



Hybrid Acoustic Feature Fusion and Stacking Ensemble Learning for Multilingual Spoken Indian Language Recognition on Edge Devices

Shreyasi Watve^{1,2*} , Mangal Patil¹ 

¹ Department of Electronics and Communication Engineering, Bharati Vidyapeeth Deemed to be University College of Engineering, Pune 411043, India

² Department of Electronics and Telecommunication Engineering, P. E. S. Modern College of Engineering, Pune 411005, India

Corresponding Author Email: shreyasi.watve@moderncoe.edu.in

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310923>

ABSTRACT

Received: 23 May 2026

Revised: 18 August 2026

Accepted: 9 September 2026

Available online: 30 September 2026

Keywords:

multilingual spoken language recognition, acoustic feature fusion, Mel-Frequency Cepstral Coefficients-Chroma representation, stacking ensemble learning, machine learning, edge computing

Automatic spoken language recognition remains challenging for multilingual environments due to variations in phonetic characteristics, speaker diversity, and acoustic conditions. This study proposes a lightweight recognition framework that combines hybrid acoustic feature fusion with stacking ensemble learning for multilingual Indian language identification. The proposed framework investigates the complementary characteristics of Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features by constructing a fused MFCC-Chroma representation for improved acoustic discrimination. Three heterogeneous classifiers, including Support Vector Machine (SVM), Random Forest (RF), and Extreme Gradient Boosting (XGBoost), are employed as base learners, while Logistic Regression (LR) is used as a meta-classifier in the stacking ensemble framework. Experiments are conducted on approximately 14,000 speech samples from the Mozilla Common Voice corpus covering six Indian languages: Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam. The experimental results demonstrate that the fused MFCC-Chroma representation consistently outperforms individual acoustic features. The proposed stacking ensemble achieves an accuracy of 89.18% with a 95% confidence interval of 87.97%–90.28% for six-language recognition. Furthermore, selected lightweight classifiers are deployed on a Raspberry Pi 4 Model B platform with a Gradio interface to evaluate their feasibility for edge-based speech recognition. The results indicate that the proposed feature fusion and ensemble learning framework provides an effective and computationally practical solution for multilingual spoken language recognition in resource-constrained environments.

1. INTRODUCTION

India is one of the most linguistically diverse countries in the world, with a large number of languages and dialects belonging to different language families including the Indo-Aryan, Dravidian, Tibeto-Burman and Austroasiatic groups. Some of them are Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam. These are widely spoken languages with different linguistic and phonological features. Despite this diversity, many Indian languages remain relatively underrepresented in contemporary speech and voice technologies. Therefore, developing accurate spoken Indian language recognition systems is essential for enabling inclusive multilingual voice-based applications. Such applications are particularly important in education, public services, human-computer interaction, and digital accessibility.

Automatic spoken Indian language recognition aims to identify the language being spoken from an audio signal without requiring the content of the speech to be explicitly transcribed. But there are many challenges in developing reliable language recognition systems for Indian languages. Some languages have similar phonetics and linguistic properties, which implies that their corresponding acoustic

patterns may be overlapped which makes a proper classification a challenge. Differences in speaker characteristics, pronunciation, recording conditions, background noise, and the availability of annotated speech data sufficient for training can affect recognition performance. These challenges motivate the development of robust acoustic feature representations and machine-learning models that can discriminate subtle differences among languages.

Acoustic feature extraction plays a crucial role in spoken Indian language recognition. Mel-Frequency Cepstral Coefficients (MFCCs) are widely used to represent the spectral characteristics of speech and capture information related to the vocal tract and timbre properties of an audio signal. Chroma features, in contrast, encode the distribution of spectral energy over twelve pitch classes and can provide complementary information related to harmonic characteristics. In this study, MFCC and Chroma features are investigated individually as well as in a hybrid representation. Specifically, 13-dimensional MFCC features and 12-dimensional Chroma features are combined to form a 25-dimensional MFCC-Chroma feature vector. The hybrid representation aims at capturing complementary spectral and harmonic characteristics that might help to discriminate

acoustically similar Indian languages.

To improve classification performance, this study employs multiple machine-learning classifiers within a stacking-based ensemble framework. Base classifiers are Support Vector Machine (SVM), Random Forest (RF), and XGBoost, and the meta-classifier is Logistic Regression (LR). The base classifiers learn complementary decision patterns on the fused acoustic feature representation, and their prediction outputs are combined by the LR meta-classifier for the final language prediction. The ensemble scheme is designed to take advantage of the complementary merits of various learning algorithms and to enhance the robustness and generalization ability of the language recognition system in comparison with single classifiers.

The experimental evaluation is conducted using the primary dataset. The primary dataset consists of approximately 14,000 audio samples obtained from the Mozilla Common Voice corpus, covering six Indian languages: Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam. The audio data are preprocessed and transformed into the required acoustic representations before being used for model training and evaluation. The performance of MFCC, Chroma, and hybrid MFCC-Chroma representations is compared across the individual classifiers and the proposed stacking ensemble. This study presents a systematic evaluation framework for multilingual Indian spoken language recognition by investigating individual and hybrid acoustic feature representations in combination with multiple machine-learning classifiers and ensemble learning strategies.

