# Novel hybrid model for music genre classification based on support vector machine

**Conference Paper** · April 2018

DOI: 10.1109/ISCAIE.2018.8405505



3 authors:

Srishti Sharma
Delhi Technological University


Prasenjeet Fulzele


Indu Sreedevi
Delhi Technological University

# Novel Hybrid Model for Music Genre Classification based on Support Vector Machine

Srishti Sharma[1], Prasenjeet Fulzele[2]
[1]Dept. of Electrical Engineering,
[2]Dept. of Computer Science,
[1]Delhi Technological University, New Delhi, India
[2]Jaypee Institute of Information Technology,
Uttar Pradesh, India
[1]srishtish21@gmail.com, [2]prasenjeet.fulzele7@gmail.com

Indu Sreedevi
Dept. of Electronics & Communication Engineering
Delhi Technological University
New Delhi, India
s.indu@dce.ac.in

***Abstract*— Automatic Music genre classification using machine learning techniques has gathered momentum in the recent past and continues to govern this area of research with promising results, saving time and human efforts. Machine Learning techniques have successfully dominated the methods used for the automation of this classification. In this paper, a novel method of music genre classification is proposed based on stacking of Support Vector Machine (SVM) with Relevance Vector Machine (RVM) and Decision Trees. The model uses the acoustic properties of the audio files as their features for classification. The three models are trained as Error-Correcting Output Code Classifiers and they individually classify the audio files with certain posterior probabilities. The results from the three classifiers are fused using the sum rule to evaluate the final outcome of the combined model. The performance of this hybrid model is evaluated on the GTZAN music dataset and is compared with individual performances of the models used in the combination. The proposed combined model outperformed the other models with an accuracy of 87% confirming the efficient utilization of the advantages of individual models.**

***Keywords—Music Genre Classification, Machine Learning, Support Vector Machine, Relevance Vector Machine, Decision Trees, Error-Correcting Output Code***

## I. INTRODUCTION

With the music industry taking a massive hit and the advent of technology that can perform tasks with a few taps on the screen, both have boosted the multimedia experience of a user, enabling users to have access to enormous audio files, music playlists and libraries making it a tiresome task to keep their music library sorted and arranged genre wise. Going through each song and audio file and to organize the collection effectively while listening to each track and then label them under their respective genres is a daunting challenge. Dividing the song collection based on their genre is the most common and practical approach when it comes to managing such huge entries of audio files. This classifier can be a useful feature in digital sound mixing platforms and in industry standard music production tools to enable users to keep their music tracks labeled in their respective genres. The statistical properties related to instrumentation and rhythmic structure of the audio signals can be used to characterize musical genres [1].

Audio signal discrimination into music, speech, silence and environment sound has been an active topic of research [2]. Various classification methods based on machine learning techniques have been proposed gaining a significant amount of acceptance in this area of study. Along with other machine learning methods employed for music genre classification, various neural network models have also been used successfully to differentiate between the classes of music. Convolutional Neural Network has been used to classify the genres based on the visual representation of signals [3]. Another approach to this problem was to exploit the property of the data to be time series. LSTM-RNN model was used to cater to this factor of the data [4]. The usage of Deep Neural Networks (DNN) helps in training of huge amounts of data [5-8]. Development of DNN has led to its active application in this field [9-10]. Other techniques used include k-Nearest Neighbour [11] which was outperformed by SVM [12].

In this paper a hybrid model of SVM, RVM and an ensemble classifier of decision trees is used to classify music genres. It is concluded that the hybrid model outperforms the individual operation of three models used with the maximum accuracy of 87%. In addition to that it is observed that the RVM and the ensemble classifier have their accuracy enhanced when stacked over an SVM classifier from 68% to 85% and from 79% to 85% respectively. This stacked model is used in the hybrid structure of the three models.

This paper has been precisely divided into six sections. The features that are extracted from audio files are described in Section II. The models used in this study are briefly described in Section III along with their usage in the proposed model. Section IV comprises of the implementation of the proposed model including the training method used. The performances of the models SVM, RVM stacked over SVM, ensemble of decision trees stacked over SVM and the hybrid model of these three, are compared in Section V along with the confusion matrix and the ROC curve to show the individual accuracies achieved for each genre. Finally the paper is concluded in Section VI.

