Computer Engineering Department

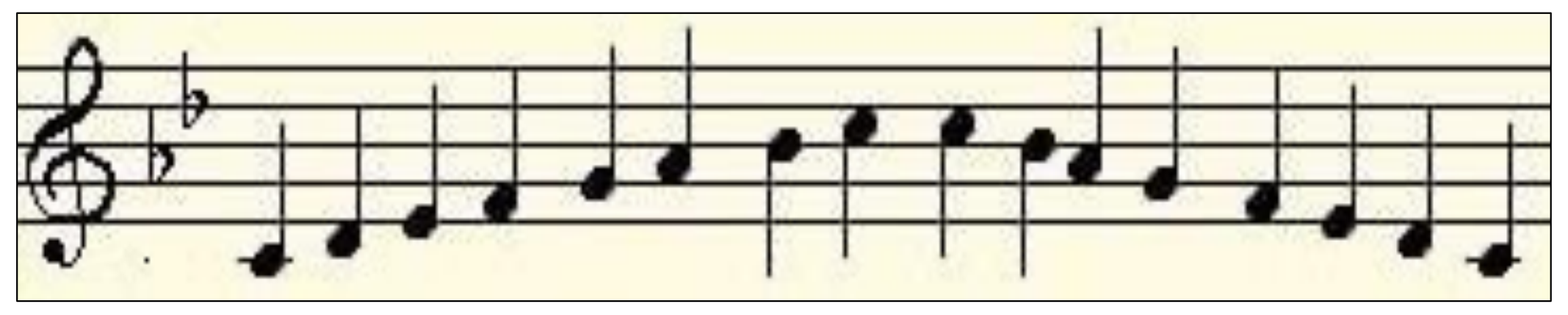
Appreciating Indian Music using AI

Project Advisor: Dr. Vishnu Pendyala

Kulkarni ,Chetan
(MS Software Engineering)
Yadav, Nupur
(MS Software Engineering)
Vadlamudi,Lokesh
(MS Software Engineering)

Introduction

Hindustani Classical music is an ancient musical form predominant in northern parts of India, Pakistan and Bangladesh. It focuses mainly on melodic development. The key element in Hindustani Classical music is **raga**. A raga is a musical theme, or a melodic framework created by choosing a specific set of notes (swaras), on a scale, ordered in melodies with musical motifs. It has characteristic intervals, rhythms, and embellishments which is used as a basis for improvisation. Raga technically is a collection of melodic atoms and a technique to develop them. Hindustani classical music focuses more on the space between the notes than the notes themselves. A musician playing a raga may use the same notes but, may improvise or emphasize on certain degrees of the scale evoking a mood that is unique to each raga. The following shows how raga Bhairav scale looks like.



As Hindustani Classical music is gaining popularity all around the world, more and more people want to learn this form of music. So, we have proposed a system for the identification of raga which will help users know the type of raga present in a song for them to practice and, also to check what they are practicing is a correct raga or not, with a provided confidence score. Hence, creating an ecosystem where Hindustani Classical music can thrive and generate interest globally which is our core objective behind designing this service.