In addition to performance evaluation, practical edge deployment is considered to demonstrate the applicability of the developed system in resource-constrained environments. The SVM and RF models are independently deployed on a Raspberry Pi using a Gradio-based graphical interface for on-device language inference. The ensemble model is evaluated separately and is not deployed on the Raspberry Pi. The deployment demonstrates the feasibility of using lightweight machine-learning models for speech-based language recognition at the edge, where computational and resource constraints may limit the use of computationally intensive architectures. The main contributions of this work are therefore:

- A systematic comparative evaluation of individual acoustic features, namely MFCC, Chroma, and hybrid features for multilingual Indian spoken language recognition.
- Investigation of a hybrid MFCC–Chroma feature representation for capturing complementary acoustic information.
- A comparative assessment of multiple machine-learning classifiers, including SVM, RF, XGBoost, and LR, followed by the evaluation of a stacking-based ensemble strategy.
- Raspberry Pi deployment of SVM and RF models to demonstrate the feasibility of edge-based spoken Indian language recognition.

The proposed framework aims to contribute toward robust, efficient, and inclusive multilingual speech technologies suitable for the diverse linguistic landscape of India.

2. RELATED WORK

Automatic spoken Indian language recognition aims to determine the language of an utterance from its acoustic characteristics. Most of the early Language Identification

(LID) systems are based on statistical and discriminative methods. Zissman [1] compared several approaches for automatic language identification of telephone speech. Torres-Carrasquillo et al. [2] studied GMM- and SDC feature-based language recognition. Later, the i-vector framework provided compact representations for language recognition and better modeling of speech variability [3]. These studies were the foundation for statistical and machine-learning methods for LID. The choice of acoustic features strongly influences language recognition performance. MFCCs are among the most widely used speech features because they characterize the spectral envelope of speech. Davis and Mermelstein [4] proposed parametric representations of speech based on the cepstrum, and later work showed that representations based on MFCCs were effective for identifying spoken language [5, 6]. However, MFCCs mainly represent the spectral-envelope information and may not describe the complementary harmonic characteristics. Chroma features, which capture the distribution of spectral energy over pitch classes, have also proven useful for speech-related classification. Chittaragi and Koolagudi [7] investigated dialect identification using Chroma-based and spectral shape acoustic representations combined with an ensemble learning strategy. Although their work addresses dialect discrimination rather than multilingual language recognition, it demonstrates the usefulness of complementary spectral and tonal acoustic information for speech classification tasks.

Spoken Indian language recognition presents additional challenges due to substantial linguistic, phonetic, and acoustic diversity. Rao and Koolagudi [8] have worked on Hindi dialect identification using spectral and prosodic features. Ramu Reddy et al. [9] worked on spectral and prosodic features for Indian spoken language identification. Dey et al. [10] reviewed spoken Indian language recognition extensively and highlighted the challenges of linguistic diversity, limited resources, and speaker variability. Such studies emphasize the need for robust acoustic representations that can distinguish between several Indian languages.

Machine-learning methods remain attractive for speech classification because they can provide effective performance with comparatively moderate computational requirements. SVMs are particularly suitable for high-dimensional classification problems [11]. SVM-based approaches have proved their applicability for closely related Indo-Aryan languages [12]. RF [13], proposed by Breiman, is a strong ensemble classifier based on the combination of multiple decision tree models. In addition to SVM and RF, XGBoost is applied in the present study as a gradient-boosting classifier [14]. The use of these heterogeneous classifiers allows to comparatively evaluate performance of the proposed feature representations in different learning paradigms.

Ensemble learning can further improve classification by combining the complementary predictions of multiple models. Dietterich [15] discussed ensemble learning as an effective approach for combining multiple classifiers, while Wolpert [16] introduced stacked generalization, in which predictions from base learners are provided to a higher-level meta-classifier. Motivated by these approaches, the present study employs SVM, RF, and XGBoost as base classifiers and LR as the meta-classifier in a stacking framework.

Deep-learning and self-supervised speech representations have also achieved substantial progress in multilingual speech processing. Deep neural networks have been widely investigated for acoustic modeling [17], while wav2vec 2.0

introduced self-supervised speech representation learning from unlabeled audio [18]. XLS-R further extended self-supervised learning to cross-lingual speech representation learning [19]. Although these approaches provide powerful representations, their computational requirements can be relatively high for resource-constrained edge devices. Therefore, the present study focuses on conventional acoustic features and machine-learning models to achieve a balance between recognition performance and computational feasibility.

