Journal of Integrated
SCIENCE & TECHNOLOGY

Automated raga recognition in Indian classical music using machine learning techniques

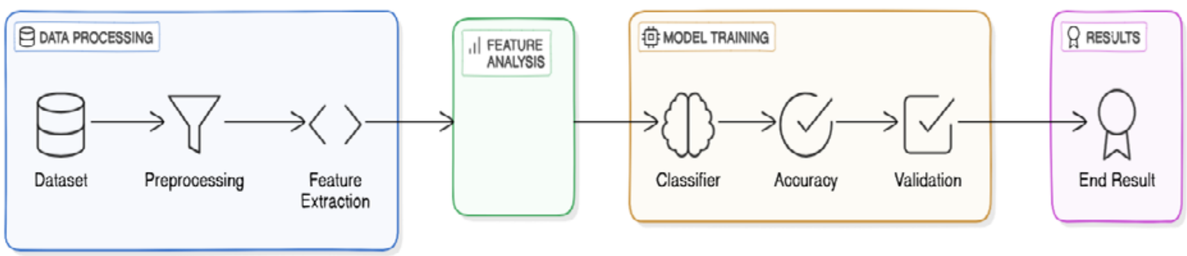
Kirankumar Humse[1], Ranjan Kumar H S[2], Veeraprathap V[3], Raju K[4], L Sri Ramachandra[5], Yathiraj G R[6], Kiran Puttegowda,[1*] Bhagyalakshmi R.[7]

[1]Department of ECE, Vidyavardhaka College of Engineering, Mysuru, Karnataka. [2]Department of Artificial Intelligence and Data Science Shri Madhwa Vadiraja Institute of Technology and Management, Bantakal, Udupi, Karnataka, India. [3]Department of ECE, ATME College of Engineering, Mysuru, India. [4]Department of Computer Science and Engineering, NITTE (Deemed to Be University), NMAM Institute of Technology, Nitte Karnataka, India. [5]Department of CSE, Govt. SKSJTI, Bangalore, Karnataka, India. [6]Department of CSE In Cyber Security, Coorg Institute of Technology. Ponnampet, Karnataka, India. [7]Department of ECE, Government Engineering College, Hassan, Karnataka, India.

Submitted on: 12-Aug-2024, Accepted and Published on: 02-Oct-2024

Review/Article

ABSTRACT



Indian classical music takes various forms depending on what part of India the dynamics behind it comes from. The current work focused on a binary classification problem involving two popular ragas: Yaman and Bhairavi. This approach allows for a more manageable dataset and facilitates effective training of the models. For machine learning (ML) classifiers, I utilized Logistic Regression, Support Vector Machine with a radial basis function (SVM(rbf)), and XGBoost. These classifiers were chosen for their diverse strengths; Logistic Regression provides a straightforward and interpretable model, while SVM(rbf) is effective for capturing complex decision boundaries. XGBoost, known for its robust performance and efficiency, enhances classification results through its boosting techniques. In addition to traditional ML approaches, we have also explored deep learning (DL) techniques by employing Deep Neural Networks (DNNs) and Long Short-Term Memory (LSTM) networks. DNNs are powerful for learning complex representations of the data, while LSTMs are particularly adept at handling sequential data, making them suitable for audio signals that vary over time. This combination of ML and DL techniques allows for a comprehensive evaluation of the classification capabilities for Yaman and Bhairavi, providing insights into the effectiveness of each method in capturing the nuances of these two distinct ragas. Our method achieves an impressive 88.1% accuracy on the full CompMusic Carnatic dataset and reaches 97% accuracy on a 10 Raga subset, positioning it as the state-of-the-art in Raga recognition. Furthermore, the approach supports sequence ranking, enabling efficient retrieval of melodic patterns from a music database, closely matching the input query sequence.

