RIYAZ

Team 15

AI3011

Hindustani Classical Raga Identification & Recommendation Model

Hindustani classical music (HCM), a branch of Indian Classical Music (ICM) has evolved over centuries in the Indian Subcontinent. It is centered around **ragas** : structured melodic frameworks defined by specific note sequences, characteristic phrases, and emotional or temporal associations. In this digital age, identifying and preserving ragas is vital. This study integrates traditional musical theory with technology to create a Raag Identification and Recommendation System, aiding learners, composers, and enthusiasts worldwide.

Lakshay Aggarwal, Lakshit Tyagi, Preesha Katial

Problem Statement

Problem Statement

What is the problem you are trying to solve and why?

Indian Classical Music

Hindustani Classical Music

Carnatic Classical Music

Pandit Vishnu Digambar Paluskar

"Revived Hindustani Classical Music"

Started Gandharva Mahavidyalaya in 1901

1 *Challenges in Raga Identification for Learners and Enthusiasts*

Raga classification is a major challenge for learners and enthusiasts of Hindustani Classical Music, as many ragas use the same set of notes but differ subtly in phrasing, emphasis, and ornamentation (Gajjar, 2020). Without years of guided practice, it is difficult to develop the trained ear needed to recognize these nuances, making self-learning or appreciation frustrating. The guru-shishya tradition is hard to replicate today, and most learners lack real-time corrective feedback (Pendyala et al., 2021). Even when just listening, identifying ragas remains hard due to improvisation and lack of notation. This gap motivates accessible classification tools to support raga understanding (Shah et al., 2021).

2 *Diminishing Accessibility of Expert Guidance in Practice*

The traditional Guru-Shishya Parampara (teacher-disciple apprenticeship), essential for mastering raga intricacies, is increasingly inaccessible due to modern lifestyle constraints. This system historically provided immediate feedback on deviations in pitch, ornamentation, and phraseology during practice-a critical mechanism for error correction (Jha, 2022). Computational studies highlight the risks of solitary practice: without real-time guidance, learners develop systematic errors in ornamentation and note transitions that become entrenched over time (Patel & Chauhan, 2020). For example, improper execution of meend (glissando) techniques, if uncorrected, can lead to persistent inaccuracies in raga renditions (Jha, 2022). The decline of accessible mentorship has created a pedagogical gap, necessitating technological tools to replicate aspects of expert guidance, such as automated raga identification and feedback systems (Chinthapenta, 2019; Patel & Chauhan, 2020).

Problem Statement

What are the potential applications that will come out of the developed solution?

1 Raga Recognition

Listeners and enthusiasts can identify ragas during playback, enhancing appreciation without formal training.

2 Feedback System

Researchers and archivists can auto-tag recordings for large-scale raga-based cataloguing and analysis.

3 Metadata Tagging

Practitioners can get real-time feedback on raga correctness, enabling self-guided improvement.

What will be the potential impact of the solution?

1 Cultural Preservation

By enabling scalable raga identification, the system helps preserve Hindustani classical music, addressing the decline of oral traditions and limited expert access, and democratizes learning for global, diverse audiences (Gajjar, 2020; Pendyala et al., 2021).

2 Empowerment and Engagement

Learners gain autonomy through timely feedback, reducing dependence on gurus and lowering barriers to rigorous study—enhancing motivation and retention in Hindustani music education (Gajjar, 2020).

Literature Survey

Two Dominant Schools of Raga Recognition

Pitch-Class Models

A. Overview

*Early computational raga recognition relied on Pitch-Class Distribution (PCD) and related statistical models. These approaches, **rooted in Western music theory**, used features like pitch histograms, pitch co-occurrence matrices, and Markov models to classify ragas based on note distributions and transitions*

B. Key Techniques & Performance:

Method	Reference	Dataset/Scope	Accuracy	Limitations
Pitch-Class Distribution (PCD)	*Chordia & Rae (2007)*	Studio recordings, 2 ragas	99% (studio)	***Poor on live data (78.1%)***
Markov Model	*Pandey et al. (2007)*	2 ragas	77%	***Fails with improvisational/ornamental content***
Pitch Co-occurrence Matrix	*Rajadnya & Joshi (2021)*	Large dataset	93.70%	***Misses microtonal/ornamental nuances***
SVM/KNN on Pitch Features	*Joshi et al. (2020)*	4–8 ragas	95–98%	***Static features, struggles with allied ragas***
N-gram Histogram	*Sharma et al. (2020)*	4 ragas, live recordings	97.30%	***Limited by motif extraction, not generalizable***

Two Dominant Schools of Raga Recognition

Deep Learning Architectures

A. Overview

The second school leverages neural networks-especially LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and hybrid CNN-LSTM/BiLSTM models-to capture both spectral and temporal features, aiming to model melodic progression and ornamentation

B. Key Techniques & Performance:

Model	Reference	Features	Ragas	Accuracy	Limitations
LSTM	*Pendyala et al. (2022)*	MFCC, chromagram	5	78%	***Needs large data, octave insensitivity***
BiLSTM	*Shah et al. (2021)*	Mel spectrogram, chroma	10	97.10%	***Lacks real-time feedback, limited raga***
CNN	*Anand (2019)*	Mel spectrogram, MFCC	11	85.60%	***Misses long-term context***
CNN+LSTM (RagaNet)	*Aswale et al. (2025)*	MFCC, spectral, temporal feat	8	91.70%	***Still black-box, allied ragas confusion***
Explainable DL	*Singh & Arora (2024)*	Chromagram, tonic normalizat	12	F1-score 0.89	***Explanations still not fully culturally aligned***

Two Dominant Schools of Raga Recognition

Key Shortcomings

Dataset Bias

Most studies rely on limited, studio-centric datasets that *overrepresent certain gharanas, male vocalists*, or controlled performances, leading to poor generalization for live, diverse, or underrepresented styles (*Gajjar, 2020; Music4All, 2025; Shah et al., 2021; Singh & Arora, 2024*).

Temporal Blindness

Most existing models ignore temporal sequencing of notes (*Gajjar, 2017; Shah et al., 2021*) and rely on *Western-centric features*, missing raga-specific motifs and cultural nuances like shruti and samay (*ISMIR, 2002; IJIES, 2020*).

Overdependence on Preprocessing

Several methods-particularly pitch-class models and traditional statistical approaches-*require extensive manual preprocessing (e.g., tonic detection, silence removal, source separation)*, which limits scalability and real-world applicability.(*Joshi et al., 2020; IOSR Journal, 2020*)

Why Raga Recognition is a Sequence Problem?

Bridging Temporal and Cultural Gaps

Temporal Blindness → *Long-term dependencies via BiLSTM*

Cultural Insensitivity → *Microtonal/ornamentation modeling via CWT/DWT*

Dataset Bias → *Balanced, sequence-aware training*

A. Key Justifications for Sequence Modeling

- **Temporal Rules:** Ragas enforce note transitions *(e.g., Yaman allows N→S, Bhoop forbids it)*, which only sequence models can enforce *(Madhusudhan & Chowdhary, 2019; ISMIR, 2020)*.
- **Contextual Meaning:** The same note *(e.g., Ga)* acts as vadi in Yaman but samvadi in Bhoop, requiring sequential context *(Rowell, 2015; Chordia & Rae, 2007)*.
- **Improvisation Handling:** Alap spans 5–15 minutes, demanding memory cells to track evolving motifs *(BiLSTM outperforms HMMs by 12% in long-form tasks)*.