Finally, edge deployment is important for practical multilingual speech applications. Mondal and Das Barman [20] proposed a deep-learning-based real-time audio-event detection system in which audio features were extracted using a Raspberry Pi edge device and transmitted to a local server for classification. Although the application was not specifically focused on language identification, the study provides relevant evidence regarding the feasibility and computational considerations associated with deploying audio-processing systems on resource-constrained embedded hardware. In this work, SVM and RF are selected for Raspberry Pi deployment, while XGBoost and the complete stacking ensemble are evaluated for classification performance but are not included in the edge deployment stage. The reviewed studies demonstrate the effectiveness of MFCC-based representations, Chroma-related features, conventional machine-learning classifiers, and ensemble learning for speech and language recognition. However, there has been relatively limited systematic investigation into the use of MFCC, Chroma, and MFCC-Chroma fusion for multilingual spoken Indian language recognition across heterogeneous classifiers such as SVM, RF, and XGBoost. Additionally, the integration of these classifiers through stacking, with LR serving as a meta-classifier, as well as the

practical deployment of selected models on Raspberry Pi, remains insufficiently explored. Consequently, this study aims to investigate hybrid MFCC-Chroma feature fusion with heterogeneous machine-learning and stacking-based classification, and evaluates the feasibility of deploying SVM and RF models on a Raspberry Pi platform.

3. METHODOLOGY

This section describes the overall methodology of the proposed spoken Indian language identification system. The system consists of different stages including speech data collection, data acquisition, preprocessing, feature extraction, feature fusion, classifier training using SVM, RF, XGBoost as base models and LR as meta model, and performance evaluation using standard classification metrics. The complete workflow of the proposed system is illustrated in Figure 1, while the individual stages are described in detail in the subsequent subsections. The language recognition system for classifying six Indian languages—Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam is structured as a sequential pipeline, as shown in Figure 1.

3.1 Data collection

The spoken Indian language recognition system was developed using the Mozilla Common Voice corpus as the only experimental dataset. The six Indian languages considered were Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam, which consisted of about 14,000 audio recordings. The number of recordings per language was around 2,300–2,400, allowing for a relatively even distribution across the six language classes.

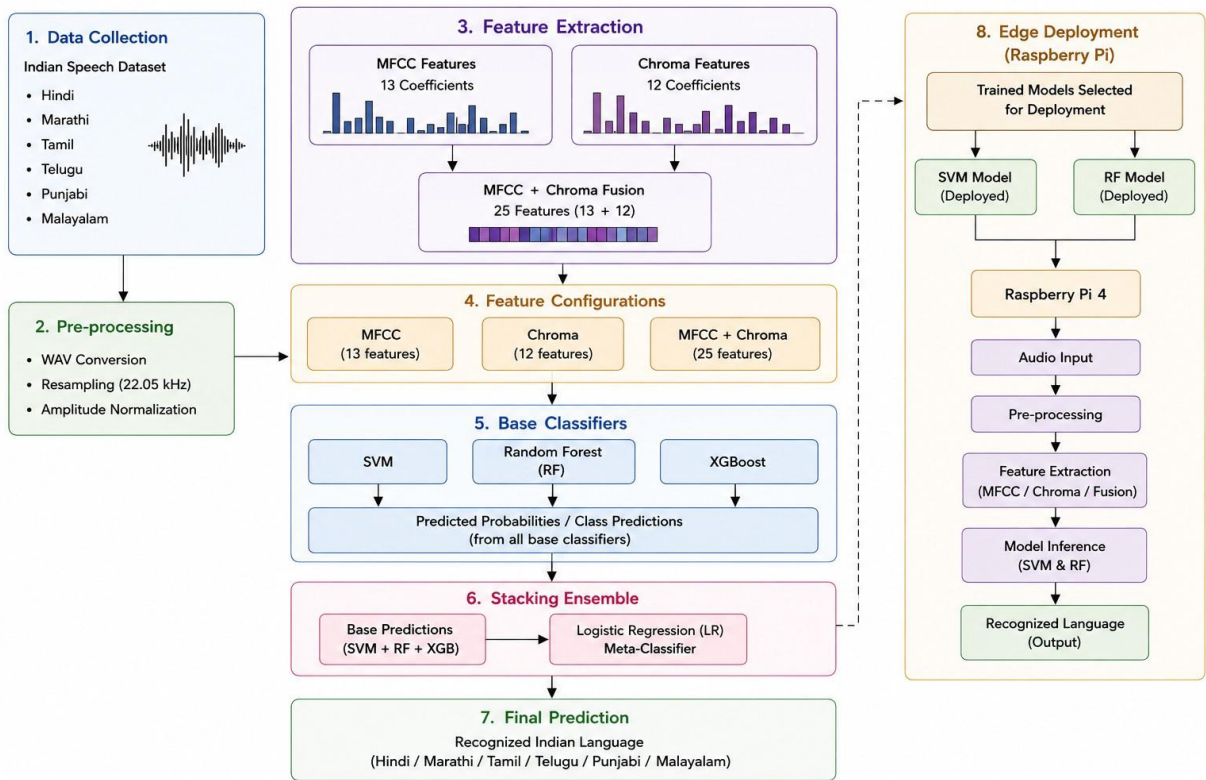


Figure 1. Proposed framework for spoken Indian language recognition using Mel-Frequency Cepstral Coefficients (MFCCs)-Chroma feature fusion, heterogeneous machine-learning classifiers, stacking ensemble, and Raspberry Pi-based deployment