Keywords: Raag Recognition, Carnatic Music, Machine learning models, Accuracy

INTRODUCTION

Raga is the melodic framework essential for all composition and improvisation in Indian Classical Music (ICM). ICM has two primary branches: Hindustani and Carnatic music, both of which significantly influence the Indian cultural landscape and contemporary Indian music. Unlike the scales in European Classical Music, Raga is a complex entity that requires substantial skill to identify. It encompasses svara (musical notes), gamaka (oscillatory movements around a note), avarohana and arohana (downward and upward melodic movement), and numerous melodic phrases and motifs. Each Raga is designed to convey a specific emotion to the audience and possesses distinct melodic and temporal characteristics.[1,2]

Numerous examples demonstrate how two Ragas can share a similar set of notes yet differ significantly in their musical effect due to their unique temporal sequencing. The goal of this work is

[*]Corresponding Author: Kiran Puttegowda, Department of ECE, Vidyavardhaka College of Engineering, Mysuru, India. Email: kiran.mtech12@gmail.com

Cite as: *J. Integr. Sci. Technol., 2025, 13(1), 1011.*
URN:NBN:sciencein.jist.2025.v13.1011
DOI:10.62110/sciencein.jist.2025.v13.1011

the automatic recognition of Ragas in musical recordings. This capability could significantly enhance content-based recommendation systems for both ICM and contemporary Indian music. Furthermore, it would be invaluable for classifying large music libraries, enabling easier access and organization of extensive collections of music based on their Raga compositions.[3,4]

Literature Survey

Machine Learning has made significant strides in audio processing, achieving performance levels close to human capabilities, particularly in tasks like speech recognition and acoustic scene classification.[5] It's foreseeable that Machine Learning will gradually take on more audio-related tasks, with music tutoring being one of the areas poised for transformation. Recently, the first author, along with two students, published their work on converting Indian classical music between different melodic frameworks.[6] Hindustani classical music, a rich and ancient musical tradition originating from the Indian subcontinent, serves as a focal point in this endeavor. Central to Hindustani classical music is the concept of "raga," sometimes referred to as "raag," which embodies a specific musical theme or melodic structure. Ragas are akin to scales, curated from a set of notes within an octave, with each combination evoking distinct moods and emotions. While Hindustani classical music is garnering global recognition, particularly in the Western world[7], its intricacies demand dedicated guidance, making it challenging for aspiring learners. Many enthusiasts seek to learn Hindustani classical music out of personal passion or curiosity, yet its complexity underscores the necessity for continuous mentorship and guidance. According to experts,[8,9] a raga is technically described as a compilation of melodic atoms and a methodology for their development. In Hindustani classical music, the emphasis lies more on the intervals between the notes rather than the notes themselves. When a musician performs within a melodic framework, they may utilize identical notes but introduce improvisations or accentuate specific degrees of the scale, thereby eliciting a mood that is distinctively characteristic of the melodic framework. The zero-crossing rate quantifies the frequency at which a signal transitions between positive and negative values. Its utility spans various applications including music discrimination, and genre classification.[10]

On the other hand, the Spectral centroid identifies the center of mass of a spectrum, often referred to as the brightness feature of a sound. It serves as a vital indicator of timbre, and finds extensive usage in music classification and mood classification.[11] Spectral roll-off characterizes the shape of a signal by denoting the frequency below which 95% of the signal energy is contained. Its applications range from musical instrument classification to audio-based surveillance systems.[12-17]

Meanwhile, Mel frequency cepstral coefficients (MFCCs) derived from the cepstral representation of an audio clip offer a concise description of its spectral envelope. Typically comprising a small set of 10-20 features, MFCCs closely mimic human voice characteristics due to their mel-scale frequency spacing. This versatile feature is integral to various applications such as music information retrieval, speech enhancement, recognition, and genre classification, as well as tasks like vowel detection.[18–21] Many machine learning and deep learning models are used for signal processing operation like image compression, networks attacks and fruit desease detections.[22]

Proposed Methodology

Figure 1 provides a detailed overview of the workflow and architecture used in the automatic Raga recognition process. In this approach, Raga identification is framed as a sequence classification challenge, which is effectively addressed using Long Short-Term Memory (LSTM) Networks. The underlying architecture is based on Recurrent Neural Networks (RNNs), which can be visualized as a series of interconnected layers where the output from one layer is passed as input to the next. LSTMs, a refined version of RNNs, are adept at capturing long-term dependencies by retaining relevant information over extended sequences, much like how the human brain processes and recalls context based on prior experiences. This memory retention capability makes LSTMs particularly suitable for complex Raga recognition, as Ragas consist of intricate melodic patterns and temporal dependencies that span across long sequences of notes.