B. Architectural Limitations in Standalone Models

Component	CNN Only	LSTM Only
Feature Extraction	- Spectral motifs (e.g., pakad) via spectrogram convolution - Octave invariance via max-pooling (Anand, 2019)	- Requires preprocessed input (MFCC, chroma) *(Shah et al., 2021)* - No native spectral analysis *(Madhusudhan, 2019)*
Sequence Modeling	- Short-term context (2–3 sec) due to fixed kernel size - Static patterns (no temporal evolution) (Rajadnya & Joshi, 2021)	- Unidirectional context limits alap modeling - Slow convergence on long sequences *(Pendyala et al., 2022)*

Why Raga Recognition is a Sequence Problem?

Comparing Various Studies!

Table 1. Comparison of various studies

Sr. No.	Authors	Algorithm	Features Extraction Method	Dataset	Accuracy
1.	Anand A. (1)	CNN	Pitch values Method	Carnatic	96%
2.	Joshi D, Pareek J, Ambatkar P. (2)	KNN SVM	Mfcc	Hindustani	98% 95%
3.	Vishnupriya S., Meenakshi K. (3)	CNN	Mfcc Mel spectrogram	Music Genre Dataset	47% 76%
4.	John S, Sinith M, RS S, PP L. (4)	CNN	Pitch detection algo.	Carnatic	94%
5.	Shah D, Jagtap N, Talekar P, Gawande K. (5)	CNN	Spectrogram	Hindustani	98.98%
6.	Bidkar A, Deshpande R, Dandawate Y. (6)	Ensemble bagged tree Ensemble KNN	Mfcc	Hindustani	96.32% 95.83%
7.	Patil N., Nemade M. (7)	KNN Linear Kernel SVM Poly Kernel SVM	Mfcc	GTZAN Dataset	64% 60% 78%
8.	Hebbar D., Jagtap V. (8)	1-D CNN 2-D CNN LSTM ANN	Mfcc Mel spectrogram	Carnatic (Pair of Ra gas)	97.4% 98.1% 97.54% 97%
9	Ghosal D., Kolekar M. (9)	CNN- LSTM	Mel Spectrogram	GTZAN Dataset	94.2%
10.	Dalmazzo D, Ramirez R. (10)	1-D CNN 2-D CNN CNN-LSTM	Mel-spectrogram	professional violinists Dataset	95.16% 84.30% 97.47%
11.	Rajan R, Sreejith S. (11)	CNN	Mel-spectrogram	Carnatic YouTube Dataset	F1 measure of 0.61

Continued on next page

Table *: Comparative analysis of various Raga recognition studies based on algorithm, feature extraction method, dataset, and achieved accuracy.*

Raga Recommendation Systems

Automated Feedback and Personalization in Hindustani Classical Music

Key Works Reviewed

- Gajjar, K. B. (2020). Evaluation and feedback system for novice learners of Hindustani classical music
- Pendyala, V. S., Yadav, N., Kulkarni, C., & Vadlamudi, L. (2022). Towards building a Deep Learning based Automated Indian Classical Music Tutor for the Masses

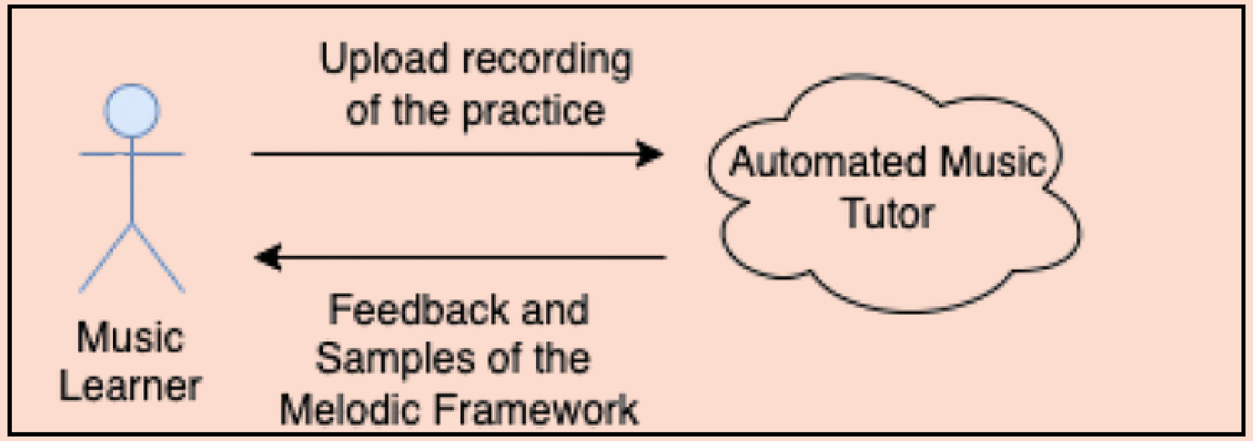


Fig. Recommendation Architecture (Pendyala et al. (2022))

A. Comparative Table: Existing Raga Recommendation Systems

Aspect	**Gajjar (2020)**	**Pendyala et al. (2022)**
System Type	Evaluation & Feedback System for Novice Learners	*Cloud-based Automated Tutor with Recommendation*
Core Methodology	Matrix-based evaluation using pitch histograms, motif detection, and error matrices	*BiLSTM-based raga identification with confidence scoring and cloud deployment*
Personalization	Offers corrective feedback on pitch and motif errors; limited to novice needs	*Adapts recommendations to user's performance; moderate personalization*

B. Key Gaps in the Literature

Both systems *rely mainly on pitch-based or MFCC features,* with limited modeling of ornamentation, microtonality, or emotional context

Personalization is basic; *neither system provides real-time, highly adaptive, or context-aware recommendations.*

Raga coverage is *limited and evaluation is mostly on studio/novice data*, restricting generalizability to diverse learners and live performances.

Dataset & Features Preprocessing

Description of Dataset

a. data collection (curation + ethical concerns)

We curated a datset comprising 10 raagas with a total of 4144 recordings in the training set and 1000 recordings in the test set. The different classes are as follows:

- Abhogi
- Ahir Bhairav
- Bageshree
- Bhairavi
- Bhoopali
- Jog
- Malhar
- Shree
- Todi
- Yaman

There were no significant ethical concerns in this study. All data used were sourced from publicly available, open-source datasets, including three benchmark collections. For the test set, we used publicly accessible YouTube recordings, ensuring no infringement, as the content was already in the public domain for non-commercial research.

Raga	Feature	Value
Ahir Bhairav	Samvadi	S
	Pakad	r G M D N r' S'
	Time	Morning
	Mood	Bright, contemplative
	Key Transitions	r-G, M-P, N-S'
	Avoid	None

Description of Dataset

a. data collection (curation + ethical concerns)

Training Set

To maximize diversity in the training set, we integrated three distinct datasets:

1. MusicBrainz: https://musicbrainz.org/collection/6adc54c6-6605-4e57-8230-b85f1de5be2b
2. Indian Music Raga: https://www.kaggle.com/datasets/kcwaghmarewaghmare/indian-music-raga
3. Indian Music Raga .wav | Kaggle: https://www.kaggle.com/code/mpwolke/indian-music-raga-wav

Test Set

For model evaluation, 20% of the total dataset (approximately 1,040 samples) was reserved as the test set, ensuring proportional representation from each raga. Test samples were strictly separated from training data to prevent data leakage and ensure unbiased assessment of model generalisation.