## II. FEATURE EXTRACTION

The most essential task while building an audio signal classifier is the extraction and selection of appropriate features. The acoustic features of audio signals are widely classified into two categories; *Perceptual*, based on the manner in which the audio is heard by humans like Rhythm, Pitch and Timbre and *Physical,* based on statistical and mathematical properties of signals such as Zero Crossing Rate (ZCR), energy and frequency [13]. Some of these features, despite of belonging to different categories, are interdependent like frequency and pitch. In order to build a strong classifier features of both categories were extracted and used.

To evaluate the performance of the model the GTZAN music database was taken which is provided by the MARSYAS open source software framework. Certain features were extracted from the audio signals that were considered to classify the music genres and were used to build the final feature vector which was eventually given as input to the classifier. Features from both categories were taken into consideration.

### *A. PERCEPTUAL FEATURES.*

#### *1) Mel Frequency Cepstral Coefficients (MFCC)*

This feature has been widely used for automatic speech and sound recognition. Some part of human speech perception and production is imitated by MFCC in order to extract a feature vector which contains all the information of the linguistic message. The logarithmic representation of pitch and loudness of human auditory system is also impersonated by MFCC. Mel frequency appreciably distinguishes between various sound sources and sound instruments [14]. To calculate the MFCCs of a signal, it is divided into several short frames to keep the signal constant. Then the periodogram estimates of power spectrum are calculated for all frames to identify the frequencies present in the frames. Then the Mel-spaced filter bank is computed and the energy in each filter is summed up. The amount of energy in various frequency regions is now known. To have the features closer to what humans hear, logarithm of these energies is calculated.

$$M(f) = 2595\, log_{10}\left(1 + \frac{f}{700}\right) = \ln\left(1 + \frac{f}{700}\right) \quad (1)$$

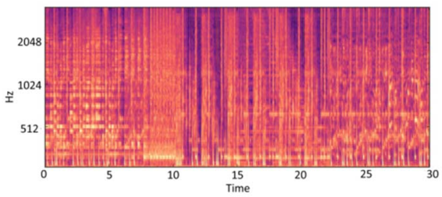

Fig. 1. Mel-Spectrogram

#### *2) Statistical Spectrum Descriptors (SSD)*

The Statistical Spectrum Descriptors (SSD) are vital for computing statistical moments on critical bands generated using the Bark Scale spectrogram as well as in outlining the rhythmic energy variations observed in these critical regions. These seven statistical moments as mentioned are computed: mean, median, variance, skewness, kurtosis, min and max values for every single critical band as grouped in the Bark scale spectrogram making up the SSD feature set. The Bark scale is converted into decibel scale. These values are further altered to produce equivalent Sone values before these moments are finally calculated on the resulting spectrogram after the transformations. The final SSD feature set of an audio file is the median of all the extracted segments of the given audio signal.

#### *3) Rhythm histogram*

The rhythm histogram features proves to be very useful in terms of gathering rhythmical characteristics of an audio file. In contrast to features like SSD, the processing is not done on every single critical band, rather a histogram is formed comprising of 60 bins by summing the modulation frequencies for all 24 critical bands with frequencies ranging from 0.168Hz to 10Hz. Similar to the SSD, the RH feature set is also the median of respective histograms of all the processed segments.

### *B. PHYSICAL FEATURES*

#### *1) Spectral Centroid*

The spectrum of any audio signal is indicative of how the frequencies are distributed within the captured signal. The weighted mean of all these frequencies is the spectral centroid of that particular signal. Different music genres and sub genres usually have characteristic spectral centroids which can prove to be a crucial factor in establishing the border line. The following equation gives the spectral centroid of an audio signal:

$$C_t = \frac{\sum_{n=1}^{n=N} M_t[n] \times n}{\sum_{n=1}^{N} M_t[n]} \quad (2)$$

where $M_t[n]$ represents the magnitude of Short-Time Fourier Transform (STFT) spectrogram frame t and frequency bin n [14].

