Analytical Comparison of Classification Models for Raga Identification in Carnatic Classical Audio

Sathwik Acharya[1], Vihar Devalla[1], Om Amitesh[1] and Ashwini[1]
[1] PES University, Bengaluru, Karnataka 560085,India
PES2UG19CS453@pesu.pes.edu,omamitesh@gmail.com,
sathwik.acharya@gmail.com,ashwinib@pes.edu

Abstract. There is this famous quote which goes as follows “MUSIC is the divine way of portraying the most beautiful things about this world”. With that being said, the diversity in this language of music is immense, to say the least. Broadly, one would be well aware of the classification between Indian classical music and western music. In music Information Retrieval (MIR), raga classification has a tremendous part role in understanding the fundamentals of Indian classical music and in a multitude of other tasks like database organisation of music files to music recommendation systems. The paper encompasses a variety of techniques like ANN, LSTM and XGBoost models for the task of Raga Identification. The work is initially carried out on a set of 10 ragas and then extended to 20 ragas. Both the tasks showed impressive results with an accuracy of 99.56% and 99.43% for a set of ten and twenty raagas respectively. The process was carried out on the raagas pertaining to Carnatic music, a division of Indian classical music. The data samples for the same were obtained from a standard data set.

Keywords: LSTM, ANN, XGBoost, Raga identification, MIR

1 Introduction

1.1 A Glimpse into Indian Classical Music and its technical aspects

When one delves into Indian classical music, the one stark thing that stands out is the existence of Ragas. In simple words, Ragas are a collection of features (like aroha-avaroha) that can uniquely identify a given song. It can be broadly classified into Hindustani Music (North Indian Music) and Carnatic Music (South Indian Music). It is very interesting to note that the two systems show more common features than differences.

The arrangement of notes of a raga in ascending order is called the arohana of that raga and the arrangement of notes of a raga in descending order is called the avarohana. Different notes are called swaras in Indian classical music. The arohana and the Avarohana form the foundation in which the rest of the song is composed.

One must also remark the fact that not a lot of attention is given to western musical concepts like chord recognition etc in Indian Classical music.

In contrast to western music, raga is analogous to melody but has more intricate features than that. It is possible for two ragas to be similar but the musical effects they produce show a strong distinctive feature. This can be drawn to several reasons like the temporal arrangement, gamakas (arrangement of the notes in an oscillatory motion) or the Pakad (phrases of swaras that are unique to a Raga).

It is strongly seen that the task of Raga classification is linked to sequential tasks. As a result, the models tested are LSTM, XGBoost and Feedforward Neural Network(ANN). As you can see, we have compared Machine Learning algorithms with Deep learning models. The advent of Deep learning models have changed the entire landscape by providing state-of-the-art results in various domains. However, it is important to note that XGBoost, an ML algorithm provides state of the results here (over 99%) and captures better features in contrast to DL models.

2 Related Work

A variety of methods have been implemented in raga classification using machine learning algorithms or deep learning approaches or a combination of both. The earliest work on this can be traced to [2]. It is the first raga identification model that uses HMM (hidden markov models) with string matching algorithm approach to classify songs according to their raga. This was followed by the method of pitch class distributions and harmonic class profiles together for classification using Support Vector Machines(SVM) as in [5]. However, a glaring problem with this is that it discards temporal information (for instance, musical timing).

This was followed by an approach involving convolutional neural networks [11] to classify raga. [9] makes use of extracting the arohana and avarohana patterns to classify music into its respective raga. The drawback of this approach is that, despite arohana and avarohana form the base on which the entire raga is structured, they cannot alone capture the richness unique to a raga. [3] makes use of Time-Delayed Melody Surface(TDMS) which addresses the above problems. This model has shown state-of-the-art performance of about 98% accuracy on a dataset of 30 ragas.

There is also [4], which is a Comparative study of machine learning different approaches to the problem, much like this paper, implementing the use of C4.5, Bayesian, Random forest and K* classifiers. Lastly, one of the recent papers [12], uses the LSTM-RNN approach to classify Indian classical music. It uses the LSTM approach for two specific tasks, sequence classification and sequence ranking and retrieval (to rank similar sequences and thereby fine-tune the classifier). This model has achieved an impressive accuracy of 97% on a dataset of 10 ragas.