Methodology

Data Collection

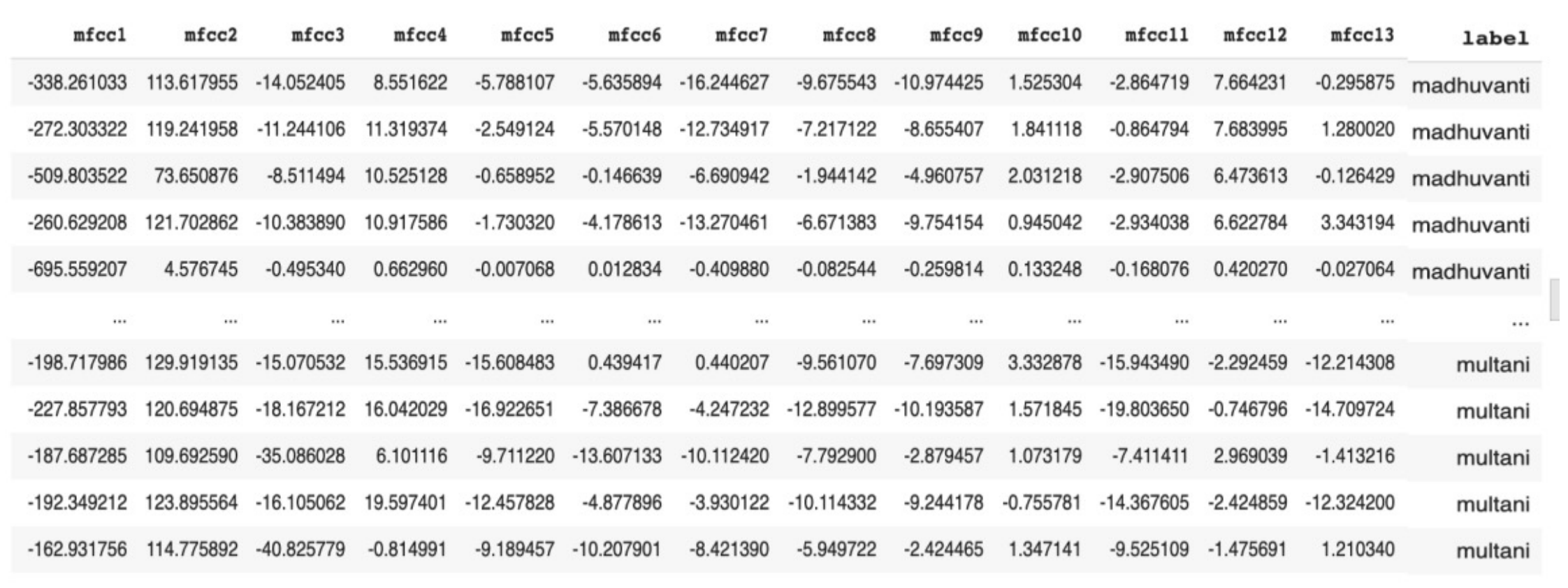
Dataset collection is the most crucial part for any machine learning task, but we could not find any relevant dataset for Hindustani Classical music available online, so we decided on creating our own dataset. The dataset collection was done in two parts.
Manually downloading songs from YouTube.
Downloading songs from Dunya using PyCompmusic library

Data Preprocessing and Feature Extraction

The HCM dataset obtained was loaded into a Google Colab Pro notebook for preprocessing and feature extraction. Initially the dataset was in the following format.
We then used librosa library to extract audio features which are understandable to a machine learning algorithm. We converted each mp3 audio data into 13 **Mel-frequency cepstral coefficients** (**MFCCs**) features and stored them into a json file along with their corresponding raga label. The MFCCs are a very powerful representation of an audio signal as it scales the frequency in order to match more closely what the human ear can hear.