Table 1. Dataset description

Sr. No.	Parameter	Details
1	No. of Indian languages	Six
2	Languages and number of samples	Hindi (2300), Marathi (2300), Tamil (2400), Telugu (2300), Punjabi (2400), Malayalam (2300)
3	Dataset source	Mozilla Common Voice
4	Common voice samples	Approximately 14,000 recordings
5	Age	Different age groups based on voluntary metadata
6	Audio duration	Approximately 5–10 seconds per recording
7	Total dataset duration	Approximately 35–40 hours of speech data
8	Recording environment	Crowdsourced recordings using different devices and environments
9	Male/Female	Male/Female distribution depends on available metadata
10	Number of speakers	Multiple contributors
11	Training set	Approximately 11,200 recordings
12	Testing set	Approximately 2,800 recordings
13	Dialect/regional variation	Variation in regional dialects and accents within each language
14	Evaluation	Speaker dependent

The Mozilla Common Voice corpus is variable in speakers, accents, recording devices, and environments, and is thus suitable to evaluate multilingual spoken language recognition. The dataset contains recordings contributed by multiple speakers, with demographic information such as age and gender available for some recordings through the associated metadata. The details of the dataset used in this study, including the language categories and dataset characteristics, are presented in Table 1.

The final experiments followed a speaker-dependent evaluation protocol. The recordings were divided into training and testing subsets using an 80:20 stratified split, with stratification performed according to language classes to preserve their overall distribution. Speaker identities were not explicitly considered when constructing the partitions. Therefore, recordings from the same speaker may occur in both the training and testing subsets.

This experimental setup enables the assessment of the classification performance of the proposed framework under the selected six-language dataset and speaker-dependent evaluation protocol. Although the dataset provides considerable speaker and recording variability, a dedicated speaker-disjoint evaluation was not performed in the present study. Future work will therefore investigate speaker-independent evaluation using disjoint speaker partitions to assess the robustness of the framework on completely unseen speakers.

3.2 Data preprocessing

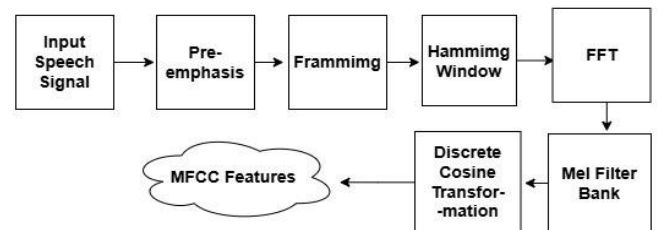
The audio recordings collected from the Mozilla Common Voice corpus were pre-processed before feature extraction. Subsequently, the recordings were resampled to 22,050 Hz and normalized to a common audio format to have a common sampling rate across the dataset. Then the Pydub library was used to normalize the amplitude to make the signal amplitude more consistent across records. A metadata CSV format was created to associate each processed audio file with the corresponding language label. Six language courses were offered—Hindi, Malayalam, Marathi, Punjabi, Tamil, and Telugu. After preprocessing, the dataset was split into training (80%) and testing (20%) subsets using stratified sampling based on language classes. This stratification maintained the relative representation of the six languages in both subsets.

3.3 Feature extraction techniques

MFCCs, Chroma, and the hybrid feature extraction technique (MFCC + Chroma) are three well-known audio features that were used in this work. The librosa package was used to extract all features at a constant sampling rate of 22,050 Hz. Their complementary ability to capture the spectral, tonal, and frequency distribution characteristics essential for reliable speech analysis served as the basis for their selection.

3.3.1 Mel-Frequency Cepstral Coefficients

Since MFCCs closely resemble how people actually hear sounds, they are one of the most used characteristics in audio processing. By evaluating music using a scale known as the Mel scale, which gives greater weight to lower frequencies, MFCCs take into account the fact that our ears are more sensitive to lower frequencies than higher ones.

**Figure 2.** Block diagram for Mel-Frequency Cepstral Coefficient (MFCC) feature extraction technique

The block diagram of the MFCC feature extraction technique is shown in Figure 2. The MFCC representation was initially considered using 13 static MFCC coefficients along with their first- and second-order temporal derivatives, resulting in a 39-dimensional representation. To obtain a compact feature representation and avoid unnecessary redundancy, the 13 static MFCC coefficients were retained for the final classification experiments. We took 13 MFCCs from every brief audio frame. The audio stream is first divided into tiny, overlapping frames. This is how it operates. A Fast Fourier Transform (FFT) is then used to translate each of these frames from the time domain to the frequency domain, providing us with an image of the distribution of energy across various frequencies. This energy is then passed via a filter bank that is shaped in accordance with the Mel scale [21]. To lessen the effect of extremely large energy differences, we

then take the logarithm of the obtained values. Finally, we apply a Discrete Cosine Transform (DCT) to compress the data and remove correlations between features. The result is a set of 13 coefficients, our MFCCs, that describe the shape of the audio spectrum. MFCCs are widely used as acoustic feature representations for speech and audio signal analysis [22]. Several studies have explored multilingual speech recognition and language identification for Indian languages [23].

