Journal for Research | Volume 06 | Issue 11 | January 2021
ISSN: 2395-7549

# kNN Ensemble for Karnatik Raga Classification

**Vinuraj Devaraj**
*PhD Fellow*
*Department of Analytics*
*Harrisburg University of Science & Technology*

**Anjana Suresh**
*Software Architect*
*Travel & Transportation*
*Mphasis Ltd.*

## Abstract

In this paper we propose a soft Voting Classifier ensemble of k-Nearest Neighbor (kNN) models to classify Karnatik music sound signals into one of six Karnatik ragas viz. Bhairavi, Kalyani, Khamboji, Mohanam, Shankarabharanam and Todi. The proposed architecture achieves a test accuracy of 91% on sound data containing noise with maximum class validation accuracy of 93.2% (for raga Khamboji) and minimum class validation accuracy of 90.4% (for raga Todi). We also observe that, sampling sounds, especially musical sounds, at intervals of 10 seconds or 20 seconds might yield better model accuracies as compared to sampling at level higher or lower than the specified duration, especially when the sound signals contain noise.
**Keywords: Voting Classifier, kNN**

## I. INTRODUCTION

Ragas in Karnatik music (Dhandapani & Pattammal, 2012) (Krishna, 2017), one of the oldest forms of Indian music, positions itself as a noteworthy example of multi-class classification problem. Ragas describe the underlying architecture of a song as it enforces the melodic structure to be followed while rendering the composition. This enforcement in turn brings melody, beauty and life to a song. Hence ragas can be treated as the lifeline of any Karnatik music composition. There are two categories of ragas – *Melakartha* and *Janya*. While there are 72 Melakartha ragas there are several thousand Janya ragas (Dhandapani & Pattammal, 2012).

Classifying a raga by listening to a musical piece is considered as a skill that an ardent Karnatik music follower a.k.a *Rasika*, continuously tries to improve by virtue of sheer number of ragas or *classes* and by years of experience in listening and/or practicing Karnatik music. Hence classifying ragas is considered as a classic example of a multi-class classification problem in Machine Learning.

In this paper we propose a fundamental classifier architecture which is an ensemble of kNN, to act as artificially intelligent agent that can perform classification with a high level of accuracy. k-Nearest Neighbor algorithm, *a.k.a kNN* (Fix & Hodges, 1951) is one of the oldest algorithms in the machine learning world that is still used extensively to solve variety of classification problems – binary and multiclass, including sound classification. It has been demonstrated widely that kNN can be used for classifying signals with high level of accuracies such as *Phonocardiogram signals* (Singh & Majumder, 2019) (kNN is used on five different datasets on Phonocardiogram signals, a high level of accuracy can be achieved for segmenting the signals), *environmental sounds* (Wang, Wang, He, & Hsu) (a hybrid of kNN and SVM was used to classify 12-class sound signal using hybrid SVM and kNN classifier), *distinguishing music and speech* (Thiruvengatanadhan, 2017) (By using features such as three MPEG-7 audio low-level descriptors, spectrum centroid, spectrum spread, and spectrum flatness a high accuracy was obtained as compared to an HMM model), *lung sounds* (Chen, Huang, Tan, Chang, & Chang, 2015) (kNN as a multiclass classifier to segmenting lung sounds to identify lung abnormalities), *gun shots* (Pichardo-Morales, Acevedo-Mosqueda, & Gomez-Coronel, November 2018) and *pulmonary acoustics signals* (Palaniappan.R, Sundaraj, & Sundaraj, 2014) (Using R.A.L.E lung sound database and MFCC features extracted from signal, SVM and kNN algorithms were compared with an inference that the accuracy of kNN were superior compared to SVM), to name a few.

With the extensive application of kNN and its demonstrated robustness in handling signals, we applied kNN to study its ability to classify sounds based on Karnatik music performances. As a part of this study, we developed and compared eight different models that classified sound signals into one of the six Karnatik ragas thereby apply kNN in the context of solving a multi-label classification problem.