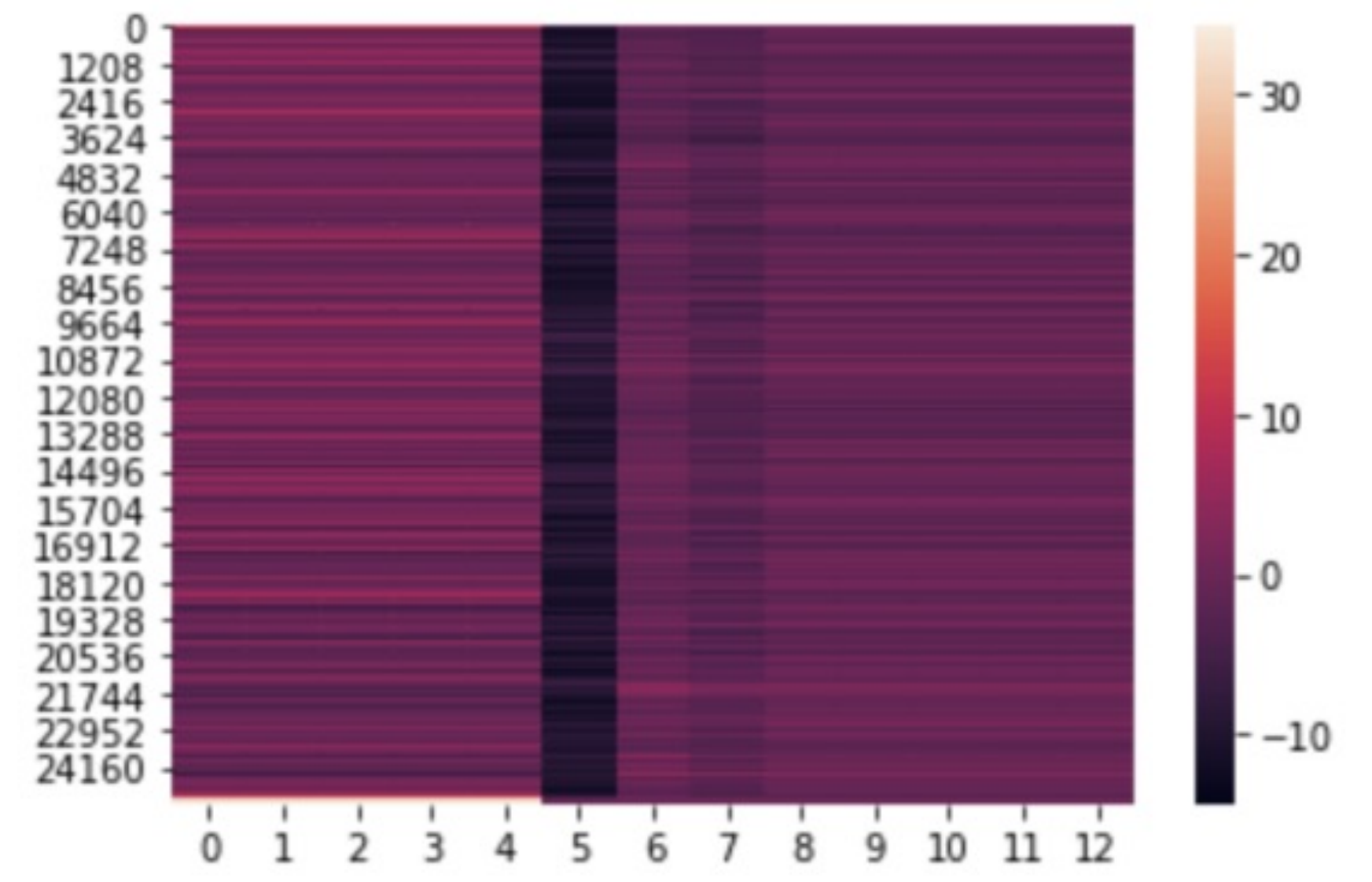
Methodology

Sample data after MFCC extraction



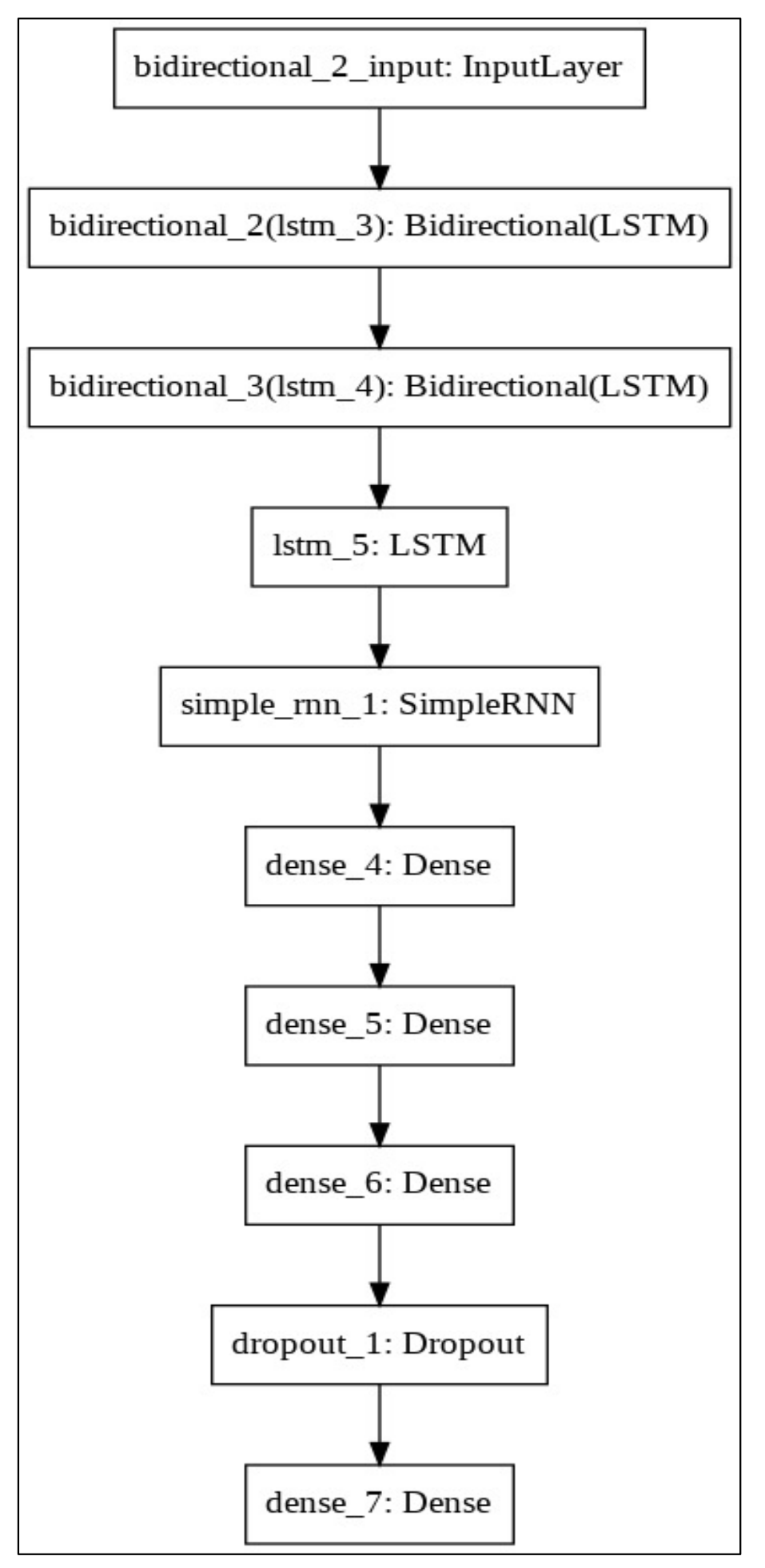
	mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	mfcc6	mfcc7	mfcc8	mfcc9	mfcc10	mfcc11	mfcc12	mfcc13	label
	-338.261033	113.617955	-14.052405	8.551622	-5.788107	-5.635894	-16.244627	-9.675543	-10.974425	1.525304	-2.864719	7.664231	-0.295875	madhuvanti
	-272.303322	119.241958	-11.244106	11.319374	-2.549124	-5.570148	-12.734917	-7.217122	-8.655407	1.841118	-0.864794	7.683995	1.280020	madhuvanti
	-509.803522	73.650876	-8.511494	10.525128	-0.658952	-0.146639	-6.690942	-1.944142	-4.960757	2.031218	-2.907506	6.473613	-0.126429	madhuvanti
	-260.629208	121.702862	-10.383890	10.917586	-1.730320	-4.178613	-13.270461	-6.671383	-9.754154	0.945042	-2.934038	6.622784	3.343194	madhuvanti
	-695.559207	4.576745	-0.495340	0.662960	-0.007068	0.012834	-0.409880	-0.082544	-0.259814	0.133248	-0.168076	0.420270	-0.027064	madhuvanti
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	-198.717986	129.919135	-15.070532	15.536915	-15.608483	0.439417	0.440207	-9.561070	-7.697309	3.332878	-15.943490	-2.292459	-12.214308	multani
	-227.857793	120.694875	-18.167212	16.042029	-16.922651	-7.386678	-4.247232	-12.899577	-10.193587	1.571845	-19.803650	-0.746796	-14.709724	multani
	-187.687285	109.692590	-35.086028	6.101116	-9.711220	-13.607133	-10.112420	-7.792900	-2.879457	1.073179	-7.411411	2.969039	-1.413216	multani
	-192.349212	123.895564	-16.105062	19.597401	-12.457828	-4.877896	-3.930122	-10.114332	-9.244178	-0.755781	-14.367605	-2.424859	-12.324200	multani
	-162.931756	114.775892	-40.825779	-0.814991	-9.189457	-10.207901	-8.421390	-5.949722	-2.424465	1.347141	-9.525109	-1.475691	1.210340	multani