3.3.2 Chroma feature extraction technique

An effective tool for analyzing audio signals, Chroma features were first created for music analysis but are now being used more for speech-related applications. Distribution of energy among the 12 pitch classes within an octave (C, C#, D, B); they highlight the tonal content of an audio signal, independent of the octave in which they occur. Because of their abstraction, they are especially useful for recognizing melodic and harmonic patterns that differ somewhat between languages, particularly in tonal or rhythmically rich languages. In music and harmonic signal analysis, Chroma features are frequently used to show how spectral energy is distributed across twelve pitch classes. Nonetheless, their use in speech processing can provide supplementary information about the pitch-related and harmonic features of voiced speech segments. Different vowel distributions, consonant patterns, prosodic characteristics, and pitch-related variations are just a few of the phonetic and phonological structure variations found in spoken languages. The spectral and harmonic organization of speech signals may be impacted by these differences. Consequently, this study investigates the potential of Chroma features as an additional acoustic representation, even though they are not usually considered primary elements for spoken language recognition. In particular, the system can use both cepstral spectral-envelope information and additional pitch-class-related spectral energy data by integrating Chroma features with MFCCs. Instead of being taken for granted, the efficacy of this combination was evaluated experimentally. Figure 3 shows the steps for Chroma feature extraction.

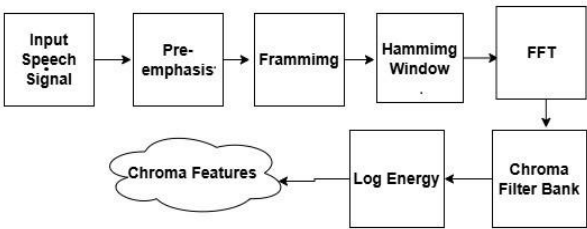


Figure 3. Block diagram for Chroma feature extraction technique

Twelve Chroma characteristics, which correspond to these twelve pitch classes, were retrieved for every audio sample used in the present research. In order to differentiate between phonetically related languages, these qualities provide a condensed yet expressive depiction of tonal characteristics. Pre-emphasis, a preprocessing technique that raises higher frequencies to balance the spectral energy and correct the natural attenuation brought on by the human vocal tract, is the first stage in the Chroma feature extraction pipeline. A Hamming window is applied to each frame, smoothing out discontinuities at the frame boundaries to reduce spectral leakage. The transformed frames are then subjected to an FFT, converting the time-domain signals into their frequency-

domain representations. From the resulting power spectrum, a Chroma filter bank maps the spectral energy into the 12 pitch classes. This grouping captures frequencies that are harmonically related across octaves, thus summarizing the harmonic content of the signal. To better align with human auditory perception, a logarithmic transformation is applied to the computed energies in each pitch class, producing the final Chroma vector. While Chroma features have been widely used in music information retrieval—such as in chord detection or key estimation—their application in multilingual speech processing is gaining traction. They prove particularly valuable in capturing language-specific tonal cues and intonation patterns, which may not be fully captured by spectral features alone.

4. MACHINE LEARNING ALGORITHMS FOR CLASSIFICATION

To enhance classification performance, an ensemble learning technique based on stacking is used. All classifiers use the hybrid feature vector as input.

4.1 Base classifiers

Three machine learning classifiers are used as base learners.

4.1.1 Support Vector Machine

SVM attempts to find an optimal hyperplane to differentiate between data points of various classes and improve the margin between them. The decision function is shown in Eq. (1):

$$f(x) = W^T x + b \tag{1}$$

where,

- x : input feature vector (fused MFCC-Chroma features),
- W : weight vector,
- $W^T x$: weighted combination of input features,
- b : bias/intercept $f(x)$: decision function used to determine the class.

For non-linear classification, Radial Basis Function (RBF) is used to map the input features to a higher-dimensional space. SVMs are efficient and powerful for data with many features, and show good generalization and standardization ability. They are therefore better suited for speech classification [11]. The hyperparameters used for the SVM classifier are presented in Table 2.

Table 2. Hyperparameters of the Support Vector Machine (SVM)

Parameter	Value
Regularization parameter (C)	1.0
Kernel	Radial Basis Function (RBF)
Random state	0
Kernel coefficient (γ)	Scale
Multiclass strategy	One-versus-One
Probability estimation	Enabled

4.1.2 Random Forest

In the proposed stacking ensemble framework, RF is employed as one of the base learners to provide complementary classification decisions. The ensemble of decision trees enables RF to capture diverse and non-linear relationships within the fused MFCC–Chroma feature space.

The class probabilities generated by the RF model were used as inputs to the subsequent meta-classifier, together with the outputs of the other base learners. This approach combined the strengths of individual classifiers and reduced the reliance on the decision boundary of a single model, thereby improved the robustness and generalization capability of the proposed spoken Indian language recognition system [13]. For the RF, a suitable Eq. (2) for the classification output is the majority-voting Eq. (2):

$$Y^{\wedge} = \text{mode}\{T_1(x), T_2(x), \dots, T_N(x)\} \quad (2)$$

where, x is the input vector, $T_N(x)$ denotes the class prediction of the N -th decision tree, and N is the total number of trees. The hyperparameter settings used in this work are mentioned in Table 3.

