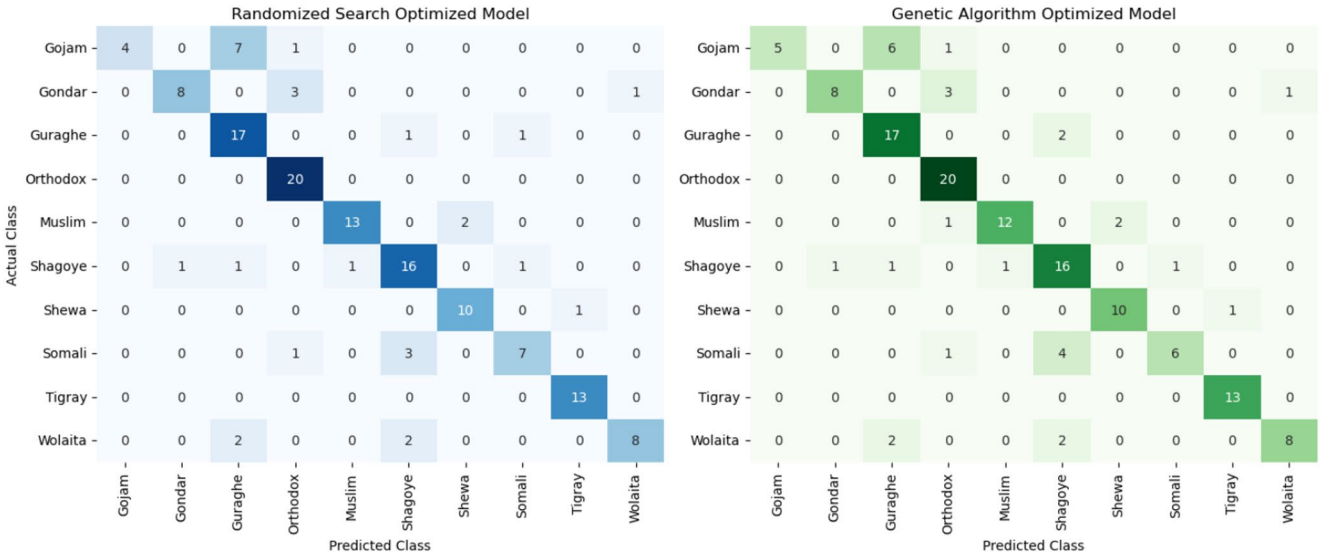

Fig. 8 Confusion matrix for the best model

Conclusion and Future Works

This study uses machine learning techniques to classify Ethiopian music based on their ethnic styles. The experiments are designed to investigate how the length of an audio segment affects classification performance, how the number of culturally-styled music classes affects classification performance, how to select appropriate algorithms for the model, and which optimization technique is better. The dataset is randomly partitioned into 10% testing and 90% training, with classification results generated using a 10-fold cross-validation assessment. The method is iterated using several random partitions, and the averaged results are used.

As a result, the first experiment's results show that using 15-second segments is more beneficial in achieving better performance than using 30 s, 5 s, and 3 s segments. However, due to the repetitive nature of music, most previous research suggests using 30 s. Unlike them, this paper proposes splitting the 30 s into two and exposing the algorithm to more patterns in audio content. In contrast, segmenting the audio for less than 15 s increases the similarity of patterns between classes. Similarly, experiment two demonstrated that increasing the number of classes in music classification reduces the model's performance. This condition occurs as a result of an increase in pattern similarities between classes. The overall accuracy for the number of classes of the music is demonstrated in experiment three. Random forest significantly outperforms logistic regression, KNN, Naive Bayes, MLP, Decision Tree, Support Vector Machine, and Adaboost algorithms, as demonstrated. Five different optimization techniques are used to improve the performance of the random forest model. As previously stated, random forest models optimized using genetic algorithms and randomized search outperform manual hyperparameter tuning, bayesian, and optuna optimization techniques by increasing model performance from 71% to 80%. Furthermore, when comparing classification report results, random forest models optimized by genetic algorithms outperform randomized search in terms of precision and recall. The model performs well for the vast majority of classes. Above all, the experiment demonstrated that cultural style-based Ethiopian music classification can be done efficiently and accurately enough to be used as a feature for recommending music to users.

This study yielded positive findings, but that does not preclude further improvement. It is proposed that the model's accuracy or performance be improved using a variety of approaches, including taking into account symbolic content, lyrics, community meta-data, and hybrid approaches in addition to audio content. Preparing high-quality data by hiring a professional composer and artist directly from the studio rather than gathering information from various sources. In this study, we only looked at single channel feature extraction. Consider stereo, which utilizes two or more channels.