## II. ANALYSIS

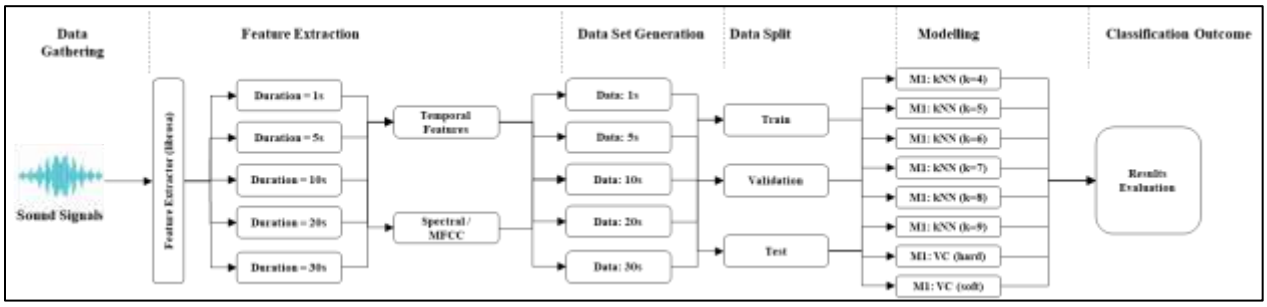


Fig. 1: Data gathering, feature extraction and modelling approach

The figure, *Fig. 1: Data gathering, feature extraction and modelling* approach , above illustrates the methodology followed in developing models using kNN classifier (Fix & Hodges, 1951). The first stage is Data Gathering where raw musical signals were extracted from (Sangeethapriya Foundation, n.d.) which is a platform (compliant with DMCA) where raw musical recordings are uploaded by volunteers. We took a sample of recordings containing vocal renditions by various artist belonging to six different ragas viz. *Bhairavi, Kalyani, Khamboji, Mohanam, Shankarabharanam* and *Todi*. Temporal and spectral features were extracted from these sound signals using python's *librosa* (Brian, et al.) library. The temporal features extracted include Zero Crossing Rate, tempo (Peter, Meinard, & Frank, 2010), Predominant Local Pulse (PLP) estimation (Grosche & Muller, 2011), Root Mean Square value, Spectral Band Width, (Klapuri & Davy, 2007), Spectral Centroid. (Klapuri & Davy, 2007), Spectral Flatness (Dubnov, 2004), Spectral Roll Off, Spectral Contrast (Jiang, Lie, Jian-Hua, & Lian-Hong, 2002) and Chroma Energy Normalized (CENS) (Meinard & Sebastian, 2011). The only spectral feature extracted were the first 13 Mel Frequency Coefficients (MFCC) (Wei, Cheong-Fat, Chiu-Sing, & Kong-Pang, 2006). The temporal and spectral features were extracted for five different durations of viz. 1 second, 5 seconds, 10 seconds, 20 seconds and 30 seconds so as to compare the variance in model behavior on duration on sounds. This resulted in five different datasets, each containing aforementioned features that were ready for modelling. We developed eight different model where in the first six were with different values of k ranging from 4 to 9 and the last two were Voting Classifier ensembles (VC) that combined the outcomes of first six kNN models to generate an outcome. The difference between these VC where on the voting method with first VC model performing a 'hard' voting and the subsequent one performing a 'soft' voting. The table, below illustrates the summary associated with data gathering.