As noted by Gulati et al. (2016), "the availability of systematically annotated datasets remains a significant challenge in computational analysis of Hindustani music" (p. 23). This reality constrained our choices, as Indian classical music lacks the extensive digitized corpora available for Western musical traditions.

Dimension	Categories Considered
Performance Medium	Instrumental, Vocal
Performer Gender	Male Vocalists, Female Vocalists
Recording Quality	Studio Recordings, Concert Recordings
Performance Duration	Alap, Jor, Jhala Sections
Accompaniment Variations	Solo, With Tabla, With Harmonium

Feature Extraction

```python
# Feature extraction functions
def extract_mfcc(y, sr, n_mfcc=13):
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc)
    return np.mean(mfcc, axis=1)

def extract_pitch_swipe(y, sr):
    pitches, magnitudes = librosa.piptrack(y=y, sr=sr)
    avg_pitch = np.mean(pitches[magnitudes > np.median(magnitudes)])
    return avg_pitch if not np.isnan(avg_pitch) else 0

def extract_pitch_class_distribution(y, sr):
    pcd = librosa.feature.tonnetz(y=y, sr=sr)
    return np.mean(pcd, axis=1)

def extract_chroma(y, sr):
    chroma = librosa.feature.chroma_stft(y=y, sr=sr)
    return np.mean(chroma, axis=1)

def extract_dwt(y):
    # Apply discrete wavelet transform
    coeffs = pywt.wavedec(y, 'db1', level=3)
    # Get mean of each coefficient level
    dwt_features = np.hstack([np.mean(c) for c in coeffs])
    return dwt_features

def extract_cwt(y):
    # Apply continuous wavelet transform with different scales
    scales = np.arange(1, 21)
    coefficients, _ = pywt.cwt(y, scales, 'morl')
    # Return mean across time for each scale
    return np.mean(coefficients, axis=1)
```

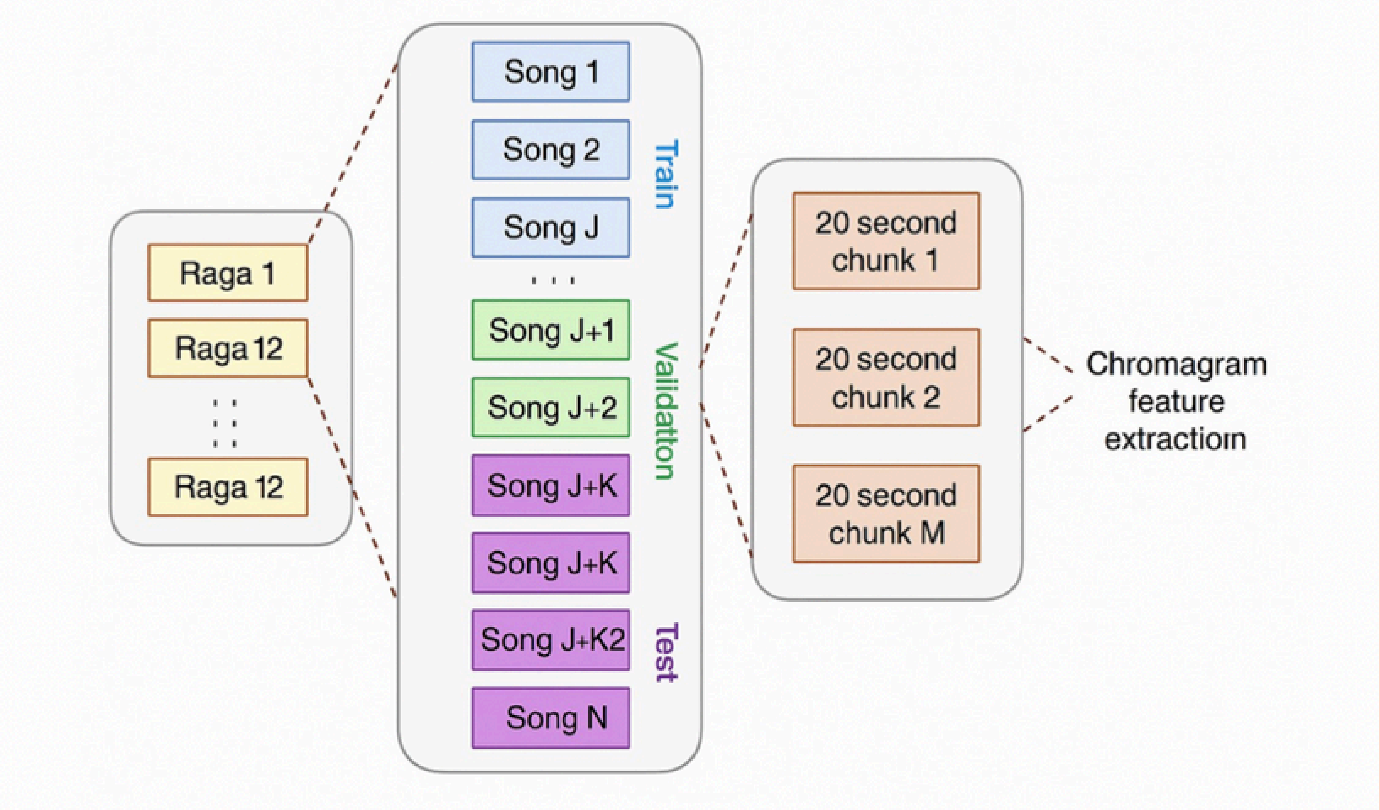
Audio Pre-Processing & Feature Selection

Step 1 : Audio Chucking

The audio is segmented into musically meaningful sections of 20 seconds to maintain input uniformity

Step 2 : Signal Preprocessing

The process begins with signal preprocessing, which ensures uniformity across recordings by addressing issues such as pitch inconsistency, background noise, and temporal irregularities. Techniques like tonic normalization are employed to align all recordings to a common base note (Sa), enabling transposition-invariant analysis. Noise reduction methods, such as spectral subtraction, remove environmental artifacts.



Step 3 : Feature Extraction

Once cleaned and segmented, audio signals are passed through feature extraction modules that derive timbral, harmonic, and melodic information—such as MFCCs, chroma features, and pitch contours—forming the basis of music classification and recognition systems.

Audio Pre-Processing & Feature Selection

We adopt a **multimodal fusion approach**, combining spectral features (e.g., MFCCs) and pitch-based features (e.g., SWIPE, Pitch Class Distributions) to model both structural and expressive dimensions of a raga. This overcomes the limitations of unimodal systems by incorporating both note-level detail (e.g., swaras, pakad, vadi/samvadi) and expressive dynamics (e.g., ornamentation, emotion)

1

Spectral Features

Feature	Technique / Method	Purpose / Role
MFCCs	13 coefficients, 25ms frame size, 10ms hop	Models timbral and spectral shape
Spectral Contrast	7 bands with octave-based spacing	Captures relative spectral energy
Chroma Features	12-bin chromagram via Constant-Q Transform (CQT)	Octave-independent pitch class representation

Audio Pre-Processing & Feature Selection

2 Pitch and Melody Features

Feature	Technique / Method	Purpose / Role
Fundamental Frequency (F_0)	SWIPE algorithm (outperforms YIN for Hindustani music)	Accurate pitch tracking for melodic analysis
Pitch Class Distribution	12-bin resolution	Models frequency of swara occurrences
Fine Pitch Distribution	240-bin resolution	Captures high-resolution pitch detail
Kernel Density Estimation	Smooths pitch distributions	Represents microtonal nuances