Following MFCC representation of Raga Ahir Bhairav audio clip



Model Architecture

We have used a bidirectional LSTM RNN architecture for our raga identification problem which this is a very popular architecture used for sequence classification and sequence to sequence learning tasks. This neural network contains 3 LSTM layers followed by a simple RNN layer followed by dense layers. This model trains itself on MFCCs features to form an output in the form of a five-dimensional array representing the probability scores towards the five different classes we have chosen. The highest score representing the raga label for that song. We have used softmax activation function in the last layer to generate the probability scores and sparse_categorical_crossentropy as the loss function. The architectural block diagram of the model is shown in below


bidirectional_2_input: InputLayer
bidirectional_2(lstm_3): Bidirectional(LSTM)
bidirectional_3(lstm_4): Bidirectional(LSTM)
lstm_5: LSTM
simple_rnn_1: SimpleRNN
dense_4: Dense
dense_5: Dense
dense_6: Dense
dropout_1: Dropout
dense_7: Dense

Loss Function and Optimizer

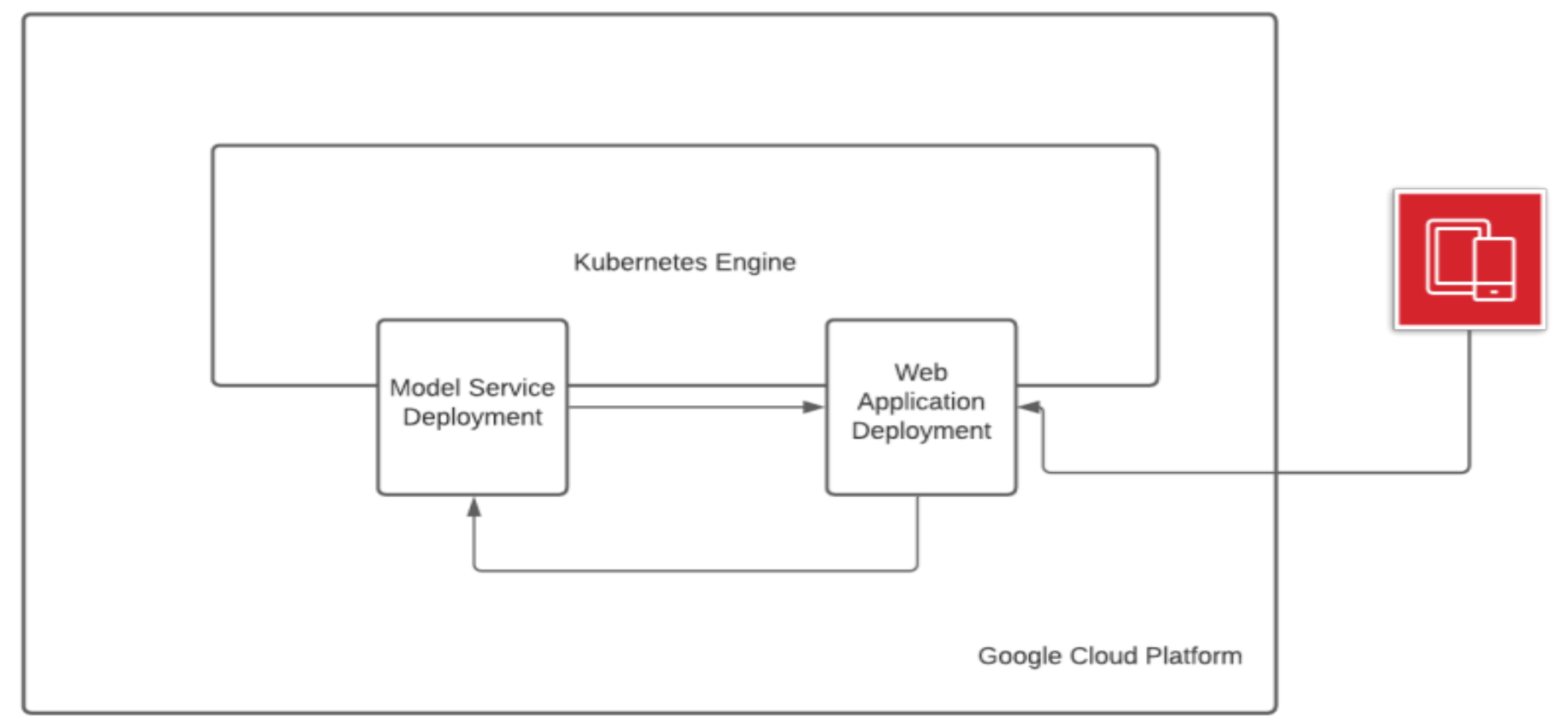
We have used sparse_categorical_crossentropy loss function which is defined as categorical crossentropy with integer targets, and we also have our targets in the integer form. This is called sparse since the target representation requires much less space than one-hot encoding. For example, a batch with b targets and k classes needs b * k space to be represented in one-hot, whereas a batch with b targets and k classes needs b space to be represented in integer form.
The optimizer used is Adam. We experimented a bit with the optimizers and different learning rates and found Adam optimizer with a learning rate of 0.001 worked the best. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. It combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