Table – 1
Data Summary by Class

| # | Class | Sound Samples (N) | Total Duration | Data Samples (n) | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Dur = 1s | Dur = 5s | Dur = 10s | Dur = 20s | Dur = 30s |
| 1 | Bhairavi | 20 | 7.521 | 27088 | 5426 | 2719 | 1364 | 914 |
| 2 | Kalyani | 20 | 7.419 | 26717 | 5349 | 2679 | 1345 | 899 |
| 3 | Khamboji | 20 | 7.439 | 26790 | 5365 | 2687 | 1349 | 901 |
| 4 | Mohanam | 20 | 6.923 | 24934 | 4996 | 2502 | 1257 | 840 |
| 5 | Shankarabharanam | 22 | 9.372 | 33748 | 6758 | 3384 | 1696 | 1136 |
| 6 | Todi | 21 | 8.028 | 28910 | 5790 | 2901 | 1454 | 973 |
| | Total | 123 | 46.702 | 168187 | 33684 | 16872 | 8465 | 5663 |

We used Accuracy and F1 score to evaluate the outcome of each model. The table, *Table – 2 Accuracies and F1 Score by* model, illustrates Accuracies and F1 Score, associated with Training, Validation and Test, observed for each model and dataset combination.

Table – 2
Accuracies and F1 Score by model

| # | Model Name | Model Architecture | Data Sample Duration | Accuracy | | | F1 Score | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Train | Valid | Test | Train | Valid | Test |
| 1 | M1 | kNN(k=4) | 1 second | 0.938 | 0.899 | 0.898 | 0.938 | 0.899 | 0.899 |
| | | | 5 seconds | 0.949 | 0.915 | 0.911 | 0.949 | 0.915 | 0.911 |
| | | | 10 seconds | 0.952 | 0.913 | 0.914 | 0.952 | 0.914 | 0.914 |
| | | | 20 seconds | 0.952 | 0.914 | 0.914 | 0.952 | 0.914 | 0.914 |
| | | | 30 seconds | 0.952 | 0.907 | 0.893 | 0.952 | 0.907 | 0.893 |
| 2 | M2 | kNN(k=5) | 1 second | 0.936 | 0.902 | 0.901 | 0.935 | 0.902 | 0.901 |
| | | | 5 seconds | 0.944 | 0.915 | 0.911 | 0.944 | 0.915 | 0.911 |
| | | | 10 seconds | 0.948 | 0.914 | 0.915 | 0.948 | 0.914 | 0.915 |
| | | | 20 seconds | 0.946 | 0.915 | 0.912 | 0.946 | 0.915 | 0.912 |
| | | | 30 seconds | 0.947 | 0.905 | 0.897 | 0.947 | 0.905 | 0.897 |
| 3 | M3 | kNN(k=6) | 1 second | 0.932 | 0.903 | 0.901 | 0.932 | 0.903 | 0.901 |
| | | | 5 seconds | 0.941 | 0.915 | 0.911 | 0.941 | 0.915 | 0.911 |
| | | | 10 seconds | 0.944 | 0.910 | 0.917 | 0.944 | 0.910 | 0.917 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | ***20 seconds*** | ***0.942*** | ***0.913*** | ***0.911*** | ***0.942*** | ***0.913*** | ***0.911*** |
| | | | ***30 seconds*** | ***0.941*** | ***0.904*** | ***0.898*** | ***0.941*** | ***0.903*** | ***0.898*** |
| ***4*** | ***M4*** | ***kNN(k=7)*** | ***1 second*** | ***0.930*** | ***0.903*** | ***0.902*** | ***0.930*** | ***0.903*** | ***0.902*** |
| | | | ***5 seconds*** | ***0.939*** | ***0.916*** | ***0.913*** | ***0.939*** | ***0.916*** | ***0.913*** |
| | | | ***10 seconds*** | ***0.941*** | ***0.913*** | ***0.914*** | ***0.941*** | ***0.913*** | ***0.914*** |