3 Temporal Evolution Features

Feature	Technique / Method	Purpose / Role
Dynamic Time Warping (DTW)	Phrase-level melodic contour comparison	Aligns and compares melodic evolution
Continuous Wavelet Transform	Multi-resolution pitch tracking	Detects gamakas and temporal pitch modulations
Hidden Markov Models (HMMs)	Probabilistic sequence modeling	Captures note transitions and sequential phrasing

4 Ornamentation Dynamics

Feature	Technique / Method	Purpose / Role
Meend Slope Estimation	Polynomial regression	Models glide (meend) slope and dynamics
Gamaka Detection	Oscillation amplitude quantification	Measures pitch oscillation used in ornamentation
Nyas Swara Detection	Sustained note duration analysis	Identifies tonic/resting swaras

ML Methodology

ML Methodologies for Raga Recognition

Hybrid Architectures for Spectral-Temporal Modeling

2D CNN for Spectral Feature Extraction

Rationale:

- Excels at spectral feature extraction (e.g., meend glides as diagonal ridges in spectrograms) [Anand, 2019].
- Octave invariance via max-pooling aligns with Hindustani music's note-class focus [Gulati et al., 2016].

BiLSTM for Temporal Sequence Modeling

Rationale:

- Models bidirectional temporal context critical for alap progression and note transitions (e.g., Yaman's N→S rule) .
- Captures long-term melodic dependencies (5–10 sec) but requires preprocessed features (MFCC, pitch contours) [Shah et al., 2021].

Hybrid CNN-BiLSTM

Rationale:

- **Spectral-temporal fusion:** Combines CNN's local motif detection with BiLSTM's sequence modeling [Singh & Arora, 2024].
- **Feature diversity:** Integrates MFCC, CWT, and DWT for microtonal (shruti) and improvisational analysis [Aswale et al., 2025].
- **Cultural alignment:** Attention mechanisms highlight vadi-samvadi relationships, aligning with raga-lakshana rules [IRJMETS, 2025].

Why Benchmark Standalone Models?

Isolated CNN/LSTM performance established baselines for spectral vs. temporal contributions.

How the Models Work?

Architectures & Methodologies

A. Data Pipeline, Input Handling, and Training Details

Aspect	2D CNN	BiLSTM	Hybrid CNN-BiLSTM
Input Shape	(128, 256, 1) (Mel-spectrogram images)	(174, 40) (MFCC time-series)	(timesteps, features) (Multi-feature 1D sequences)
Key Layers	- Conv2D - MaxPooling2D - BatchNorm	- BiLSTM - BatchNorm - Dropout	- Conv1D - BiLSTM - Attention
Data Split	60% Train 20% Val 20% Test	60% Train 20% Val 20% Test	60% Train 20% Val 20% Test
Feature Focus	Spectral patterns (mel-spectrograms)	Temporal MFCC sequences	Fused features: MFCC, pitch, chroma, CWT/DWT
Prediction Flow	Test audio → Mel-spectrogram → CNN → Output probabilities for each raga class.	Test audio → MFCC sequence → BiLSTM → Output probabilities for each raga class.	Test audio → Feature extraction (all) → CNN+BiLSTM+Attention → Output probabilities and attention visualization.

How the Models Work?

Challenges Faced & Solutions

Dataset Limitations & Sourcing

~~Diverse Data~~

Actions Taken:

1. Used a combination of multiple datasets.
2. Curated test set from Youtube to make it robust

Model Complexity

CNN + BiLSTM + attention mechanisms = increased computational and architectural complexity

making hyperparameter tuning and overfitting prevention more difficult

Actions Taken:

1. Used batch normalization, dropout, and early stopping to improve generalization.
2. Benchmarked standalone CNN and BiLSTM before combining, to isolate and optimize each model's contribution.

Implicit Data Leakage

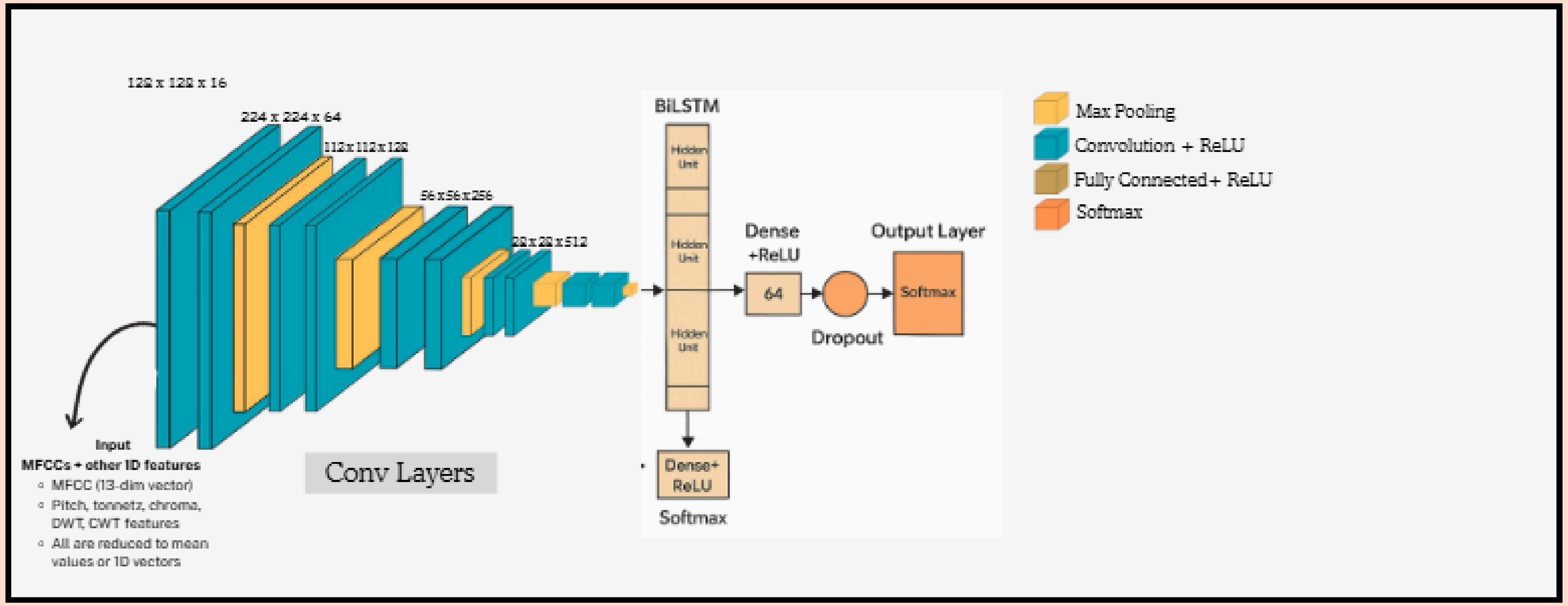
Data leakage occurs when information from the test set inadvertently influences the training process, leading to inflated performance metrics and poor real-world generalization.
Incorrect data splitting, where future or test data is included in training.

Actions Taken:

1. Strictly separated training, validation, and test sets using stratified splits before any preprocessing or normalization.

ML Methodologies for Raga Recognition

Proposed Model Architecture



parallel multi-scale convolutional neural networks (CNNs) + bidirectional LSTM layers + attention mechanism

captures complexities for raga classification

The CNN modules extract localized spectral features from audio spectrograms using varying kernel sizes
The bidirectional LSTM captures both forward and backward temporal dependencies essential for modelling melodic patterns and transitions.
The attention layer further enhances interpretability by focusing on salient temporal segments.