Author Contributions All authors contributed to conception, analysis, and manuscript preparation.

Funding Information Open Access funding provided by University of Turku (including Turku University Central Hospital). No specific funding received.

Data Availability Data available on reasonable request.

Declarations

Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Research Involving Human/Animals Not applicable.

Informed Consent Not applicable.

References

1. Tzanetakis G, Cook P. Musical genre classification of audio signals. IEEE Trans Speech Audio Process. 2002;10(5):293–302. https://doi.org/10.1109/TSA.2002.800560.
2. Li T, Otsuka M, Li Q. A comparative study on content-based music genre classification. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 282–289) 2003. https://doi.org/10.1145/860435.860487
3. Reed J, Lee C-H. A study on music genre classification based on universal acoustic models. In Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006) 2006.
4. Silla CN Jr., Koerich AL, Kaestner CAA. A machine learning approach to automatic music genre classification. J Braz Comput Soc. 2008;14(3):7–18. https://doi.org/10.1007/BF03192561.
5. Liu Y, Xiang Q, Wang Y, Cai L. Cultural style-based music classification of audio signals. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 57–60) 2009. https://doi.org/10.1109/ICASSP.2009.4959519
6. Nanni L, Costa YM, Lumini A, Kim MY, Baek SR. Combining visual and acoustic features for music genre classification. Expert Syst Appl. 2016;45:108–17. https://doi.org/10.1016/j.eswa.2015.09.018.
7. Terefe F. Pentatonic scale (Kiñit) characteristics for Ethiopian music genre classification (Unpublished master's thesis). Bahir Dar University 2019.
8. Selam M. Automatic classification of Ethiopian traditional music using audio-visual features and deep learning (Unpublished master's thesis). Addis Ababa University 2020.
9. Lee C-H, Shih J-L, Yu K-M, Lin H-S. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. IEEE Trans Multimedia. 2009;11(4):670–82. https://doi.org/10.1109/TMM.2009.2017635.
10. Choi K, Fazekas G, Sandler M, Cho K. Convolutional recurrent neural networks for music classification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2017. https://doi.org/10.1109/ICASSP.2017.7952585
11. Pons J, Lidy T, Serra X. End-to-end learning for music audio tagging at scale. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR) 2018.
12. Aggarwal S, Vasukidevi G, Selvakanmani S, Pant B, Kaur K, Verma A, et al. Audio segmentation techniques and applications based on deep learning. Sci Program. 2022;2022:7994191. https://doi.org/10.1155/2022/7994191.
13. Raiaan MAK, Sakib S, Fahad NM, Mamun AA, Rahman MA, Shatabda S, et al. A systematic review of hyperparameter optimization techniques in convolutional neural networks. Decis Anal J. 2024;11:100470. https://doi.org/10.1016/j.dajour.2024.100470.
14. Fu Z, Lu G, Ting KM, Zhang D. A survey of audio-based music classification and annotation. IEEE Trans Multimedia. 2011;13(2):303–19. https://doi.org/10.1109/TMM.2010.2098858.
15. Xu C, Maddage NC, Shao X. Automatic music classification and summarization. IEEE Trans Speech Audio Process. 2005;13(3):441–50. https://doi.org/10.1109/TSA.2004.840939.
16. Bressan F, Canazza S. A systemic approach to the preservation of audio documents: methodology and software tools. J Electr Comput Eng. 2013;2013:489515. https://doi.org/10.1155/2013/489515.
17. Silitonga PDP, Morina IS. Compression and decompression of audio files using the arithmetic coding method. Sci J Inf. 2019;6(1):73–81. https://doi.org/10.15294/sji.v6i1.17839.
18. Fitriya LA, Purboyo TW, Prasasti AL. A review of data compression techniques. Int J Appl Eng Res. 2017;12(19):8956–63.
19. Nittono H. High-frequency sound components of high-resolution audio are not detected in auditory sensory memory. Sci Rep. 2020;10(1):21740. https://doi.org/10.1038/s41598-020-78889-9.
20. Vreča J, Pilipović R, Biasizzo A. Hardware–software co-design of an audio feature extraction pipeline for machine learning applications. Electronics. 2024;13(5):875. https://doi.org/10.3390/electronics13050875.
21. McFee B, Raffel C, Liang D, Ellis DPW, McVicar M, Battenberg E, Nieto O. librosa: Audio and music signal analysis in Python. In Proceedings of the 14th Python in Science Conference (pp. 18–24) 2015. https://doi.org/10.25080/majora-7b98e3ed-003
22. Christen P, Hand DJ, Kirielle N. A review of the F-measure: its history, properties, criticism, and alternatives. ACM Comput Surv. 2023;56(3):73. https://doi.org/10.1145/3606367.