3 Methodology

3.1 Preprocessing

It is quite often for a musical arrangement to have an ensemble of different elements come together. These might range from the percussion kits to the instruments and the vocals. For the task of identification of raga, two approaches have been tried- one without separation of voice from the accompaniment, and the other with the separation of vocal from the accompaniment. With that in mind, Spleeter is used to isolate the vocal from the accompaniment part. It is a neural network based source separation library with pretrained models that makes it easy to train source separation models and provide state-of-the-art models for performing a plethora of source separation operations.

3.2 Feature Extraction

There are a multitude of features that come into picture for the analysis of audio samples. Here a popular python library called Librosa is used for the same. It includes the prerequisite packages to build a functional and operational MIR(Music information retrieval) system. The files in our dataset were of the standard .wav format.

Visual representation of audio samples is typically done via the help of spectrograms. The spectral features are as follows:

- Spectral Rolloff
- Spectral Bandwidth
- Chroma feature

The pitch classes of chroma features in Carnatic music are analogous to the swara notes (Sa, Ri, Ga etc). In comparison to western music this is equivalent to C, C#, D and so on. The other features are tabulated below:

Table 1. Tabulated important Features along with their range.

Feature	Definition	Equation	Range of the values
Spectral Centroid	It indicates the frequency at which the energy of the spectrum is focused upon	$\frac{\Sigma_k S(k) f(k)}{\Sigma_k S(k)}$ where S(k) is the spectral magnitude at frequency bin k, f(k) is the frequency at bin k.	284.59 to 7527.58
Zero-Crossing Rate	It is the rate at which the signal changes its sign, that is, the rate at which the value of our signal changes from negative value to 0 or from 0 to positive value and so on.	$\frac{1}{T-1}\sum_{t=1}^{T-1} II\{s_t s_{t-1} < 00\}$	0.005 to 0.7203

		s_t is the signal of length t II{X} is the indicator function(=1 if X is True, else 0)	
MFCC	The Mel frequency cepstral coefficients (MFCCs) of a waveform aare a small set of features (usually about 10–20) which elegantly describe the overall shape of a spectral waveform.	$\sum_{u=0}^{N} X[h\tau + n]W[n]e^{-j2\pi kn/}$	-58.36 to 13.261 (for MFCC7)

The reason for choosing the above features can be traced to the fact that they capture most, if not all the information about a given audio file.

4 Analysis of the Classification Models used

It is important to note that the approach for the process of feature extraction involves taking the features from the audio file by considering the 10 different samples of 30 seconds duration from the song i.e 3 samples from the beginning, 4 from the middle and 3 from the end parts of the song respectively.

The above extracted features are extracted and fed into a CSV file. Following are the different models which we have used for the task:

4.1 Artificial Neural Network (ANN)

A feedforward neural network architecture which is used for most supervised learning tasks is chosen.

The model consists of 4 fully connected layers, in entirety, with 2 hidden layers. The model begins with an input layer, which takes a batch of rows(each row consisting of 27 values as per the features of the clip) from the csv file as input, and gives an output vector of size 256. This is then passed into the first and subsequently, the second hidden layers of size 128 and 64 neurons. The input, and the two hidden layers implement a ReLU (Rectified Linear Unit) activation. The final output layer with a softmax activation gives us a vector of size 10 (for each clip input), containing the probabilities of the 10 ragas the input row can belong to.

- **Model Training and Optimisation**

The model is trained for 500 epochs, with a batch size of 128 (128 rows or clips of songs as input in one training step) with a learning rate of 0.001.

- **Loss Function**

The loss function implemented for this method is the categorical cross entropy (sparse variant in this case, as we encoded our label ragas as integers). It is apt to use this loss function as this is a classification problem with more than 2 classes. The categorical cross entropy loss is given as:

$$\text{Categorical Cross Entropy} = \sum_{i=1}^{N} y_{i,j} \cdot log(p_i)$$

Here, N corresponds to the number of classes(number of ragas in our case), and $y_{i,j}$ is the indicator function that assumes 1 when i and j are equal and j is the training label (or the target raga in this case) for the given data from the .csv file, p_i corresponds to the ith entry of the output vector from our model which corresponds to the probability of the select clip to be of that particular raga.

- **Optimiser**

Adam optimiser, an optimiser popular in training models for classification purposes is used. It is pre-set with the values for beta1 and beta 2 as 0.9 and 0.999 respectively and the value for epsilon being 1e-07.