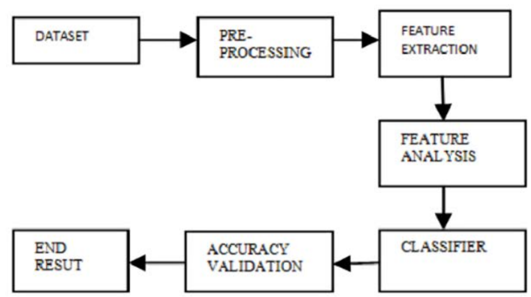


Figure 1. Overview of the Raga Inference Process

The diagram illustrates a workflow for classifying audio signals into predicted class labels using a deep learning approach. Here's a detailed breakdown of each step in the process:

A. Data Preparation

The audio files were all converted to WAV format to ensure compatibility with Python's *librosa* module for audio analysis. Each file was then segmented into samples of uniform duration, specifically 10-second clips, with an overlap of 5 seconds between consecutive segments. For instance, a 20-second audio file would be split into three segments: 0–10 seconds, 5–15 seconds, and 10–20 seconds. This sampling approach assumes that a 10-second clip provides enough information to identify a Raga, while the overlapping segments maximize the amount of data for analysis without redundancy.

B. Feature extraction

Various existing audio feature extraction techniques were explored, including well-known methods like Mel-frequency cepstral coefficients (MFCC), chroma features, and spectral contrast, which are commonly used in audio analysis tasks. Additionally, new techniques were experimented with, aimed at improving the recognition of complex musical patterns in Ragas. These novel approaches involved enhancing temporal and harmonic feature extraction to better capture the nuances of Indian classical music. By comparing the performance of both established and newly developed methods, the most effective feature extraction techniques were identified and integrated into the Raga recognition system.

i. Raw Signal

When loading an audio file using the librosa module in Python, the raw signal is represented as a time series of amplitude values. This time series data reflects the changes in amplitude (or loudness) of the audio over time. Each value in this time series corresponds to a specific point in the waveform of the audio, allowing for a detailed representation of the sound.

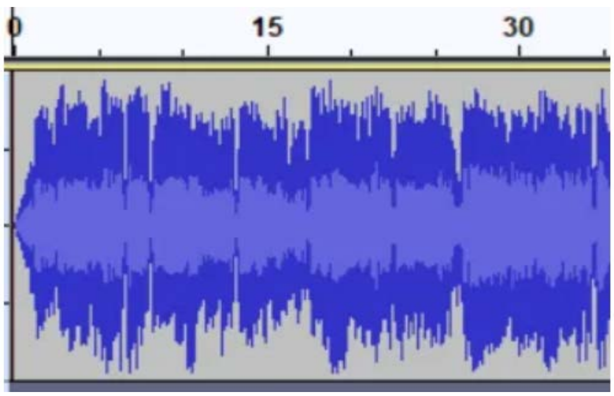


Figure 2. Sample audio signal representation

At a sampling rate of 22,050 Hz (i.e., 22,050 samples per second), each 10-second audio sample generates a total of 220,500 data points. This means for each 10-second clip, we end up with 220,500 individual features representing the raw audio signal. However, both machine learning (ML) and deep learning (DL) classifiers struggled to effectively train on this high-dimensional raw audio data due to the sheer volume of features and the complexity of the signal.

ii. Fourier transform and spectrogram

Since a raga is characterized by specific patterns of relative frequencies, it makes intuitive sense to extract features in the frequency domain rather than the time domain. Applying the Fourier transform to the time-domain audio signal gives us the individual frequency components and their magnitudes, providing insight into the frequency content of the signal. A spectrogram, which is essentially a sequence of Fourier transforms over small, overlapping time windows (typically 20-40 ms), displays how the frequencies present in the audio vary over time. This time-frequency representation helps capture the evolving harmonic structure of the raga. Here's an example of a spectrogram to illustrate in figure 3.