| | | | ***20 seconds*** | ***0.937*** | ***0.913*** | ***0.906*** | ***0.938*** | ***0.913*** | ***0.906*** |
| | | | ***30 seconds*** | ***0.938*** | ***0.902*** | ***0.894*** | ***0.938*** | ***0.902*** | ***0.894*** |
| ***5*** | ***M5*** | ***kNN(k=8)*** | ***1 second*** | ***0.928*** | ***0.904*** | ***0.901*** | ***0.928*** | ***0.904*** | ***0.901*** |
| | | | ***5 seconds*** | ***0.935*** | ***0.914*** | ***0.910*** | ***0.935*** | ***0.914*** | ***0.910*** |
| | | | ***10 seconds*** | ***0.939*** | ***0.912*** | ***0.912*** | ***0.938*** | ***0.912*** | ***0.913*** |
| | | | ***20 seconds*** | ***0.936*** | ***0.909*** | ***0.906*** | ***0.936*** | ***0.909*** | ***0.906*** |
| | | | ***30 seconds*** | ***0.935*** | ***0.898*** | ***0.894*** | ***0.935*** | ***0.898*** | ***0.894*** |
| ***6*** | ***M6*** | ***kNN(k=9)*** | ***1 second*** | ***0.926*** | ***0.904*** | ***0.902*** | ***0.926*** | ***0.904*** | ***0.902*** |
| | | | ***5 seconds*** | ***0.933*** | ***0.914*** | ***0.909*** | ***0.933*** | ***0.915*** | ***0.909*** |
| | | | ***10 seconds*** | ***0.936*** | ***0.909*** | ***0.912*** | ***0.936*** | ***0.909*** | ***0.912*** |
| | | | ***20 seconds*** | ***0.932*** | ***0.909*** | ***0.905*** | ***0.932*** | ***0.909*** | ***0.905*** |
| | | | ***30 seconds*** | ***0.931*** | ***0.900*** | ***0.888*** | ***0.931*** | ***0.900*** | ***0.888*** |
| ***7*** | ***M7*** | ***VC (voting = 'hard')*** | ***1 second*** | ***0.933*** | ***0.904*** | ***0.902*** | ***0.933*** | ***0.904*** | ***0.902*** |
| | | | ***5 seconds*** | ***0.942*** | ***0.915*** | ***0.912*** | ***0.941*** | ***0.915*** | ***0.911*** |
| | | | ***10 seconds*** | ***0.945*** | ***0.912*** | ***0.917*** | ***0.945*** | ***0.912*** | ***0.917*** |
| | | | ***20 seconds*** | ***0.942*** | ***0.914*** | ***0.911*** | ***0.942*** | ***0.914*** | ***0.912*** |
| | | | ***30 seconds*** | ***0.942*** | ***0.904*** | ***0.893*** | ***0.942*** | ***0.904*** | ***0.893*** |
| ***8*** | ***M8*** | ***VC (voting = 'soft')*** | ***1 second*** | ***0.936*** | ***0.905*** | ***0.904*** | ***0.936*** | ***0.905*** | ***0.903*** |
| | | | ***5 seconds*** | ***0.945*** | ***0.918*** | ***0.914*** | ***0.945*** | ***0.918*** | ***0.914*** |
| | | | ***10 seconds*** | ***0.947*** | ***0.915*** | ***0.918*** | ***0.947*** | ***0.916*** | ***0.918*** |
| | | | ***20 seconds*** | ***0.946*** | ***0.918*** | ***0.912*** | ***0.946*** | ***0.918*** | ***0.912*** |
| | | | ***30 seconds*** | ***0.947*** | ***0.907*** | ***0.900*** | ***0.947*** | ***0.908*** | ***0.900*** |

The figure, *Fig. 2: Accuracies comparisons of* Models, offers a comparison of accuracies across various model and dataset combination.