#### *2) Spectral Roll off*

The spectral roll off is a reliable feature in measuring the shape of the input signal. In the power spectrum, the spectral roll of point can be calculated by determining the boundary frequency or the number of bins at which 85% of the energy is distributed below this frequency [14]. The spectral roll off point can be found by the following equation where $M_t[n]$ is the Fourier Transform for $t^{th}$ frame and $n^{th}$ frequency bin.

$$\sum_{n}^{R_t} M_t\left[n\right] = 0.85 \times \sum_{n=1}^{N} M_t\left[n\right] \quad (3)$$

*3) Zero Crossing Rate (ZCR)*

Zero Crossing Rate (ZCR) offers a concrete method to evaluate noisiness in a given audio input [14]. ZCR is computed by keeping count of the number of times the waveform crosses zero in one unit of time. The following relation can be employed to evaluate ZCR with *I{x}* being the indicator function and $w_t$ being the signal of length *t*:

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} I\{w_t w_{t-1} < 0\} \tag{4}$$

*4) Chroma Frequency*

The Chroma frequency feature is a robust method that finds extensive use in establishing a similarity between two different pieces of music signals. It divides the spectrum into corresponding chromatic keys, and marks all the keys at their respective places of occurrence. A corresponding chord can be identified by just taking the histogram of the marked notes on scale separated as a 12-note scale.

*5) Root Mean Square (RMS)*

In an audio signal, the RMS value represents the effective and continuous power delivered by the amplifier. It provides a measure of the amplitude of the signal. It can be evidently used for discriminating among the different kinds of audio signals [15].

*6) BPM (Beats Per Minute)*

The tempo or "speed" of any piece of music is determined by simply counting the beats per minute (BPM). The beat histogram provides with certain beat periodic characteristics of a signal. In some cases BPM of an audio signal is crucial in distinguishing between the genres as its values is sometimes genre specific. The Beat Histogram gives more intuitive representation describing the temporal changes in tempo of given music piece.

## III. Methodology

The implementation of the proposed model is divided into two parts. The first part consists of training of three individual classifiers. In the second part the results of the three classifiers is fused by applying sum rule on their predicted scores. Final classification is based on the evaluated score from all the three models when combined together.

### *A. Training Classifiers*

All the three classifiers used in the proposed model are based on the Error-Correcting Output Code (ECOC) framework. It is one of the methods of coding designs which are used to reduce a multiclass problem into a series of binary problems [16]. The operation of ECOC framework is based on codeword design for each class to be classified [17]. A matrix M with values {1, -1} and dimensionality CXB is calculated where C is the number of classes to be classified and B is the number of binary classifiers taken. Each column vector corresponds to a binary classifier which separates the classes into two metaclasses and each row belongs to the target classes; hence forming a unique codeword for each class.

In the proposed model, the error correcting output code framework uses support vector machine (SVM) as binary learners based on one-versus-one coding design.

*1) Decision Tree Ensemble Model*

An ensemble model combines a number of weak classifiers to produce a strong one by enhancing the predictive performance of the individual classifier. In the proposed model, the base classifier taken is decision tree and the ensemble-aggregation method used is Adaboost.

Decision Tree is an inductive algorithm and an automatic tree structure representing data classification [18]. It is a tree based prediction model in which the whole data set is the root node and each variable of training is represented by the child nodes. Ensemble methods are used to improve the generalizability of a number of base learners by combining their separate predictions. Adaboost is a self-adapting Boosting algorithm which adapts to the error generation rate in a training process by dynamically updating the weights of every sample [19]. It fits a series of weak learners by repeatedly modifying the weights of samples in the data and the final prediction is computed by combining the predictions from all these learners using majority vote scheme or simple sum rule. At every iteration, the weights of samples producing lesser errors are reduced and the weights of samples that are resulting into predictions with larger errors are increased. The error rate is recorded using the given equation.

$$\varepsilon_t = \sum_{i=1}^{N} w_i^t \,|h_t(x_i) - y_i| \tag{5}$$

where $h_t$ is the weak classifier whose weights are to be updated, $w_i^t$ is its normalised weight of i^th^ feature and $y_i$ is the actual output.