Table 3. Hyperparameters of the Random Forest (RF)

Parameter	Value
Number of estimators	100
Maximum tree depth	None
Minimum samples required to split	2
Random state	0
Feature normalization	StandardScaler
Bootstrap sampling	True
Hyperparameter tuning strategy	Default hyperparameters with preliminary experimentation
Split criterion	Gini impurity

4.1.3 XGBoost

In this study, XGBoost is used as one of the base learners in the suggested stacking ensemble framework. Its ability to capture complex non-linear relationships between the fused MFCC-Chroma features makes it a good complement to other base classifiers. In ensemble learning, XGBoost produces class predictions, which are then sent to the meta-classifier together with the outputs of other base models [14]. This combination allows the ensemble to utilize the complementary decision boundaries of individual learners and improve the overall robustness and generalization ability of the proposed spoken Indian language recognition system. Model prediction is calculated with the following Eq. (3):

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (3)$$

where, x_i = Features of one audio clip, f_k = prediction on x_i .

The outputs of multiple decision trees are combined to generate the final prediction. XGBoost is a good choice for classification problems that use hybrid feature representations, like the MFCC-Chroma features, in speech-based language recognition systems. Hyperparameters for XGBoost are as follows:

- Number of estimators: 100
- Objective function: multi: softprob
- Evaluation metric: mlogloss
- Booster: gbtrees
- Learning rate: 0.1

4.2 Meta-classifier: Logistic Regression

The proposed stacking ensemble used LR as the meta-

classifier to combine the predictions of the base learners. The outputs of the base models, such as SVM, RF, and XGBoost, were combined to form a meta-feature vector. This vector was then provided as input to the LR meta-classifier [16]. The proposed language recognition task has six language classes, so multinomial LR is used. The softmax function estimates the probability of each class of language. The model estimates the probability of each language class and assigns the final prediction to the class with the highest probability. In the implementation, LR was configured using the LBFGS solver, multinomial classification, and a maximum of 1000 iterations. LR provides a computationally efficient linear decision function in the meta-feature space and is therefore suitable for combining the outputs of the base classifiers.

4.3 Stacking mechanism

The proposed stacking ensemble enhances spoken Indian language recognition performance by combining the complementary capabilities of three basic classifiers. The fused MFCC-Chroma feature vector is utilized to train SVM, RF, and XGBoost, which serve as base learners in the first level. For the input audio sample, each base classifier produces class-probability predictions. LR, which serves as the meta-classifier, receives the meta-feature vector created by concatenating these probability outputs. The stacking process can be represented as follows: The outputs of each basic classifier are $h1(x)$, $h2(x)$, and $h3(x)$, in that order. Eq. (4) describes how these outputs are integrated to create meta-features.

$$Z = [h1(x), h2(x), h3(x)] \quad (4)$$

4.4 Algorithm

The proposed algorithm is illustrated through the flowchart in Figure 4. The process begins with audio preprocessing, followed by extraction of 13-dimensional MFCC and 12-dimensional Chroma features. These features are fused to form a 25-dimensional hybrid feature vector and normalized before classification. We propose a stacking ensemble learning framework to enhance the robustness of the spoken Indian language recognition system. The set includes three different base classifiers: SVM, RF, and XGBoost. The base models' predictions are combined with LR as the meta-classifier. One major problem in stacking is to prevent information leaks between the base-level and meta-level learning processes. Consequently, the meta-classifier was not trained directly on the predictions made by the base classifiers using the same samples utilized for training those classifiers. Instead, an out-of-fold (OOF) prediction strategy based on stratified K -fold cross-validation was employed. The training dataset was split into K stratified folds, preserving the distribution of the six language classes. OOF predictions were made using a five-fold cross-validation procedure ($K = 5$) within the training dataset. Each time, the base classifiers were trained on four folds ($K - 1 = 4$), and the remaining fold was left out to generate OOF predictions. This process was repeated five times, ensuring that each fold served exactly once as the held-out fold. The resulting OOF predictions from the base classifiers were subsequently used to train the LR meta-classifier. For every speech sample, the class probability outputs from SVM, RF, and XGBoost were concatenated to create the meta-feature vector. Given that the spoken Indian

language recognition task involved six language classes, each base classifier produced six class probability values. Therefore, the combined outputs from the three base classifiers formed an 18-dimensional meta-feature representation. The LR meta-classifier was trained using these OOF meta-features alongside their corresponding ground-truth language labels. Once the stacking training was done, each base classifier was trained again on the whole training dataset. The retrained models output the class probability predictions for the independent test set, which were concatenated and fed into the LR model to give the final language prediction. Using OOF predictions reduced the risk of overfitting, as the meta-classifier was trained on predictions of samples that were not used in the training of the corresponding base models. The independent test set was also completely isolated from the stacking training procedure and was used only to evaluate the final performance.