Experiments and Results

As the above experiments show Deep BiLSTM RNN architecture achieved the best accuracy of 78%. .

Model Deployment

Model deployment is one of the crucial stages in machine learning application. In order to serve thousands of users, we need our model to be highly scalable and optimized.
In order to handle hundreds of requests at the same time, we needed to have a good cloud infrastructure. So, we have used Google Cloud Platform services for hosting the model and web application. We have used Docker to create two separate images one for our Django app and another for our model which is being served using TensorFlow serving in TFX. Both the images are then deployed on a Kubernetes engine of GCP. One of the deployments contains three replica pods each serving the machine learning model. The other deployment hosts the Django web application with replicas for uninterrupted service.
Following figure shows the model and web deployment architecture. It shows the user accessing web application service which in turn uses model service to fetch predictions.



TensorFlow Extended (TFX) is an end-to-end platform for deploying production ML pipelines. In this project we have used the pusher component of TensorFlow extended platform known as TensorFlow serving to serve our ML model. By default, TensorFlow serving provides the rest endpoint for users to get their predictions.

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. Docker enables developers to easily pack, ship, and run any application as a lightweight, portable, self-sufficient container, which can run virtually anywhere. Docker containers are easy to deploy in a cloud. The two main docker containers in this project are
1) Django web application Container 2) Raga Prediction Model Container.

Kubernetes is an open-source container orchestration platform that enables the operation of an elastic web server framework for cloud applications. It offers portability, and faster, simpler deployment times. We have deployed both Django web application Container and Raga Prediction Model Container on the Kubernetes cluster of GCP

Summary/Conclusions

In this work we have proposed a deep learning solution for the problem of raga identification. Through various experiments and validations, we found that the best deep learning solution uses 3 LSTM layers (2 BiLSTM and 1 LSTM) followed by a simple RNN layer which is further followed by 3 dense layers. The system was able to distinguish between five ragas with an accuracy of 78%. The system is adaptable and can be used for other AIR tasks as well.
The work is only tested on Hindustani Classical music dataset. As part of future work, it can be used to analyze different styles of music and instruments. Also, the HCM dataset we had was very limited so maybe we can gather more data in future to achieve better results.

Key References

[1] Krishnaswamy, A. (2004, October). Melodic Atoms for Transcribing Carnatic Music. In *ISMIR* (Vol. 2004, p. 5th).
[2] Krishnaswamy, A. (2004, August). Multi-dimensional musical atoms in South-Indian classical music. In *Proc. of International Conference on Music Perception and Cognition*.
[3] Downie, J. S. (2003). Music information retrieval. *Annual review of information science and technology*, 37(1), 295-340.
[4] Gottlieb, R. S. (1977). *The major traditions of North Indian tabla drumming: Supplement*. Musikverlag Emil Katzbichler.
[5] Sharma, G., Umapathy, K., & Krishnan, S. (2020). Trends in audio signal feature extraction methods. *Applied Acoustics*, *158*, 107020.

Acknowledgements

Authors are deeply indebted to Prof. Vishnu Pendyala for providing us with opportunity to take up this project. Also thank him for his invaluable guidance and encouragement throughout the project in difficult times of pandemic