*2) Support Vector Machine (SVM)*

Another classifier used in the proposed model is Support Vector Machine [20] which is used to find a boundary that splits two classes containing different or overlapping information. It distinguishes between two classes based on their features that are represented by unique coordinates in an n-dimensional space where n is the number of features. It creates a hyper-plane that distinctly separates the two classes. It follows an iterative algorithm, minimizing the error in each iteration using certain error functions. It handles non-linearly separable data by making use of kernel functions which modifies lower dimensional input space into higher dimensional space converting non-separable data to separable. There are four kernel functions that can be used; linear, radial basis function (RBF), polynomial and sigmoid. The proposed model uses linear kernel function. It creates certain number of support vectors which are computed by training on subsets of the whole training data. These support vectors are saved and used later for prediction, making the model memory efficient.

The separation boundary is found by operating on the most closely related points making the model robust to missing data. In the case of more than two classes, N(N-1)/2 classifiers are built where N is the number of classes to be classified. Each classifier is trained on data from the classes and these classifiers are consistently interfaced with each other.

*3) Relevance Vector Machine (RVM)*

Relevance Vector Machine is a general Bayesian framework constructed to provide sparse solutions to classification and regression problems [21]. It is a supervised learning technique which operates on Bayesian probabilistic framework during the learning process. It has an identical functional form to SVM producing generalized model with sparse kernel functions. In this study, linear kernel function is used. Unlike SVM, it provides probabilistic predictions of the data points when the model is evaluated, enabling the model to capture the uncertainty in the prediction. It uses lesser number of kernel functions while maintaining the generalization capability. The weights of the model are formerly assigned and are governed by a number of hyper parameters. Iteratively the values of these weights are updated and the most probable ones are assigned with the help of training data.

Sparse Bayesian network for classification follows Bernoulli likelihood and a sigmoidal link function to keep track of changes in the target values [22].

### *B. Combining Predictions*

All the three models compute the weight with which a sample belongs to a class in the form of posterior probabilities. The output of each classifier is a vector of dimension $t$X$C$ where $t$ is the number of test samples and $C$ is the number of classes to be classified. Each row vector is a set of probabilistic scores, indicating the extent to which that sample belongs to a particular class corresponding to each column.

These scores of all three models are fused by applying the sum rule over the posterior probabilities. The final prediction is made by selecting the column with maximum probability for each test case, representing the equivalent class. The performance of the model is then evaluated by comparing the predicted results with actual target values.

## IV. IMPLEMENTATION

The dataset used for training, validation and performance evaluation of the models is the GTZAN music database. The dataset is publicly available on the official website of MARSYAS software framework [23]. Two of the models used in this work are first trained individually then stacked upon another model for better accuracy. The combination of the final three models is used to compute the final predictions. The performance of the model is evaluated by calculating the accuracy and plotting the confusion matrices and ROC curve.

### *A. Dataset and Features*

The GTZAN dataset contains data for ten genres comprising of 1000 music files in .au format, with 100 files belonging to each genre. Each audio file is of 30seconds duration labeled as one of the ten genres: Hip-Hop, Rock, Reggae, Classical, Jazz, Blues, Pop, Disco, Country and Metal.

The sampling frequency, $F_s$ of the audio files for feature extraction was taken as 22050Hz. On account of interdependence of physical and perceptual features on each other, features from both categories were exploited and tested.

i. *Mel-Frequency Cepstral Coefficients*: The 30s audio file was divided into frames of 20ms with a frame shift of 10ms and 13 Mel-Frequency Cepstral Coefficients were calculated giving an output vector of dimension 1X13.
ii. *Statistical Spectrum Descriptors*: With an FFT window size of 512 and 24 bark bands, an SSD vector of dimension 1X168 was calculated.
iii. *Rhythm Histograms*: Adding up the frequencies in 24 critical bands, a rhythm histogram consisting of 60 bins was computed. The dimension of the rhythm histogram of each file was, hence, 1X60.
iv. *Spectral Centroid*: The audio file was divided into 26 frames and a window size of $5*F_s$ was used with $F_s$ being the sampling frequency of the audio signal and a step size of $F_{s.}$ A vector of spectral centroids with 26 frames was computed for each file having dimensions 1X26.
v. *Spectral Roll-off*: The window length and step size were taken same as that in the case of spectral centroid giving an output vector of dimension 1X26.
vi. *Zero Crossing Rate*: Taking window length $5*F_s$ and step size $F_s$, zero crossing rates were calculated for 26 frames giving an output vector of 1X26.
vii. *Chroma Frequency*: Instantaneous frequency is used to calculate the chroma frequencies of the audio signal with 12 bins giving an output vector of dimension 1X12.
viii. *RMS*: Audio signal with sampling rate $F_s$ is used to calculate the RMS value for each sample.
ix. *BPM*: Tempo for each signal was calculated which provided maximum and minimum values of BPM of the complete signal. Based on a threshold value of 0.5, the final value to be taken was decided.