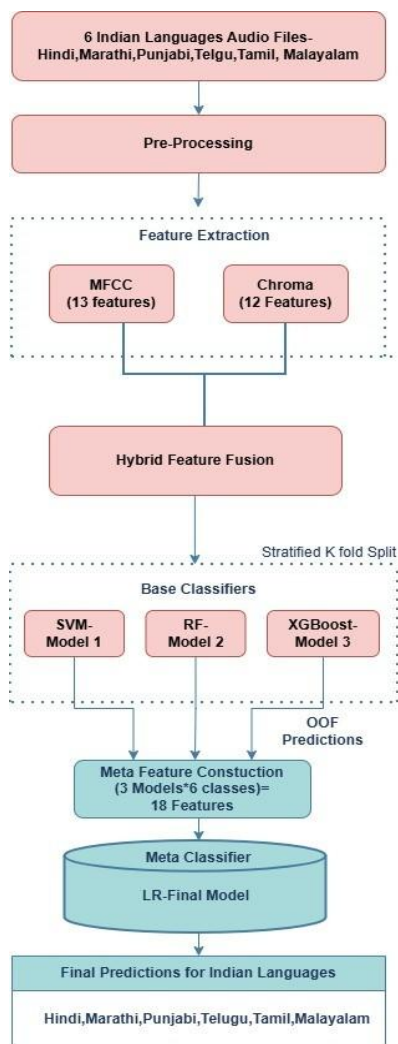


Figure 4. Algorithmic workflow of proposed hybrid feature-based ensemble algorithm

4.5 Real-time inference

4.5.1 Development and training environment

The development and training experiments were mainly conducted using Google Colab, which offered a free-tier CPU-only runtime with approximately two virtual CPUs, a 2.2 GHz Intel Xeon processor, 8 GB of RAM, and about 40 GB of storage. Additional development activities were carried out on

a local workstation featuring an Intel Core i3-8130U processor, 4 GB of DDR4 RAM, a 256 GB HDD, and running Ubuntu 20.04 LTS. For edge-device deployment, the trained SVM and RF classifiers that utilized the MFCC-Chroma feature combination were selected. The trained models and the corresponding feature scaler were saved as .pkl files for future deployment.

4.5.2 Raspberry Pi deployment environment

To analyze the practical feasibility of the proposed spoken Indian language recognition system on a resource-constrained edge device, the trained classifiers were deployed on a Raspberry Pi 4 Model B. Speech input was via a standard 3.5-mm plug-in omnidirectional microphone. The audio was represented as 16-bit samples with a sample rate of 22,050 Hz. To maintain consistency between the training and deployment environments, the same preprocessing and MFCC-Chroma feature extraction procedures used during model development were also applied during inference. The deployed system used Gradio as the user interface where users could record their voice or upload audio files. The SVM and RF inference applications were executed using the Raspberry Pi, which was accessed via the respective Gradio servers.

4.5.3 Inference procedure

The deployment and inference procedure consisted of the following steps:

1. Connect the Raspberry Pi with the required operating system and Python libraries (Raspbian, librosa 0.9.2, gradio 3.23.0, tensorflow 2.11.0, Pandas 1.5.3, numpy 1.24.3, scikit-learn 1.2.2, joblib 1.2.0).
2. Launch the SVM and RF Gradio applications on the Raspberry Pi.
3. Upload an audio sample (file upload or microphone recording).
4. Apply the same preprocessing and feature extraction procedures used during training.
5. Scale the extracted features using the saved feature scaler.
6. Pass the resulting MFCC-Chroma features to the trained classifier.
7. Obtain the predicted language and class probabilities.
8. Display the predicted language and relevant information through the Gradio interface.

For computational evaluation, representative test audio samples were processed ten times under consistent deployment conditions, and the execution times were recorded and averaged.

4.5.4 Network and connectivity

The development environment had an internet connection of approximately 20 Mbps for accessing the dataset and interacting with Google Colab. During the deployment, the Raspberry Pi accessed the Gradio-based application via Wi-Fi (802.11 n, ~20 Mbps) connectivity. The model files were copied into raspberry pi via a USB drive.

5. EXPERIMENTAL RESULTS

5.1 Results for Mel-Frequency Cepstral Coefficient feature extraction technique

After completing the system implementation steps, MFCC

features were calculated. Among all features, the first 13 features are more prominent, and they capture the dominant spectral envelope of speech, which contains the vital information for language recognition [24]. Figure 5 illustrates how the spectral characteristics of the audio signal change over approximately 3.5 seconds, which displays a heatmap of MFCCs taken from a speech sample. Time in seconds is plotted along the x-axis, and the 13 MFCC coefficients, which are frequently employed to describe the acoustic structure of speech, are plotted along the y-axis. The energy of each coefficient at a specific point in time is shown by the color scale, where blue implies lower energy and red indicates higher energy.

Six Indian languages—Hindi, Malayalam, Marathi, Punjabi, Tamil, and Telugu—have their average MFCCs taken from speech samples. The mean values of the 13 MFCC coefficients estimated over a typical collection of audio samples for each language are shown by each curve. In line with normal.

MFCC behavior, where the lower-order coefficients capture the majority of the spectrum energy and formant structure, a noticeable peak is seen in the second coefficient for all

languages. Because of their unique retroflex phonemes and more syllable-timed rhythm, Malayalam and Tamil exhibit somewhat higher MFCC patterns in average coefficients in the graph above. The disparity in phonological characteristics between Punjabi and Marathi, where it is reflected in the lower mean values throughout the same range. This comparative MFCC profile demonstrates that inter-language acoustic variability can be captured even with a very low-dimensional spectral representation (13 MFCCs).