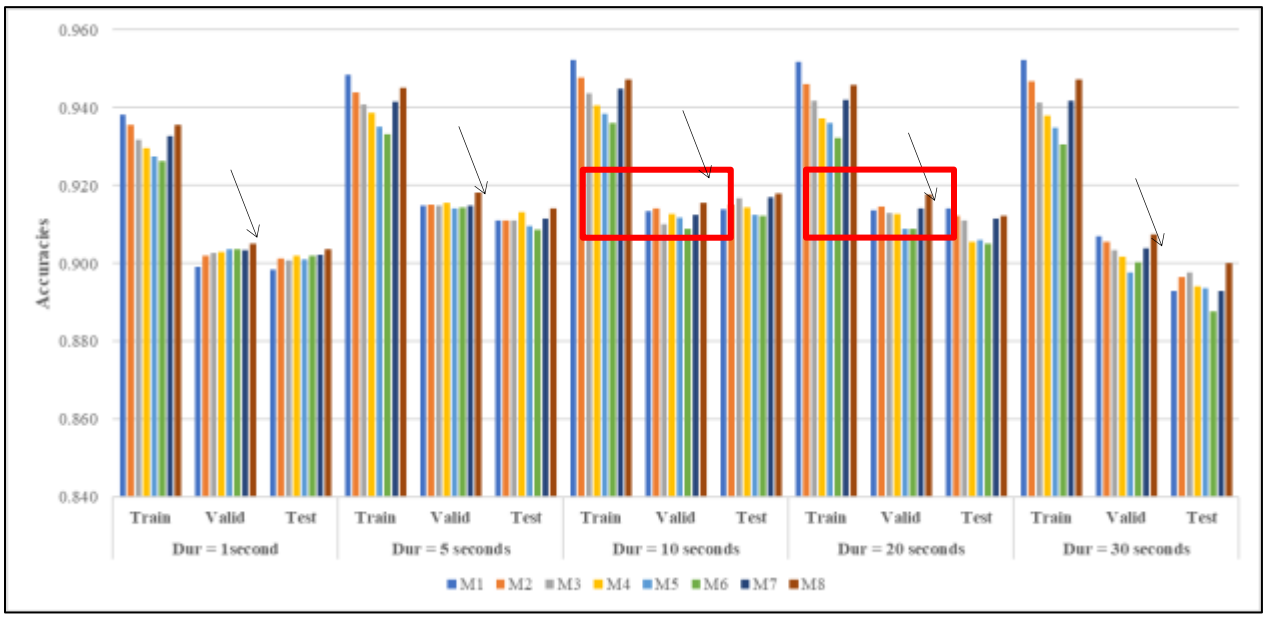


Fig. 2: Accuracies comparisons of Models

From the visual, it can be observed that the test and validation accuracies remain stable across all models for dataset containing sound sampled at 10 seconds duration and 20 seconds duration respectively, though the test accuracy is higher in case of dataset with 10 seconds duration. In addition, it was observed that model M8, a Voting Classifier with 'soft' voting (indicated by arrow mark in the figure above), achieves higher validation accuracies across all datasets. To further analyze the observed behavior, we looked at the variation of accuracies across classes (*Fig. 3: Validation Vs Test Accuracies - across* classes) for models based on data set with 10s sound samples and 20 second sound samples respectively.