All these features were computed to finally form a feature vector of dimensionality 1X333. Values of different features lied in various ranges; hence all features were normalized by subtracting the mean of individual features from each value and dividing it by its standard deviation from the mean.

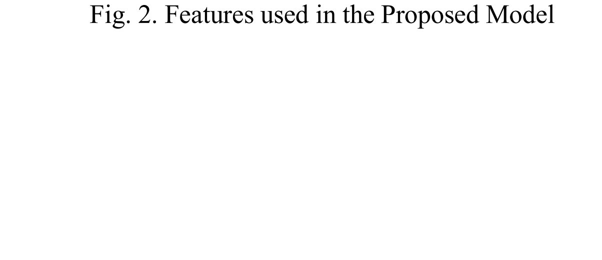


Fig. 2. Features used in the Proposed Model

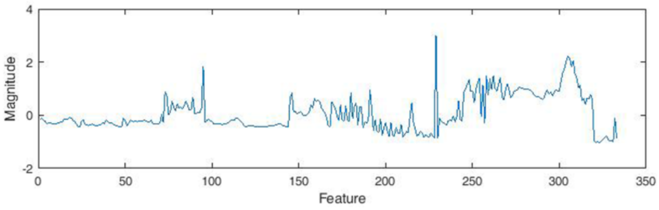


Fig. 3. Normalized Feature Vector

Data normalization is necessary since the ranges of all feature vectors vary widely and normalized data helps in fast convergence of training. One of the normalized feature vectors is plotted in Figure 3.

### *B. Models*

Standardized data was used as input to the proposed hybrid model in order to achieve fast convergence and avoid assigning unequal weights to certain features with larger values. The model proposed in this study is divided into two levels as stated earlier. The hyper-parameters of all the models were tuned using Random Grid Search.

i. An SVM classifier with linear kernel was built which was based on the ECOC scheme using one-versus-one coding design. Predictions were computed over the test set which included files that were unknown to the classifier.
ii. An ensemble model of 100 classification decision trees was trained on the training set and its performance was cross validated using k-fold cross validation with value of k as 10. This model was stacked over an SVM classifier which was constructed using the ECOC design by giving the posterior probabilities as a feature to the SVM classifier which was further trained on the same training set.
iii. Similarly, an RVM model with linear kernel was stacked upon an SVM classifier which was built using ECOC scheme. RVM was trained as a classifier on the same training dataset and a cross-validated model was constructed using k-fold cross validation with value of k as 10. The predicted values of posterior probabilities were provided as input to the SVM classifier which was trained in turn on the same training dataset with an added feature of the RVM model's prediction.

The posterior probabilities from all the three models were finally combined using simple sum rule and the final result of predicted class was evaluated over the summed up scores.

## V. RESULTS

The dataset consisted of 1000 files out of which 900 files were used for training and 100 files for testing. The values of hyperparameters of all the models were optimized using Random Grid Search for hyperparameter tuning. The combined hybrid model outperformed all the three individual models giving the maximum accuracy among all the four models. As shown in Table I, the accuracy of all models are evaluated on the same dataset and the highest accuracy of 87% resulted from the proposed hybrid model.