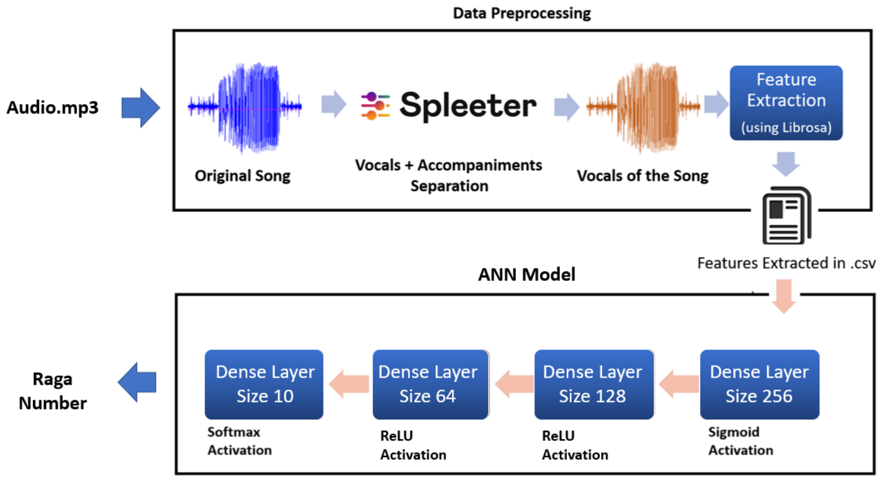


Fig 1. A Flowchart depicting the ANN(feedforward) model including the data preprocessing steps

4.2 Long Short Term Memory(LSTM)

Amongst the many models used, one of them was LSTM(Long Short Term Memory). Long Short Term Memory Model, since its inception is used mainly for sequence prediction problems and fits very well with the input samples used in the project. It relies on its feedback connections and ability to learn long-term dependencies. After trying various configurations to get the best results, we settled with a 4 layered Neural Network having 2 LSTM layers followed by a flattening layer and a Dense layer having 10 or 20 units corresponding to the number of ragas required to be classified. The model was sent into training for about 300 epochs with a batch size of 35 and an initial learning rate of 0.001. The size of the training dataset was 1680 samples and a test set of 560 samples.

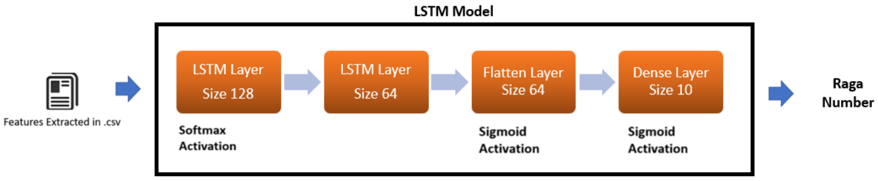


Fig. 2. Architecture of LSTM network

The first layer using LSTM with a ReLU activation function converts the data into a 128 dimensional vector and a dropout regularizer to probabilistically exclude the input and recurrent connections in the network. This solves the predominant problem (common in these tasks) of overfitting and improving model performance. The second layer has a similar working except for the fact that it converts the input it gets to a 64 dimensional vector and sends it to a flattening layer. Here data for each sample is converted into a 1D array and sent to the dense layer to get a single prediction value to each raga for the given sample. The loss function used is the same as used in the ANN model (sparse categorical cross entropy) and the Adam optimiser to better the results of the model.

4.3 XGBoost

Based on the popular decision-tree-based ensemble algorithm that uses a gradient boosting framework, XGBoost tends to outperform all other models when it comes to small sized structured data. In prediction scenarios, however, involving unstructured data, neural networks(i.e artificial neural networks ANN) shows state-of-the-art results compared to all other algorithms It(XGBoost) is an ideal blend of software and hardware fine-tuning techniques to yield impressive results using less computational resources. Another important thing to remark is its fast and efficient computational speed. The place where XGBoost stands out can be traced to its high performing GBM framework. This is realised via system optimization and enhancement of the algorithms. XGBoost uses DMatrices in place of general forms of data like numpy arrays or Pandas Dataframes, These DMatrices can contain both the features and the targets. The implementation of XGboost carries with it a number of hyperparameters. Tuning these hyperparameters, depends on the task at hand and can yield better results. It is important to note that hyper-parameters often take on a set of values, with glaring exceptions being things like regularization values. Therefore, in order to avoid the intricacies involved in hybrid discrete optimization, it is a common practise to tune the hyperparameters by discretizing the values of all hyper-parameters in question. For instance, five hyperparameters will be fine-tuned for the XGBoost model. The extent of possible values that will be taken into consideration can be seen in table 3.