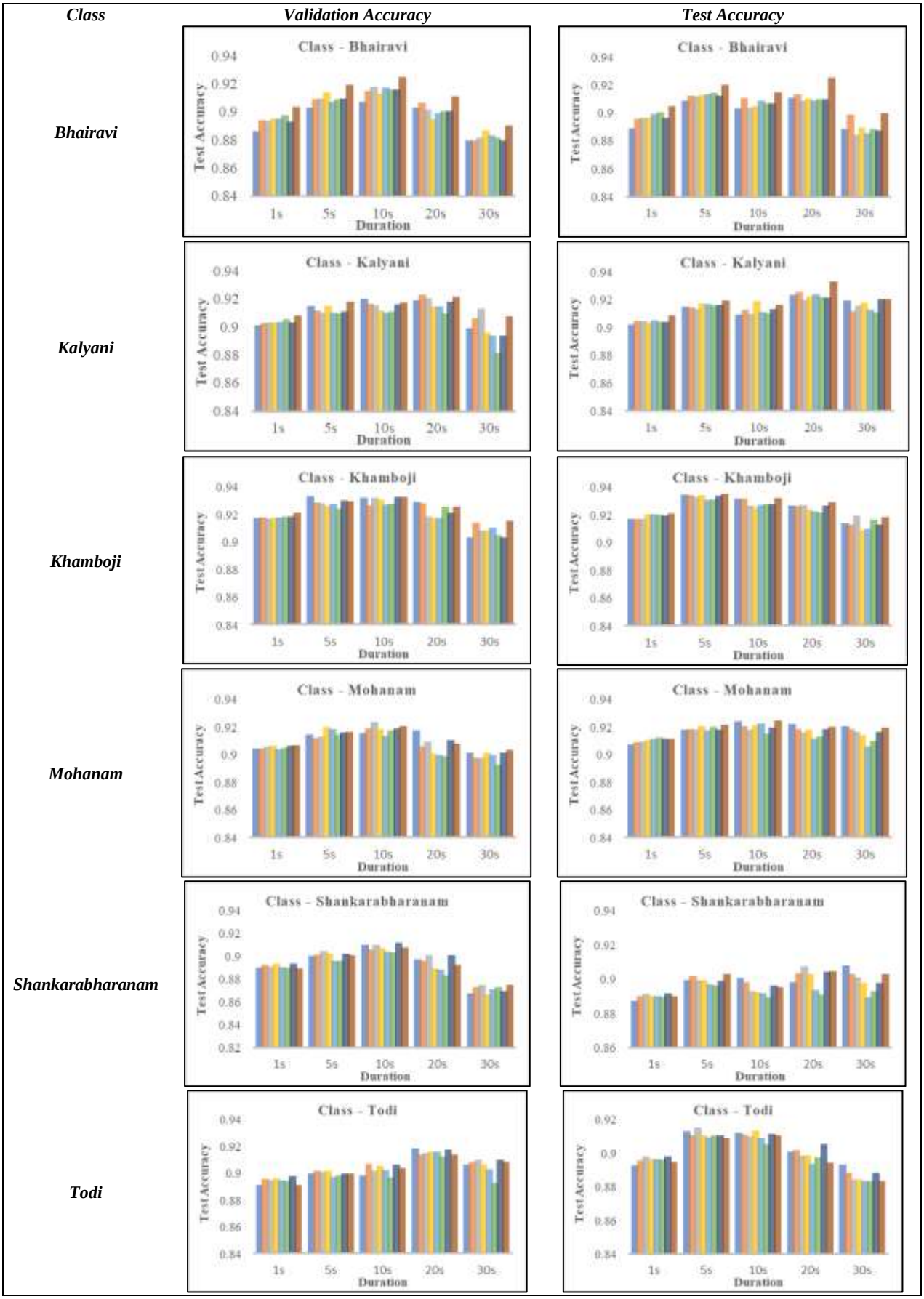


Fig. 3: Validation Vs Test Accuracies - across classes

## III. Discussion

On analyzing the distribution of model accuracies across each class, we make following observations. The first class and third class, raga *Bhairavi* and raga *Khamboji* respectively illustrates higher validation accuracy on data with 10 second duration though the test accuracy is higher on dataset containing sounds with 20 second intervals. In case of raga *Kalyani*, the second class, the test and validation accuracies are higher on dataset containing sounds with 20 seconds interval. Raga *Mohanam* exemplifies that models have higher validation & test accuracies on datasets containing 10 second interval sounds. Raga *Todi*, which is a derived or Janya raga (*derived from Hanumatodi raga*), illustrates higher test accuracies for model developed on data with 10 second duration samples as compared to other data sets, though the validation accuracy is higher on a data set with 20 second interval sounds across all models. Raga *Shankarabharanam*, the class with comparatively lower test and validation accuracies illustrates higher test accuracy for data set containing sound samples with 20 seconds and 30 seconds duration. We further analyzed the Voting Classifier with 'soft' voting to determine the distribution of accuracies across classes. The figures, Fig. 4: Validation Vs Test Accuracy by class - 10 seconds interval sound data and Fig. 5: Validation Vs Test Accuracy by class - 20 seconds interval sound data, below illustrate the validation and test accuracies distribution across classes for datasets with 10 seconds and 20 seconds intervals respectively.