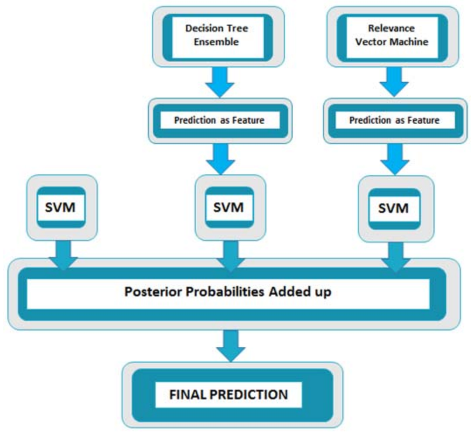


Fig. 4. Proposed Hybrid Model

This stacking of models resulted in a major improvement of accuracy. The ensemble model alone gave an accuracy of 79% while after stacking it over an SVM classifier the accuracy was enhanced to 85%. The performance of RVM classifier was also improved due to stacking giving an accuracy of 85%, while that of the individual RVM classifier on this dataset with this feature vector was 68%. Performance of models over individual classes was also evaluated as shown in Figure 5. As it is clear from the plot that classification of the genre "blues" was fairly improved by the hybrid model while maintaining the accuracies of other classes, it can be evidently stated that the proposed hybrid model is an improvement over the individual models for music genre classification.

The ROC curves plotted in Figure 6 show the performance of the proposed classifier over its entire operating range. The genres having maximum area under ROC curve are most accurately classified. Confusion matrix was also plotted to show the performance of the proposed hybrid model in Figure 7.

TABLE I. ACCURACY COMPARISON OF MODELS

| Model | Accuracy (%) |
|---|---|
| **RVM** | 68 |
| **Ensemble (Decision Tree)** | 79 |
| **SVM** | 84 |
| **RVM-SVM (stacked)** | 85 |
| **Ensemble-SVM (stacked)** | 85 |
| **Proposed** | **87** |

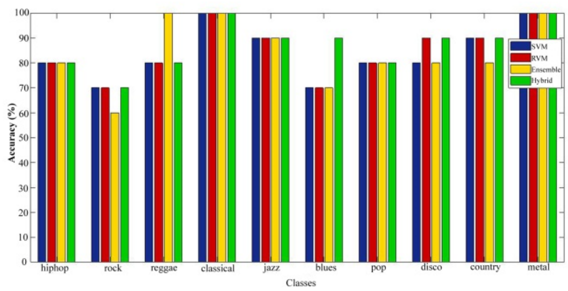


Fig. 5. Accuracy comparison for each Genre

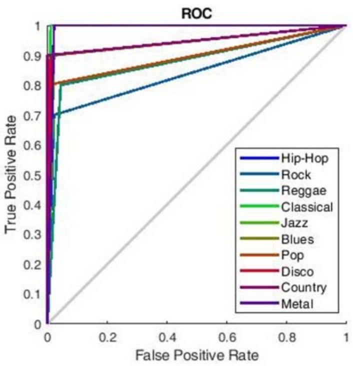


Fig. 6. ROC Curves for 10 Genres

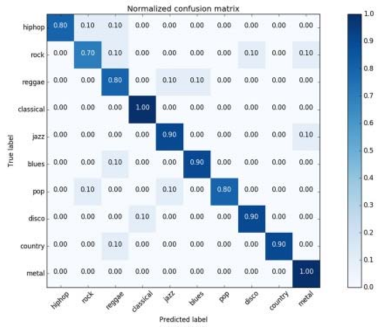


Fig. 7. Hybrid Model Confusion Matrix

## VI. CONCLUSION

In this paper, a novel and efficient hybrid model of three classifiers was proposed which combined the separate results of these classifiers to enhance their performances. The stacking of weaker classifiers over SVM also improved their accuracies by fair amount. Eventually, four different models were evaluated and compared on the same training dataset for unbiased assessment. The genres were classified with an accuracy of 87% by the proposed hybrid model of SVM-RVM-Ensemble of Decision Tree Classifier. The proposed model utilizes the advantages of the three classifiers and combines them in order to classify all the 10 genres with maximum accuracy.

## REFERENCES

[1] G. Tzanetakis, G. Essl and P. Cook. “Automatic musical genre classification Of audio signals”. Speech and Audio Processing, IEEE, pp. 293-302, 2002.