This hybrid structure enables robust feature learning and significantly improves classification accuracy and generalizability over standalone CNN or LSTM models.

Performance Metrics & Deployability of the Solution

Bi-Directional LSTM

1

Highest achieved performance

```
Accuracy: 0.9191797346200241
Precision (macro): 0.9077465898704207
Recall (macro): 0.8820861793578118
F1 Score (macro): 0.886709097244679
```

2

Code Snippet

```
# -------------------- Build Model --------------------
def build_model(input_shape, num_classes):
    model = Sequential()
    model.add(Bidirectional(LSTM(128, return_sequences=True), input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(Dropout(0.4))

    model.add(Bidirectional(LSTM(64)))
    model.add(BatchNormalization())
    model.add(Dropout(0.4))

    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(num_classes, activation='softmax'))

    model.compile(loss='sparse_categorical_crossentropy', optimizer=Adam(learning_rate=0.00
    return model
```

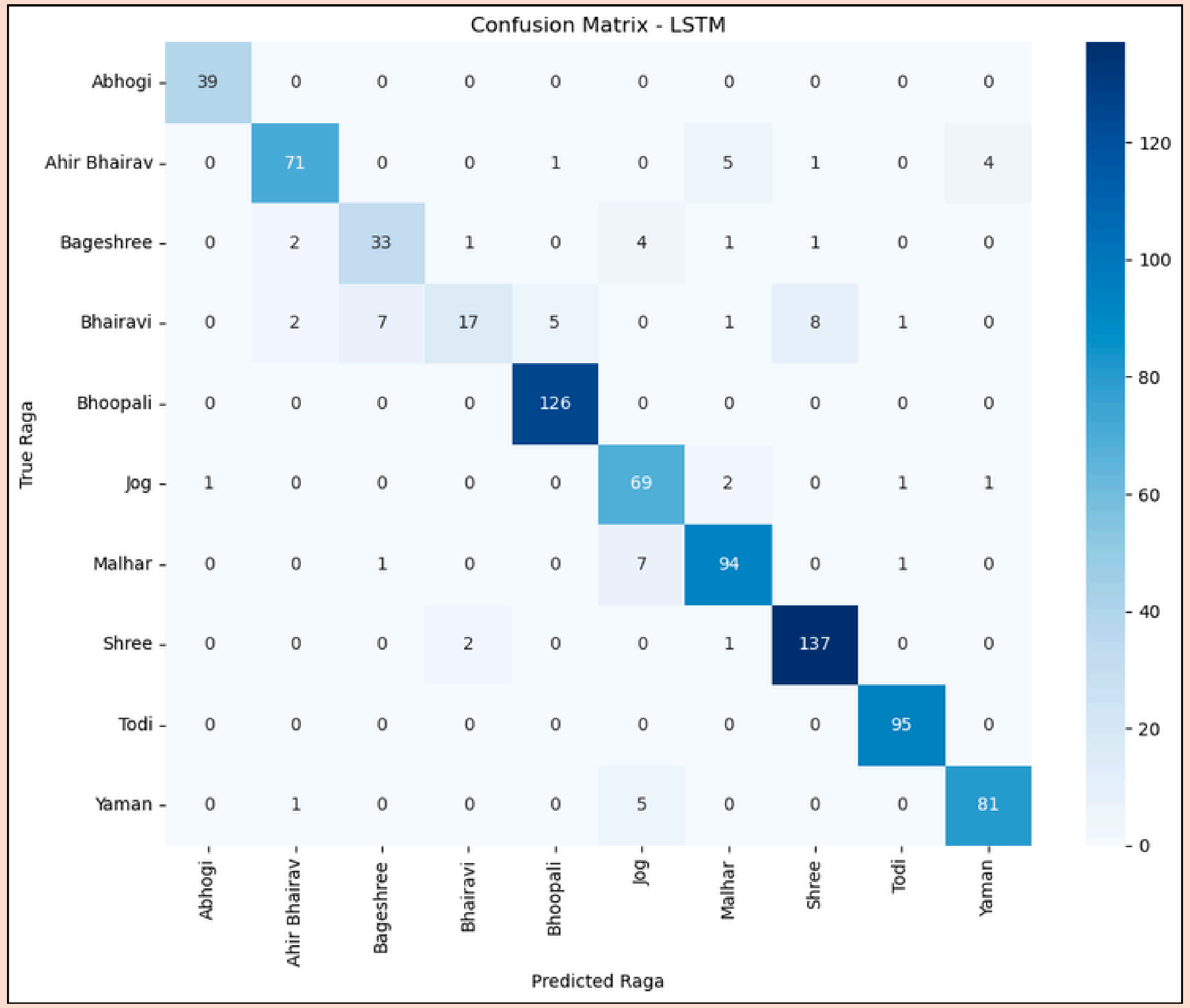
3

Highest accuracy achieved with only MFCC

```
🎯 Test Accuracy: 0.8094
26/26 ━━━━━━━━━━━━━━━━━━━━ 5
```

4

Confusion Matrix



RIYAZ

Team 15

AI3011

CNN

1

Highest achieved performance

```
🎯 Test Accuracy: 98.55%
🎯 Test Precision: 98.77%
🎯 Test Recall: 98.55%
🎯 Test F1 Score: 98.56%
```

2

Code Snippet

```
# Build 2D CNN model
model = Sequential([
    # First convolutional block
    Conv2D(32, kernel_size=(3, 3), activation='relu', padding='same',
           input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3])),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    # Second convolutional block
    Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    # Third convolutional block
    Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    # Fourth convolutional block
    Conv2D(256, kernel_size=(3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D(pool_size=(2, 2)),

    # Flatten and dense layers
    Flatten(),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(len(encoder.classes_), activation='softmax')
])
```