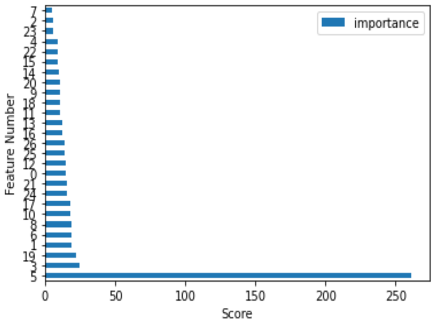


Fig. 3. Graph of Feature Importance

Table 2. Optimal values for hyperparameters

Parameter	Optimal Value
eta	0.05
Max Depth	3
Gamma	0.0
colsample_bytree	0.7
Min_child_weight	1

In very simple terms, Grid search (GS) can be traced to a brute force approach that looks for all the possible combinations of hyper-parameters in some discrete order. This is followed by the computation of the cross-validation loss for each combination and finds the optimal values in this manner. The optimal values received can be seen in table 2. These values were then fed into the model. Fig 3 gives a visual representation of the importance of the features (27 in total) for the XGBoost model.

Table 3. Grid Search for XGBoost

Parameter	Property	Range of values
eta	It controls the learning rate	[0.05,0.1,0.15,0.2,0.25, 0.3]
Max Depth	It is the maximum number of nodes allowed from the root to the farthest leaf of a tree.	[3,4,5,6,8,10,12,15]
min_child_weight	It is the minimum weight required in order to create a new node in the tree.	[1,3,5,7]
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree	[0.0,0.1,0.2,0.3,0.4]
colsample_bytree	It corresponds to the fraction of features (the columns) to use	[0.3,0.4,0.5,0.7]

5 Results

We must emphasise on the fact that the entire collection of ragas for our task was extracted from the Dunya website [7][3]. Only those ragas were chosen which had around 15 songs. This was done to ensure homogeneity of the dataset. The list of the ragas can be seen in table 6.

The following approaches were employed for the raga classification task. the results can be divided into two main branches:

A. Without separating the vocals from the accompaniment: This approach involves dividing the song into different samples and extracting the features for each of these segments. The results for the different models can be seen in table 4 and table 5

B. By separating the vocal from the accompaniment: This approach again involves dividing the song into different samples and extracting the features for each of these segments. The results for the different models can be seen in table 4 and table 5

Table 4

ANN Model

Approach	Validation loss (10 ragas)	Validation accuracy (10 ragas)	Validation loss (20 ragas)	Validation accuracy (20 ragas)
A	0.1197	96.00%	0.2309	94.17%
B	0.2162	93.07%	0.4123	89.17%

LSTM Model

Approach	Validation loss (10 ragas)	Validation accuracy (10 ragas)	Validation loss (20 ragas)	Validation accuracy (20 ragas)
A	0.0402	98.67%	0.1641	96.50%
B	0.1311	97.07%	0.1769	95.52%

Table 5 XGBoost model.

Approach	Validation accuracy (10 ragas)	Validation accuracy (20 ragas)
A	99.56%	99.43%
B	99.49%	99.21%

Table 6. List of Ragas used

Name of Raga	
Abhogi	Darbar
Ananda Bhairavi	Varali
Atana	Jaunpuri
Begada	Vasanta
Behag	Yamuna Kalyani
Bhairavi	Jaganmohini
Bilahari	ShanmukhaPriya
Hamsadhwani	Yadukula Kambhoji
Hindolam	Todi
Purvikalyani	Mayamalava Gaula

Note: A- Refers to the approach where no vocal separation is done from the accompaniment

B- Refers to the approach where vocal separation is done from the accompaniment

The images below give us a visual representation of the accuracy and losses associated with the ANN and LSTM model respectively.