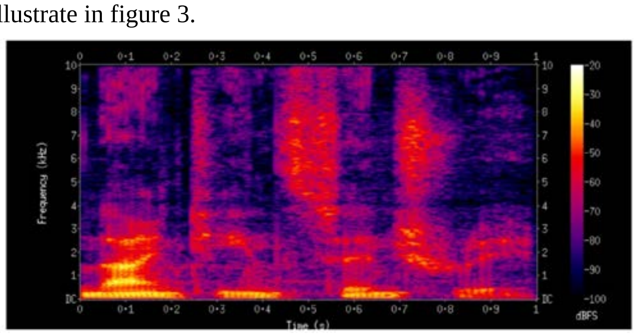


Figure 3. Spectrum representation using Fourier transform

iii. MFCC

Another widely used audio feature in research and applications is the Mel Frequency Cepstral Coefficient (MFCC). While MFCC is effective for various tasks such as genre detection, speech recognition, and emotion analysis, it proved less useful in our project for distinguishing ragas. The reason lies in its primary design, which focuses on mimicking the human ear's perception, making it suitable for voice-related tasks but less effective for capturing the intricate and continuous variations in musical pitches that are fundamental to ragas. As shown in the comparison of MFCCs for both classes, the features were not sufficiently distinguishable, leading to unsatisfactory model training. Figure 4 shows the mean MFCC of raga yaman.

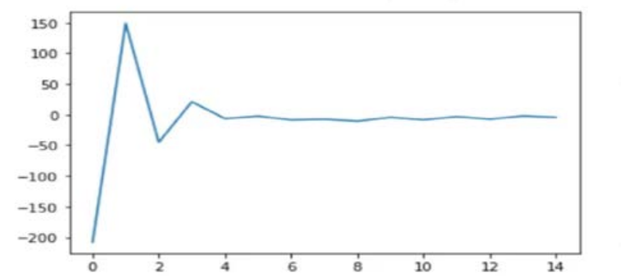


Figure 4. Representation of Mean MFCC for Yaman raaga

iv. Main Frequency

To place greater emphasis on frequency components, we have attempted to extract only the most dominant frequency within each time window. Rather than analyzing the entire spectrogram, this approach focuses solely on the frequency with the highest power for each time window. This simplified frequency representation is expected to capture the most significant tonal information relevant to raga recognition, reducing the noise from less relevant frequencies. The figure 5 depicting the dominant frequency over time for a given audio sample. By concentrating on these prominent frequencies, the model can better align with the structure of ragas, which often involve specific melodic patterns tied to particular pitches.

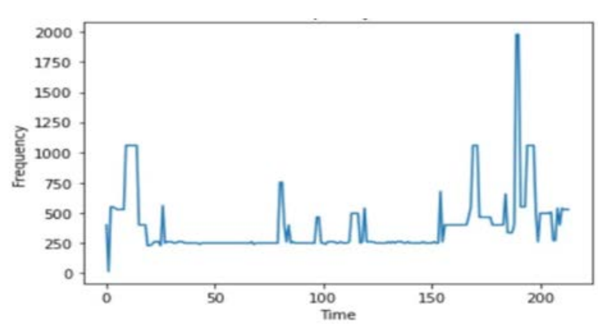


Figure 5. Frequency vs time representation of sample yaman raga

C. DNN Model tuning

To optimize the model performance for Raga recognition, several variations of key parameters were explored:

1. **Optimizer:** Stochastic Gradient Descent (SGD) and Adam were tested to assess convergence speed and accuracy. Adam, known for its adaptive learning rates, often yielded better results in deep learning contexts.

2. **Learning Rate:** Ranges from 0.01 to 0.00001 were experimented with. A higher learning rate led to faster convergence but risked overshooting, while a lower rate offered more precise convergence at the cost of slower training times.

3. **Batch Size:** Tested values between 8 to 128. Smaller batch sizes (like 8) tend to offer more granular updates, while larger batch sizes (like 128) stabilize gradient estimates but may converge slower.