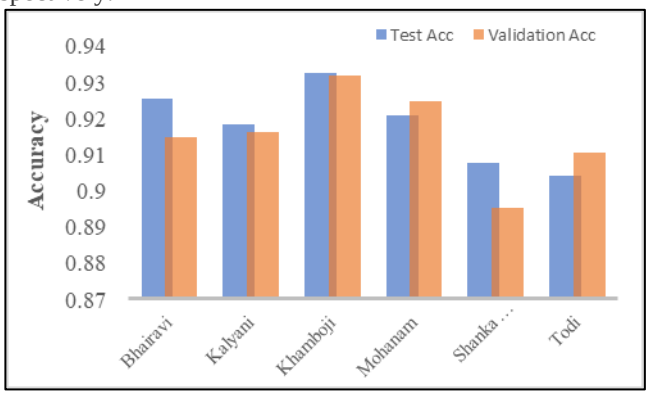


Fig. 4: Validation Vs Test Accuracy by class - 10 seconds interval sound data

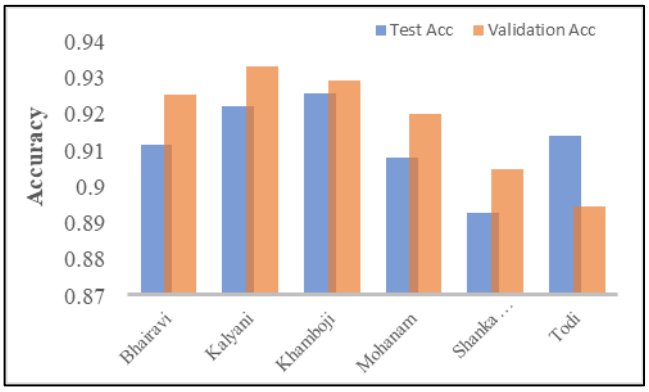


Fig. 5: Validation Vs Test Accuracy by class - 20 seconds interval sound data

The model M8, Voting Classifier with 'soft' voting, achieves validation accuracy of 91.5% and test accuracy of 91.8% on dataset with 10 seconds interval sounds. The same model architecture achieves 91.8% validation accuracy and 91.2% test accuracy on dataset with 20 seconds interval sounds. On performing a class level comparison, it can be observed that all classes except for raga *Todi* and raga *Mohanam* illustrate higher test accuracies as compared to validation accuracies on the dataset with 10 second interval sounds. However, M8 accuracies on dataset with 20 second interval sound demonstrates that the test accuracies remained lower across all classes except for raga *Todi*. In addition, the test accuracy of raga Shankarabharanam also drops below 90% for M8 model.

Hence, based on the accuracy comparison across all classes, we conclude that really cannot infer on a specific interval that can be used while classifying sounds. However, based on our analysis, we recommend that, as a best practice, it is always suggested to model sounds based on at least 2 datasets – sounds samples based on 10 seconds and 20 seconds intervals respectively, and evaluate the performance metrics before deciding on which model to choose from.

## IV. Conclusion

Based on the analysis performed, we conclude that k-Nearest Classifier (kNN), (Fix & Hodges, 1951), despite being called as "lazy learner", when used in an ensemble architecture can transform into a powerful classification algorithm which can perform at par with the state-of-the-art architectures in segmenting Karnatik music signals. Based on the approaches that we adopted, we find that the Soft Voting Classifier ensemble of kNN achieves more than 91% model accuracy in solving a multiclass problem.