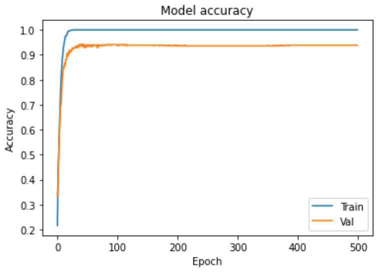


Fig 4. Accuracy plot for ANN

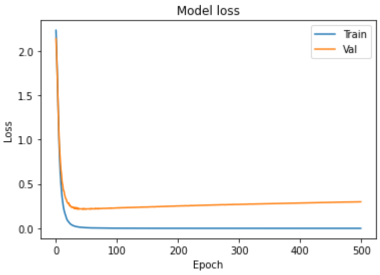
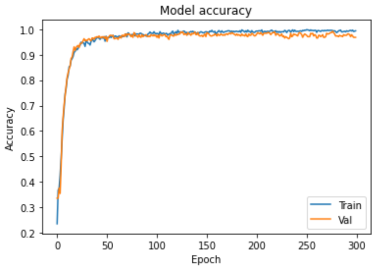


Fig 5. Loss plot for ANN



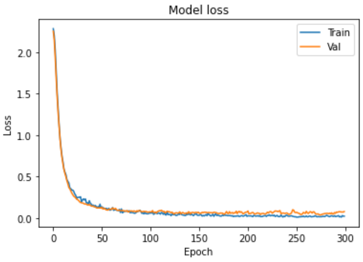


Fig 6. Accuracy plot for LSTM

Fig 7. Loss plot for LSTM

6 Conclusion and Future Scope

Drawing inferences from the table and the work, we can see that the first approach which is, without separating the voice from the accompaniment provides better results. This can be attributed to the fact that the accompaniment part can add significantly to the features of a given raga which is an unusual observation. Among these approaches the XGBoost model provided the best results. This can be attributed to its effectiveness of tree pruning, cross validation and regularisation. The same approach can be extended to identify more number of ragas from the Hindustani Classical Genre as well as from instrumental audio.

References

1. Ashwini and V. Krishna A, "Feature Selection For Indian Instrument Recognition Using SVM Classifier," 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2020, pp. 277-280, doi: 10.1109/ICIEM48762.2020.9160223.
2. G. Pandey, C. Mishra, and P. Ipe, “Tansen: A system for automatic raga identification” Proc. of Indian International Conference on Artificial Intelligence,2003, pp. 1350-1363.
3. Gulati, S., Serrà, J., Ganguli, K. K., ¸Sentürk, S., & Serra, X. (2016). Time-delayed melody surfaces for raga recognition. In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR), pp. 751–757. New York, USA.
4. Hiteshwari Sharma and Rasmeet S.Bali “Comparison of ML classifiers for Raga recognition”, International Journal of Scientific and Research Publications, October 2015 ,Volume 5
5. P.Chordia and A.Rae,”Raag recognition using Pitch-class and pitch-class dyad distributions”,Proc. of ISMIR,2007,pp. 431-436.

6. Rajeswari Sridhar and TV Geetha. Raga identification of carnatic music for music information retrieval. International Journal of recent trends in Engineering, 1(1):571, 2009.
7. S. Gulati, J. Serrà, V. Ishwar, S. Sentürk and X. Serra, "Phrase-based rĀga recognition using vector space modeling," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 66-70, doi: 10.1109/ICASSP.2016.7471638.
8. S. V. Chandan, M. R. Naik, Ashwini and A. V. Krishna, "Indian Instrument Identification from Polyphonic Audio using KNN Classifier," 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 2019, pp. 135-139, doi: 10.1109/WiSPNET45539.2019.9032860
9. S.Shetty and K.Achary,"Raga mining of Indian music by extracting arohana-avarohana pattern",International Journal of Recent trends in Engineering,vol.1,no.1,2009
10. Sankalp Gulati, Joan Serrà Julià, Kaustuv Kanti Ganguli, Sertan Sentürk, and Xavier Serra. Time-delayed melody surfaces for raga recognition. In International Society for Music Information Retrieval Conference, 2016.
11. Sathwik Tejaswi Madhusdhan and Girish Chowdhary. Tonic independent raag classification in indian classical music. 2018
12. Sathwik Tejaswi Madhusudhan, & Girish Chowdhary. (2019). DeepSRGM - Sequence Classification and Ranking in Indian Classical Music Via Deep Learning. In Proceedings of the 20th International Society for Music Information Retrieval Conference (pp. 533–540). Delft, The Netherlands: ISMIR.
13. Shrey Dutta and Hema A Murthy. Raga verification in carnatic music using the longest common segment set. In International Society for Music Information Retrieval Conference, volume 1, pages 605–611, 2015.