4. **Number of Deep Layers:** Deep architectures with 2 to 6 layers were tested. Increasing the number of layers allows the model to capture more complex patterns but risks overfitting and longer training times.

5. **Kernel Initializers:** RandomUniform and HeUniform were evaluated. HeUniform, often preferred in deep networks, is designed to maintain a balanced variance, especially in ReLU-based networks.

6. **Regularization:** L1 and L2 regularization techniques were employed to reduce overfitting. L2 typically led to better generalization in this case, especially when combined with deep networks.

7. **Dropout Layers:** To prevent overfitting, dropout layers were inserted. By randomly deactivating a portion of neurons during training, the model was encouraged to develop more robust features.

These variations were crucial in fine-tuning the model to perform optimally for Raga recognition.

RESULT AND DISCUSSION

After effectively training the model using an LSTM as outlined in the paper, the first experiment involved replacing the LSTM block with a GRU (Gated Recurrent Unit). The LSTM architecture features a cell state that functions like a conveyor belt, carrying information through all the LSTM cells, along with a hidden state. Information flow is managed by components known as gates—specifically, the input, forget, and output gates—which determine what information to update, retain, or produce. These gates incorporate a sigmoid neural network layer and a pointwise multiplication operation, enabling the selective passage of information.

A GRU streamlines this architecture by integrating the forget and input gates into a single update gate and consolidating the cell and hidden states into a reset gate. This modification leads to a reduction in the model's complexity, as the operations among these components are adjusted accordingly. Despite having fewer gates, GRUs deliver performance comparable to LSTMs, making them a preferred choice when faster training and reasonable accuracy are essential, particularly under infrastructural constraints.

In our experiment, we replaced the LSTM block with a GRU in PyTorch while keeping all other model parameters and layers unchanged. We also tested various majority voting thresholds on the final sample predictions to evaluate their impact on model accuracy. Following successful training with 10 classes, we expanded our experiment to encompass all 40 classes in the dataset. The data consolidation code was modified accordingly, utilizing 200 random samples of length 5000 from each recording, resulting in a total of 72,000 samples for training and 24,000 samples for testing.

Classifier	Logistic Regression	SVM	XGBoost	DNN
Val Accuracy	0.89	0.96	0.97	0.9

Classifier	Logistic Regression	SVM	XGBoost	DNN
Test Accuracy	0.6	0.64	0.59	0.61

Figure 6. Classification Accuracy of various machine learning models

Figure 6 illustrates the classification accuracy of various machine learning models, highlighting the impressive performance of the XGBoost algorithm, which achieves a remarkable accuracy rate of 97%. This figure serves as a visual representation of the comparative effectiveness of different models employed in the classification task. Each model is evaluated based on its ability to correctly classify instances from the dataset, showcasing a range of accuracies that reflect their individual strengths and weaknesses. The notable success of XGBoost can be attributed to its advanced boosting techniques, which iteratively refine predictions by focusing on the errors made by previous models. This ensemble method not only enhances accuracy but also provides robust performance across diverse datasets. The high accuracy rate of 97% underscores the model's capability to generalize well on unseen data, making it a strong candidate for applications requiring reliable and efficient classification. Overall, Figure 6 emphasizes the superiority of XGBoost relative to other machine learning models, reinforcing its status as a leading choice for classification tasks in various domains.

Sr. No.	Audio File	Prediction
1	Yaman	Yaman
2	Yaman	Bhairavi
3	Yaman	Yaman
4	Yaman	Yaman
5	Yaman	Yaman
6	Yaman	Bhairavi
7	Yaman	Yaman
8	Yaman	Bhairavi
9	Bhairavi	Yaman
10	Bhairavi	Bhairavi

Figure 7. Prediction results of various Indian ragas

Figure 7 presents the prediction results for various Indian ragas, showcasing the model's ability to accurately identify and classify these traditional musical compositions. Each raga is represented with its corresponding predicted label, illustrating the model's performance across the different classifications. The figure provides insights into the effectiveness of the implemented algorithms in recognizing the intricate patterns and nuances characteristic of Indian ragas.