[2] L. Lu, H. Jiang and H. J. Zhang, “A Robust Audio Classification and Segmentation Method”, In Proc. ACM Multimedia 2001, Ottawa, Canada, pp. 203-211, 2001.

[3] Y. M. G. Costaa, L. S. Oliveira and C. N. Silla Jr. , “An evaluation of Convolutional Neural Networks for music classification using spectrograms”, Elsevier B.V., 2016, Applied Soft Computing 52, vol. 52, pp. 28-38, 2017.

[4] J. Dai, S. Liang, W. Xue, C. Ni and W. Liu, “Long Short-term Memory Recurrent Neural Network based Segment Features for Music Genre Classification”, in 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), IEEE, 2016.

[5] I. H. Chung, T. N. Sainath, B. Ramabhadran, M. Picheny, J. Gunnels, V. Austel, U. Chauhari and B. Kingsbury, “Parallel deep neural network training for big data on bluegene/q”, in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 745-753, 2015.

[6] Y. Zhao, D. P. Tao, S. Y. Zhang and L. W. Jin, “Similar handwritten chinese character recognition based on deep neural networks with big data”, Journal on Communications, vol. 35, no. 9, pp. 184-189, 2014.

[7] G. E. Dahl, D. Yu, L. Deng and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition”, IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 30-42, 2012.

[8] I. Mcloughlin, H. Zhang, Z. Xie, Y. Song and W. Xiao, “Robust sound event classification using deep neural networks”, IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 23, no. 3, pp. 540-552, 2015.

[9] X. Yang, Q. Chen, S. Zhou and X. Wang, “Deep belief networks for automatic music genre classification”, ntM, vol. 92, no. 11, pp. 2433-2436, 2011.

[10] S. Sigtia and S. Dixon, “Improved music feature learning with deep neural networks”, in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference, pp. 6959-6963, 2014.

[11] M. Asim and Z. A. Siddiqui, “Automatic Music Genres Classification using Machine Learning”, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 8, pp. 337-344, 2017.

[12] C. Xu, N. C. Maddage, X. Shao, F. Cao and Q. Tian , “Musical Genre Classification Using Support Vector Machines”, in IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, pp. 429-432, 2003.

[13] D. Gerhard, “Audio Signal Classification: History and Current Techniques”, University of Regina, Saskatchewan, Canda, November 2003.

[14] Y. M. D. Chathuranga and K. L. Jayaratne, “Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches”, GSTF International Journal on Computing (JoC), vol. 3, no.2, pp. 13-24, July 2013.

[15] C. Panagiotakis and G. Tziritas, “A speech/music discriminator based on RMS and zero-crossings”, IEEE Transactions on Multimedia, Vol. 7, pp. 155-156, Feb. 2005.

[16] M. A. Bagheri, G. A. Montazer and S. Escalera, “Error Correcting Output Codes for multiclass classification: Application to two image vision problems”, The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), IEEE, pp. 508-513, 2012.

[17] T. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes”, Journal of Artificial Intelligence Research, vol. 2, no. 1, pp. 263-286, 1995.

[18] F. Yuan, F. Lian, X. Xu and Z. Ji, “Decision Tree Algorithm Optimization Research Based on MapReduce”, 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 1010-1013, 2015.

[19] P. Wu and H. Zhao, “Some Analysis and Research of the AdaBoost Algorithm”, Intelligent Computing and Information Science, Communications in Computer and Information Science book series, CCIS, vol. 134, pp. 1-5, 2011.

[20] M. I. Mandel, G. E. Poliner and D. P. W. Ellis, “Support vector machine active learning for music retrieval”, Multimedia Systems, vol. 12, no. 1, pp. 3-13, 2006.

[21] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine”, J. Mach. Learn. Res., vol. 1, no. 3, pp. 211-244, 2001.

[22] M. Tipping and A. Faul, “Fast marginal likelihood maximisation for sparse Bayesian models”, in Proc. 9th Int. Workshop Artif. Intell. Statist., Key West, FL, 2003.

[23] Marsyas (Music Analysis, Retrieval and Synthesis of Audio Signals) WebPage.Available:http://marsyasweb.appspot.com/download/data_sets /,accessed 3rd October 2017.