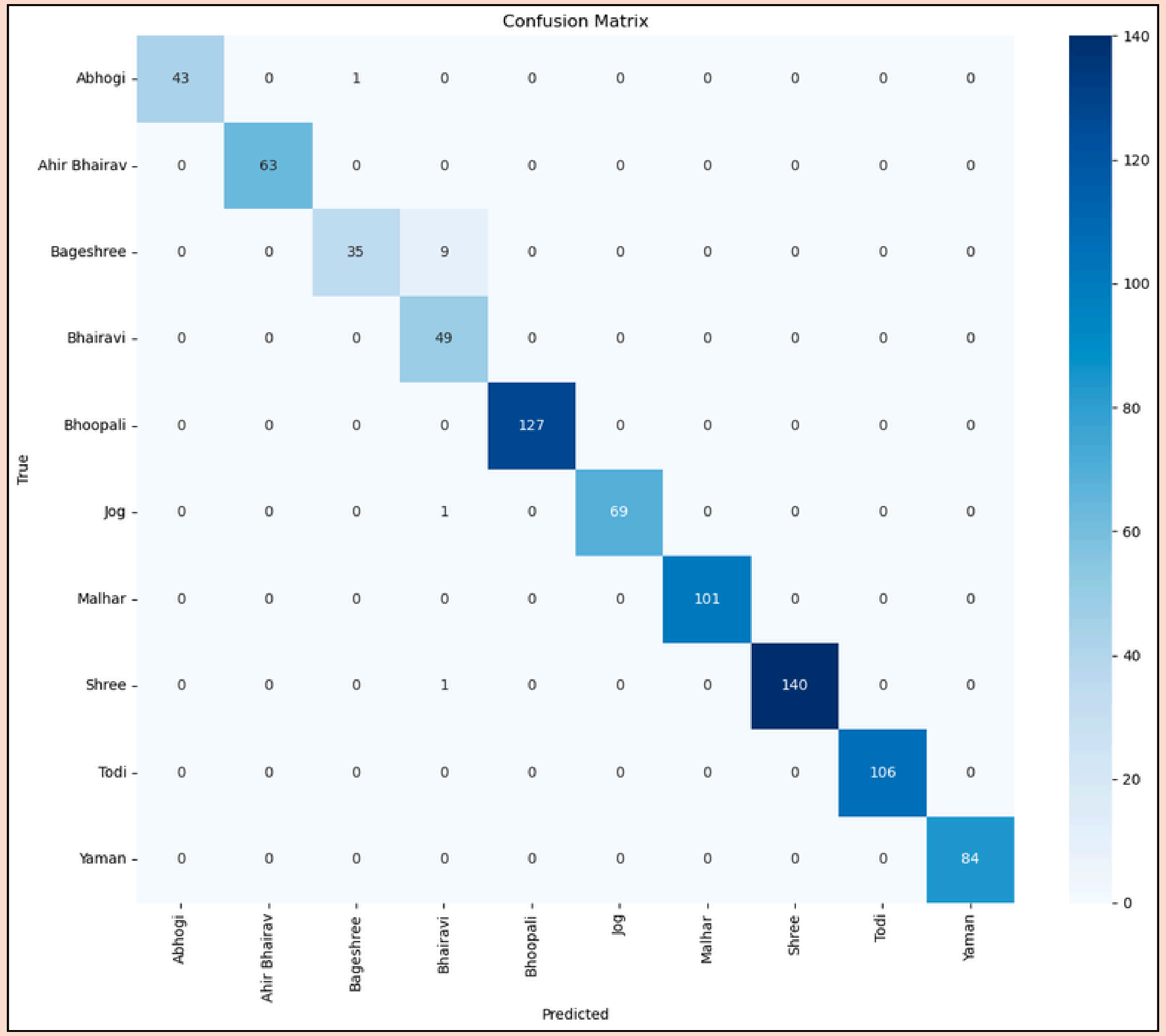
3

Highest accuracy achieved with only MFCC

```
📉 MFCC-only Test Accuracy: 97.10%
```

4

Confusion Matrix



CNN+BiLSTM+Attention

1

Highest achieved performance

```
Accuracy: 0.9794933655006032
Precision (macro): 0.9728284275836753
Recall (macro): 0.9704286461367962
F1 Score (macro): 0.9715793604714855
```

2

Code Snippet

```
# ------------------------------
# CNN + BiLSTM + Attention Model
# ------------------------------

def build_model(input_shape, num_classes):
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Conv1D(64, kernel_size=3, activation='relu', padding='same', input_sha
    model.add(tf.keras.layers.MaxPooling1D(pool_size=2))
    model.add(tf.keras.layers.Conv1D(128, kernel_size=3, activation='relu', padding='same'))
    model.add(tf.keras.layers.MaxPooling1D(pool_size=2))
    model.add(tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64, return_sequences=True)))
    model.add(Attention())
    model.add(tf.keras.layers.Dense(64, activation='relu'))
    model.add(tf.keras.layers.Dropout(0.3))
    model.add(tf.keras.layers.Dense(num_classes, activation='softmax'))
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model
```