The results highlight not only the accuracy of the predictions but also the model's capacity to distinguish between the subtle differences in melody, rhythm, and mood that define each raga. By displaying the predicted outcomes alongside the actual labels, Figure 7 facilitates an evaluation of the model's precision and reliability in the context of Indian classical music. This visualization is essential for understanding how well the model captures the complex features of the ragas and informs potential areas for improvement. Overall, Figure 7 serves as a valuable tool for assessing the classification performance of the model in recognizing the rich tapestry of Indian ragas, demonstrating its applicability in music analysis and research.

While the method showcased superior performance compared to the then-state-of-the-art PCD and VSM methods, it wasn't devoid of challenges. A notable issue emerged concerning Ragas within allied sets, which share common swaras and similar musical phrases. This similarity posed a significant challenge, leading to confusion during the analysis. Additionally, there were occasional inaccuracies in estimating the tonic pitch, further highlighting the complexity and nuances involved in computational music analysis. These observations underscored the need for further refinement and augmentation of the method to address such intricacies effectively.

Conclusion and Future work

In this work, we researched and implemented solutions for automatic Raga classification in Indian Classical Music recordings. Our goal was to develop an algorithm that can accurately identify the Raga of any given recording, regardless of the melody's complexity, the number of instruments, or temporal variations. With increased space and computing power, there is significant potential to enhance the classification of a larger number of ragas—such as expanding from the current dataset to include 20 or more ragas—by employing more sophisticated architectures. By implementing cascaded models that integrate Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, coupled with Deep Neural Networks (DNNs), the model can leverage the strengths of each architecture to achieve superior performance. Cascaded CNN/LSTM models are particularly well-suited for this task due to their ability to extract hierarchical features from audio signals. The CNN component excels at capturing local patterns and spatial hierarchies, which are crucial for recognizing the intricate textures and melodic contours of ragas. Meanwhile, the LSTM component can effectively handle temporal dependencies, allowing the model to understand how these features evolve over time, which is essential for the rhythmic and melodic progression inherent in Indian classical music. Incorporating DNNs into this architecture further enhances the model's capacity to learn complex representations and patterns within the data. DNNs can effectively process high-dimensional inputs, providing a powerful means to refine the classification results obtained from the CNN and LSTM layers. This multi-faceted approach not only broadens the model's capability to classify a wider range of ragas but also improves its robustness against variations in performance, such as different instrumental renditions or vocal interpretations.

Furthermore, the additional computing power would facilitate more extensive training datasets, enabling the model to learn from a diverse array of raga performances. This would contribute to better generalization and accuracy when classifying new, unseen data. Ultimately, by exploring this advanced methodology with cascaded CNN/LSTM models and DNNs, researchers can unlock new possibilities in the automated classification of Indian ragas, pushing the boundaries of what is achievable in music analysis and classification.

Conflict of Interest Statement

Authors declare that there is no conflict of interest for publication of this work.