As a future study, we recommend to further enhance this study by evaluating the impact of similar architectures on cleaner data, i.e., by processing sound signals through a filter bank before extracting the features and compare the results with what was observed in the current study. In addition, we also recommend to extend the voting classifier architecture to cover entire Melakartha ragas (72 ragas) and determine the performance of model.

## References

[1] Brian, M., Colin, R., Dawen, L., Daniel, P. E., Matt, M., Eric, B., & Oriol, N. (n.d.). librosa: Audio and music signal analysis in python.". In Proceedings of the 14th python in science conference, (pp. pp. 18-25. 2015.).

[2] Chen, C. H., Huang, W. T., Tan, T. H., Chang, C. C., & Chang, Y. J. (2015). Using K-Nearest Neighbor Classification to Diagnose Abnormal Lung Sounds. Sensors, 15(6), 13132–13158. doi:https://doi.org/10.3390/s150613132

[3] Dubnov, S. (2004). Generalization of spectral flatness measure for non-gaussian linear processes". IEEE Signal Processing Letters, Vol. 11.

[4] Fix, E., & Hodges, J. L. (1951). Discriminatory Analysis:Nonparametric Discrimination: Consistency Properties (Report). . Randolph Field, Texas.: USAF School of Aviation Medicine.
[5] Grosche, P., & Muller, M. (2011). Extracting predominant local pulse information from music recordings. IEEE Transactions on Audio, Speech, and Language Processing, 19(6), 1688-1701.
[6] Jiang, D.-N., Lie, L. H.-J., Jian-Hua, T., & Lian-Hong, C. (2002). Music type classification by spectral contrast feature.". In Multimedia and Expo, 2002. ICME'02. Proceedings. 2002, vol. 1, pp. pp. 113-116. IEEE, 200.
[7] Klapuri, A., & Davy, M. (. (2007). Signal processing methods for music transcription. Springer Science & Business Media.
[8] Meinard, M., & Sebastian, E. (2011). Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), .
[9] Palaniappan.R, Sundaraj, K., & Sundaraj, S. (2014). A comparative study of the SVM and K-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals. BMC bioinformatics,, 15, 223. doi:https://doi.org/10.1186/1
[10] Peter, G., Meinard, M., & Frank, K. (2010). Cyclic tempogram - A mid-level tempo representation for music signals. ICASSP.
[11] Pichardo-Morales, F. D., Acevedo-Mosqueda, M. A., & Gomez-Coronel, S. L. (November 2018). Classification of Gunshots with KNN Classifier. EATIS '18: Proceedings of the Euro American Conference on Telematics and Information Systems, (pp. Article No.: 23 Pages 1–5). doi:https://doi.org/10.1145/3293614.3293656
[12] Sangeethapriya Foundation, I. (n.d.). Retrieved from Sangeethapriya: https://www.sangeethapriya.com
[13] Singh, S. A., & Majumder, S. (2019). Classification Of Unsegmented Heart Sound Recording Using KNN Classifier. Journal of Mechanics in Medicine and Biology, Vol. 19, No. 04, 1950025 (2019).
[14] Thiruvengatanadhan, R. (2017). Speech/Music Classification using MFCC and KNN. International Journal of Computational Intelligence Research, Volume 13, Number 10 (2017), pp. 2449-2452.
[15] Wang, J.-C., Wang, J.-F., He, K. W., & Hsu, C.-S. (n.d.). Environmental Sound Classification using Hybrid SVM/KNN Classifier and MPEG-7 Audio Low-Level Descriptor. The 2006 IEEE International Joint Conference on Neural Network Proceedings. BC, 2006, pp. 1731-1735,. Vancouver: IEEE. doi:doi: 10.1109/IJCNN.2006.246644.
[16] Wei, H., Cheong-Fat, C., Chiu-Sing, C., & Kong-Pang, P. (2006). An efficient MFCC extraction method in speech recognition. IEEE International Symposium on Circuits and Systems, (p. pp. 4). Island of Kos. doi:10.1109/ISCAS.2006.1692543.