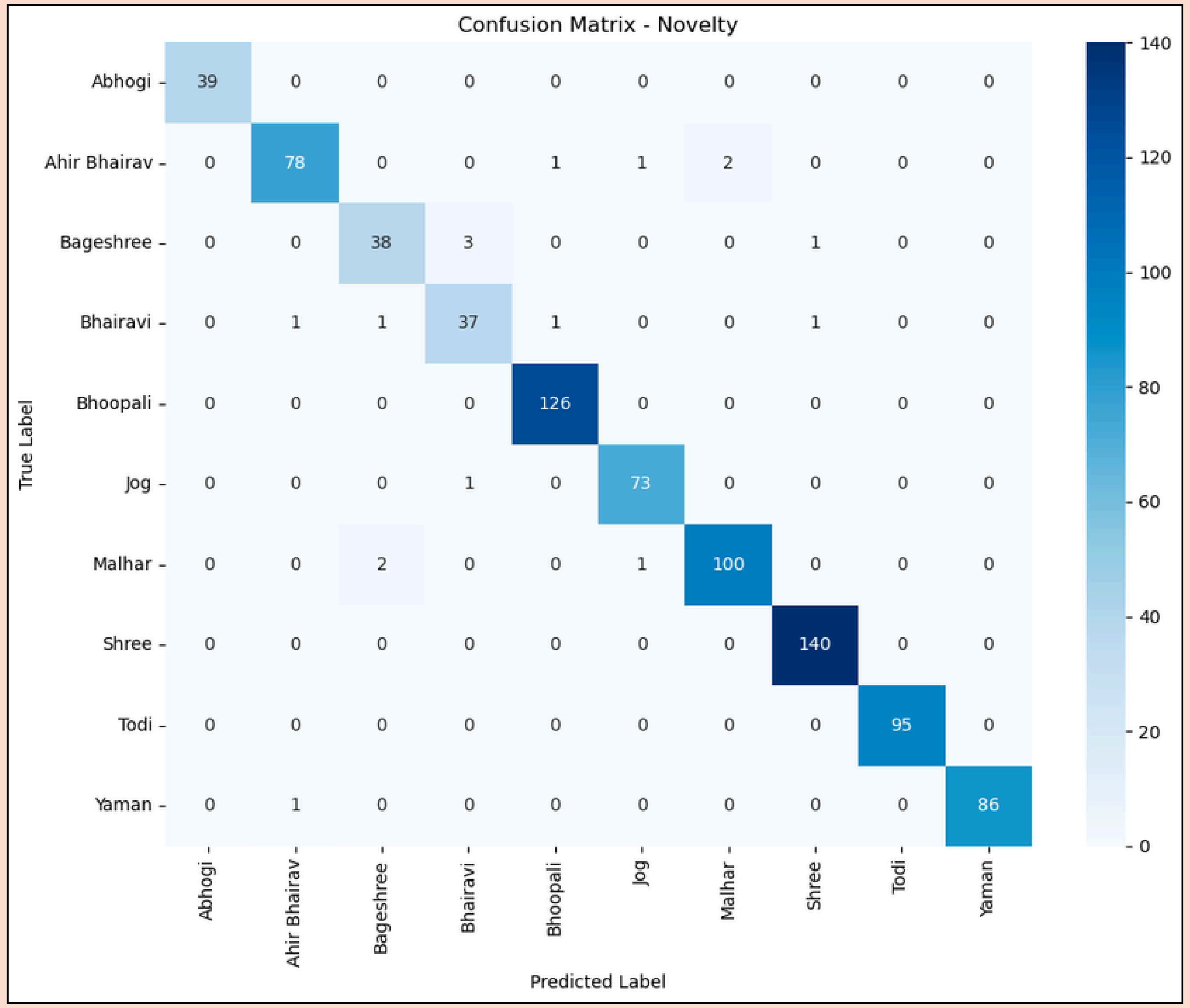
3

Highest accuracy achieved with only MFCC

```
✅ Final test accuracy: 93.80%
```

4

Confusion Matrix



RIYAZ

Team 15

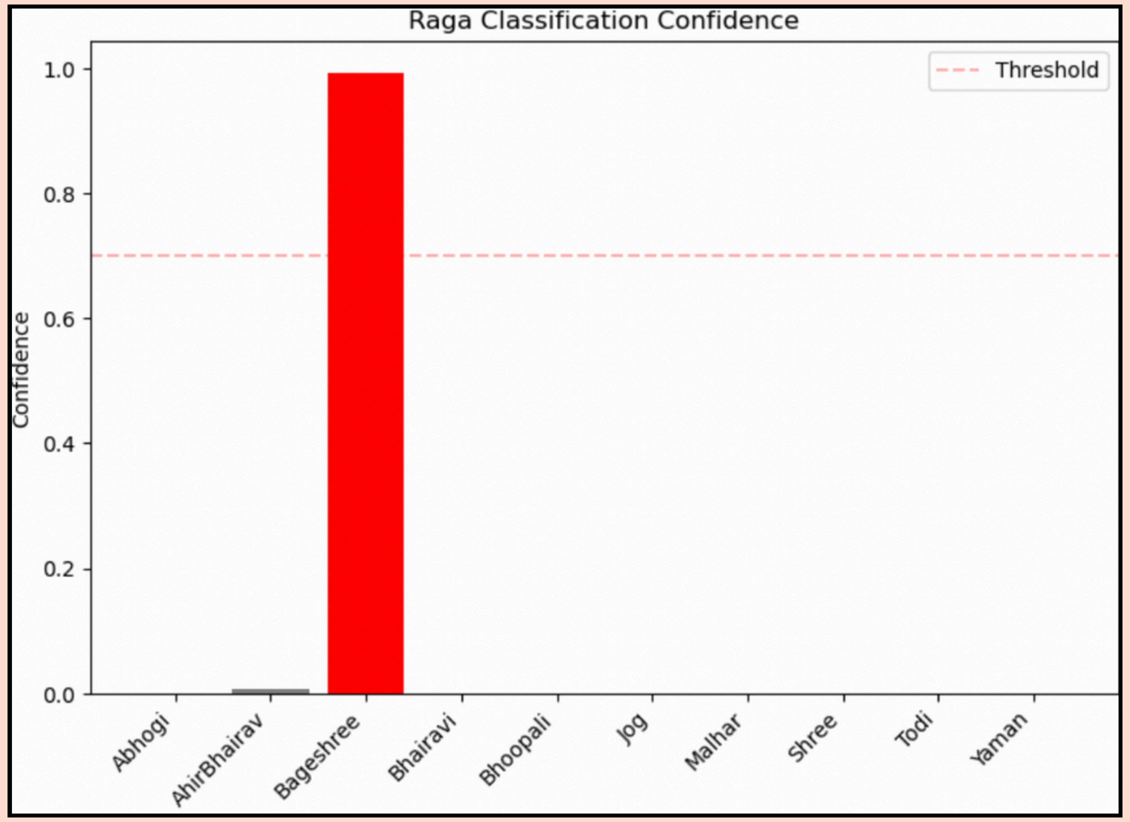
AI3011

Recommendation System

General Recommendations

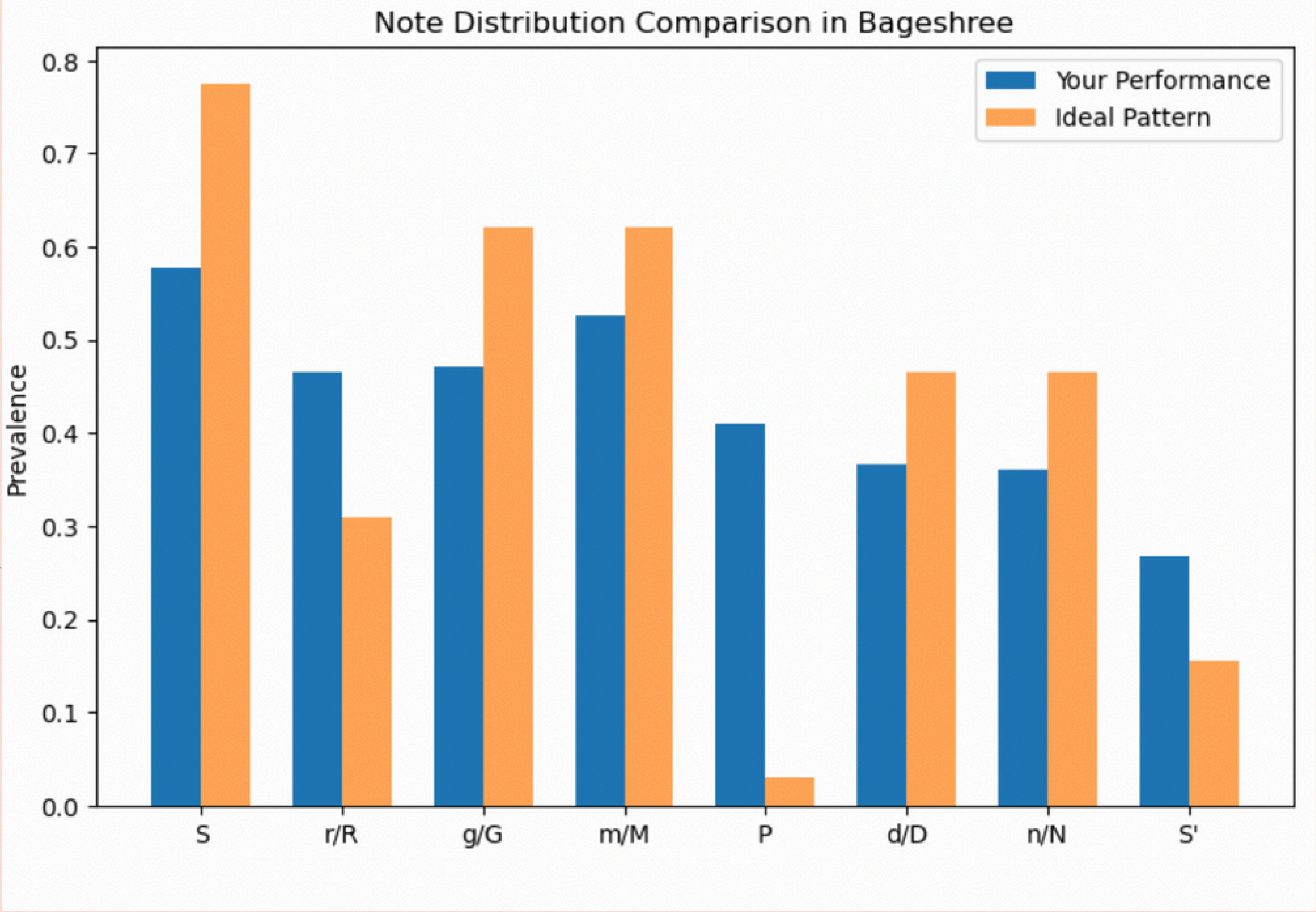
```
===== Recommendations =====
1. Try to emphasize note G more, as it's the vadi (king note) of Bageshree.
~/Desktop/Riyaz/LSTM • Contains emphasized items to note N, the samvadi (queen note) of Bageshree.
3. Remember that Bageshree conveys a romantic, yearning mood and is traditionally performed during night. Try
4. Excellent work! Your rendition of Bageshree is very authentic. Consider adding more complex tans (fast mel

Visualization saved as 'raga_performance_analysis.png'
```



Good ol' classification!