References and notes

1. M. Velankar, A. Deshpande, P. Kulkarni. Melodic pattern recognition in Indian classical music for raga identification. *Int. J. Inf. Technol.* **2021**, 13 (1), 251–258.
2. U. Farooq, K.K.S. Reddy, K.S. Shishira, M.G. Jayanthi, P. Kannadaguli. Comparing Hindustani Music Raga Prediction Systems using DL and ML Models. *International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications, ICETCS 2024*. 2024, pp 1–6.
3. A. Paul, S. Pahadsingh, T. Kar, Shruti. Carnatic Classical Raga Identification Using Stacked Ensemble Learning Model. *2024 1st International Conference on Cognitive, Green and Ubiquitous Computing, IC-CGU 2024*. 2024, pp 1–6.
4. N. Ananth, R.M. Bharadwaj, S.S. Shekar, S. Jain, V.R. Srividhya. Investigation into the Identification of Ragas: A Detailed Analysis. *IEEE International Conference on Data Engineering and Communication Systems, ICDECS 2024*. 2024, pp 1–6.
5. S. Aswale, P.C. Shrivastava, R. Ranjan, S. Shende. Indian Classical Music Recognition using Deep Convolution Neural Network. *Int. J. Electr. Electron. Res.* **2024**, 12 (1), 73–82.
6. S.B. Shirude, S.R. Kolhe. Recognizing Raga of Indian Classical Songs Using Regular Expressions. *Intelligent Systems for Smart Cities*. 2024, pp 367–383.
7. S. Mewada, F. Patel, S. Degadwala, D. Vyas. Enhancing Raga Identification in Indian Classical Music with FCN-based Models. *International Conference on Sustainable Communication Networks and Application, ICSCNA 2023 - Proceedings*. 2023, pp 980–985.
8. D.C. Shubhangi, B. Gadgay, M. Moolbharathi. Machine Learning Based Indo-Western Music Identification, Classification and Disease Healing. *5th IEEE International Conference on Cybernetics, Cognition and Machine Learning Applications, ICCCMLA 2023*. 2023, pp 454–458.
9. K.S. Patil, P.J. Deore, V.S. Kharote. Raga Identification Using Mel Frequency Cepstral Coefficient. *Int. J. Innov. Eng. Sci.* **2023**, 8 (9), 46–50.
10. R. Jain, C. Arthur. An Algorithmic Approach to Automated Symbolic Transcription of Hindustani Vocals. *ACM International Conference Proceeding Series*. 2023, pp 12–21.
11. A.R. Chhetri, K. Kumar, M.P. Muthyala, M.R. Shreyas, R.A. Bangalore. Carnatic Music Identification of Melakarta Ragas through Machine and Deep Learning using Audio Signal Processing. *2023 4th International Conference for Emerging Technology, INCET 2023*. 2023.
12. B.S. Gowrishankar, N.U. Bhajantri. Raga classification using enhanced spatial bound whale optimization algorithm. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, 30 (2), 825–837.
13. A. Sharma, A. Salgaonkar. Raga Recognition Using Neural Networks and N-grams of Melodies. In *Advances in intelligent systems and computing*; **2023**; pp 93–109.
14. R. Jain, C. Arthur. An Algorithmic Approach to Automated Symbolic Transcription of Hindustani Vocals. *ACM International Conference Proceeding Series*. 2023, pp 12–21.
15. P. Kavitha, J. Charles, L.S. Lekamge. A Deep Learning Based Approach for Janya Raga Classification in Carnatic Music. *6th SLAAI - International Conference on Artificial Intelligence, SLAAI-ICAI-2022*. 2022.
16. M. Shanthamallappa, K. Puttegowda, N.K. Hullahalli Nannappa, S.K. Vasudeva Rao. Robust Automatic Speech Recognition Using Wavelet-Based Adaptive Wavelet Thresholding: A Review. *SN Comput. Sci.* **2024**, 5 (2), 248.

17. K. Anitha, B.D. Parameshahcari, Kiran. Fisher-Jenks Classification Technique for Music Raaga Identification. In *IEEE 1st International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics, AIKIIE 2023*; IEEE, **2023**; pp 1–6.
18. K. Puttegowda, S.K. DS, S. Mallu, et al. Automatic COVID-19 Prediction with Comprehensible Machine Learning Models. *Open Public Health J.* **2024**, 17 (1).
19. K. P., S. K. V., V. Ravi, et al. A New Deep Learning Model based on Neuroimaging for Predicting Alzheimer's Disease. *Open Bioinforma. J.* **2023**, 16 (1).
20. K.N. Bharath, K. Suresh Babu. Optimal machine learning model based medical image compression techniques for smart healthcare. *J. Integr. Sci. Technol.* **2024**, 12 (5), 821.
21. N. Bahaley, A. Sharma. BDMDNA: Design of an efficient Blockchain-based Deep learning model for identification and mitigation of dynamic network attacks. *J. Integr. Sci. Technol.* **2024**, 12 (6), 829.
22. C.P. Vijay, K. Pushpalatha. DV-PSO-Net: A novel deep mutual learning model with Heuristic search using Particle Swarm optimization for Mango leaf disease detection. *J. Integr. Sci. Technol.* **2024**, 12 (5), 804.