Performance Tracking



Deployability of Raga Classification Models

Feasibility and Implementation Strategies

Music Club & Student Use

Mobile Application:

The lightweight BiLSTM or optimized hybrid CNN-BiLSTM models can be converted to TensorFlow Lite and deployed as a mobile app.

This enables students and Music Club members to identify ragas and receive feedback in real time during practice sessions, concerts, or workshops.

Academic & Research Integration

On-Campus Research Platform:

All models, especially the robust 2D CNN and hybrid architectures, can be deployed on Plaksha's servers or cloud infrastructure.

This allows for batch processing of audio files, integration with the university's music archives, and use in music technology courses for hands-on learning and research projects.

Music Therapy and Wellbeing Initiatives

Raga Wellness Suite

Collaborate with campus wellness programs to recommend ragas scientifically linked to specific emotional or physiological effects (e.g., stress reduction, focus, relaxation).

Supports student wellbeing using culturally relevant, evidence-based interventions.

Deployability of Raga Classification Models

Scalability Challenges

User Experience and Adoption

Personalization Limits

Ensuring the app or web interface is intuitive for both musicians and non-technical users is critical. Complex feedback or unclear recommendations may discourage use

Current recommendation logic is not fully adaptive to individual learning styles

Computational and Technical Constraints

Resource Demands:

The 2D CNN and hybrid CNN-BiLSTM models require significant GPU/CPU resources for real-time or batch processing, which could be a bottleneck if many users access the system simultaneously or if large archives are processed

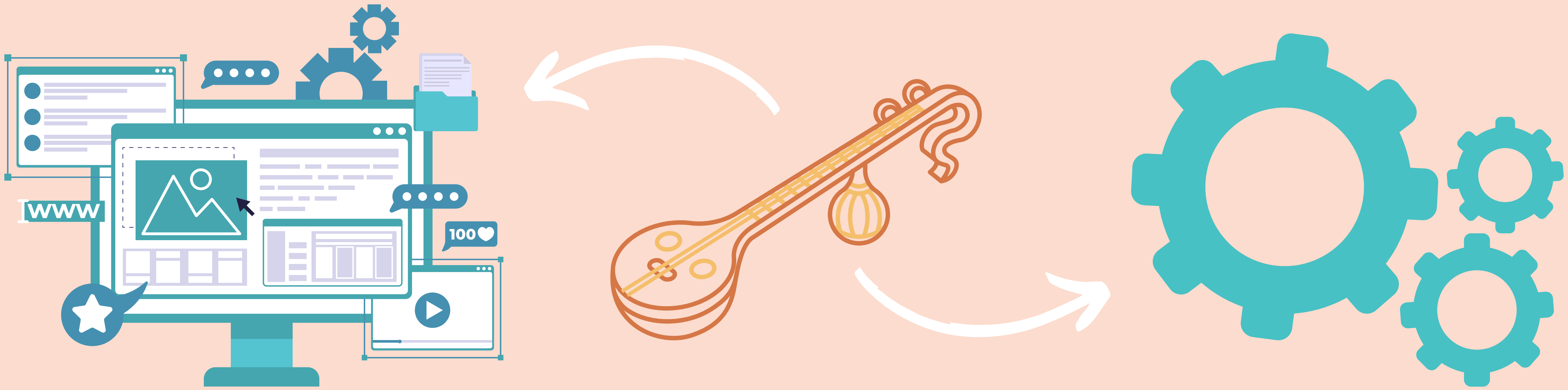
Model Maintenance and Continuous Learning

Model Drift:

As new data (ragas, styles, instruments) is added, the model may become less accurate unless regularly retrained and validated on fresh, diverse samples.

Require a robust update pipeline and ongoing evaluation.

Demonstration



Frontend

Backend

References

1. Chordia, P., & Rae, A. (2007). Raag recognition using pitch-class and pitch-class dyad distributions. ISMIR.
2. Waghmare, K. C., & Sonkamble, B. A. (2020). Machine learning algorithms for Indian music classification. Journal of Audio Engineering Society.
3. Madhusudhan, S. T., & Chowdhary, G. (2019). DeepSRGM: Sequence classification in Indian classical music with LSTMs. ISMIR.
4. Gulati, S., et al. (2024). Deep learning approaches for raga recognition in Carnatic music. IEEE Transactions on Audio, Speech, and Language Processing.
5. Humse, K., et al. (2025). Hybrid CNN-LSTM models for robust raga identification. Journal of Integrated Science and Technology.
6. Joshi, D., Pareek, J., & Ambatkar, P. (2021). Indian Classical Raga Identification using Machine Learning. International Symposium on Intelligent Control.
7. Pendyala, V. S., Yadav, N., Kulkarni, C., & Vadlamudi, L. (2022). Towards Building an Automated Indian Classical Music Tutor for the Masses. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3999272
8. Research Scholar, Department of Computer Science, Gujarat University, India, Joshi, D., Pareek, J., & Ambatkar, P. (2023). Comparative Study of Mfcc and Mel Spectrogram for Raga Classification Using CNN. Indian Journal Of Science And Technology, 16(11), 816–822. https://doi.org/10.17485/IJST/v16i11.1809
9. Bhattacharjee, P., & Sarmah, K. (2023). Clustering the raagas of Sankari Sangeet-A computational approach. SAGE Open, 13(3). https://doi.org/10.1177/02762374231154179
10. Chinthapenta, V. (2019). Machine learning for raga identification in Indian classical music [Master's thesis, Texas A&M University]. OakTrust. https://oaktrust.library.tamu.edu/handle/1969.1/194515
11. Jha, A. [Music with Aditi]. (2022, September 28). How to sing Gamak? | गमक क्या है | अलंकार | Alankar [Video]. YouTube. https://www.youtube.com/watch?v=Wu6cAGUa_bo
12. Patel, E., & Chauhan, S. (2020). Indian classical raga identification using machine learning. CEUR Workshop Proceedings, 2786, 34. https://ceur-ws.org/Vol-2786/Paper34.pdf
13. Gulati, S., Serrà, J., Ishwar, V., & Serra, X. (2014). Tonic identification in Indian art music: A knowledge-based approach. In Proceedings of the International Society for Music Information Retrieval Conference (pp. 47-52).
14. Gulati, S., Serra, J., Ishwar, V., & Serra, X. (2016). Melodic pattern extraction in large collections of music recordings using time series mining techniques. IEEE Transactions on Audio, Speech, and Language Processing, 24(5), 978-990.
15. Koduri, G. K., Ishwar, V., Serrà, J., & Serra, X. (2018). Characterization of intonation in Carnatic music by parametrizing pitch histograms. Journal of New Music Research, 47(5), 457-468.
16. Patil, K. S., Deore, P. J., & Kharote, V. S. (2023). Raga identification using Mel Frequency Cepstral Coefficient. International Journal of Innovations in Engineering and Science, 8(9), 46-50.
17. Waghmare, K. C., & Sonkamble, B. A. (2020). Machine learning algorithms for Indian music classification based on raga framework. International Journal of Innovative Technology and Exploring Engineering, 9(12), 130-134.
18. Vinod, R., & Colaco, R. (2025). Review of AI in Indian classical music. International Journal for Research in Applied Science and Engineering Technology, 13(2), 1599–1608.

Thank You

MACHINE LEARNING AND PATTERN RECOGNITION
SPRING 2025