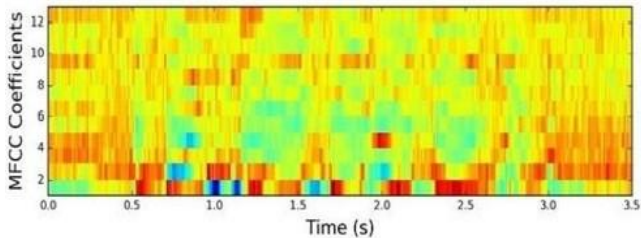


Figure 5. Sample Mel-Frequency Cepstral Coefficients (MFCC) features extracted from an audio signal visualized as a heatmap

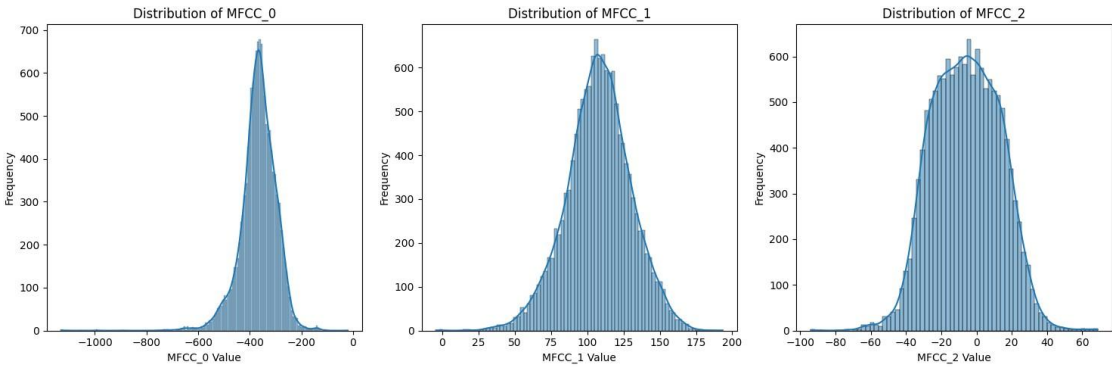


Figure 6. Mel-Frequency Cepstral Coefficients (MFCC) features

Figure 6 illustrates the distributions of the first three MFCC coefficients extracted from the speech samples. MFCC_0 exhibits a relatively concentrated distribution with negative values, whereas MFCC_1 and MFCC_2 show approximately bell-shaped distributions with different ranges and variances. The observed differences in scale among the MFCC coefficients highlight the importance of feature normalization before classification. Therefore, StandardScaler normalization was applied to the extracted feature vectors to ensure that features with larger numerical ranges did not disproportionately influence the machine learning models.

5.2 Results for Chroma feature extraction technique

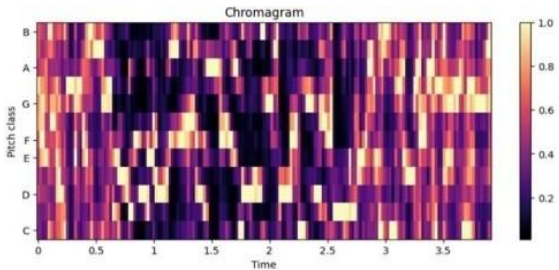


Figure 7. Sample Chroma features extracted from an audio signal visualized as a Chromagram

Figure 7 illustrates the average Chroma feature values extracted from audio samples across six Indian languages: Hindi, Marathi, Tamil, Telugu, Punjabi, and Malayalam. A language's tonal profile over the 12 Chroma pitch classes (from C to B), demonstrating the distribution of harmonic energy inside an octave, is shown by each line in the graph. One noteworthy finding from the plot is that, for the majority of pitch classes, Malayalam has the highest average Chroma values, especially from Chroma bins 6 to 11.

Figure 8 shows the distribution of the first three Chroma features (Chroma_0, Chroma_1, and Chroma_2). The values are concentrated mainly between 0 and 0.8, and all three features exhibit approximately right-skewed distributions. The features show slightly different centers and spreads, indicating variation in their statistical characteristics. Since the hybrid feature vector combines MFCC and Chroma features with different numerical ranges, feature standardization was applied prior to classification to provide comparable feature scales and improve the stability of the learning process. Acoustic speech parameters and spectral characteristics have also been investigated for spoken Indian language recognition, including applications involving Indian languages [25].

5.3 Result for combined feature extraction technique

Figure 9 illustrates the average combined MFCC and

Chroma feature values across six Indian Languages. MFCC feature indices are from 0 to 13. Chroma feature index from 14 to 25. The hybrid model demonstrates effective performance in spoken Indian language recognition. When we apply the hybrid (combined) features to different machine

learning algorithms, it demonstrates effective performance. Hybrid feature representations and deep learning approaches have been investigated for spoken Indian language recognition, demonstrating the potential of combining complementary acoustic or learned representations [26, 27].

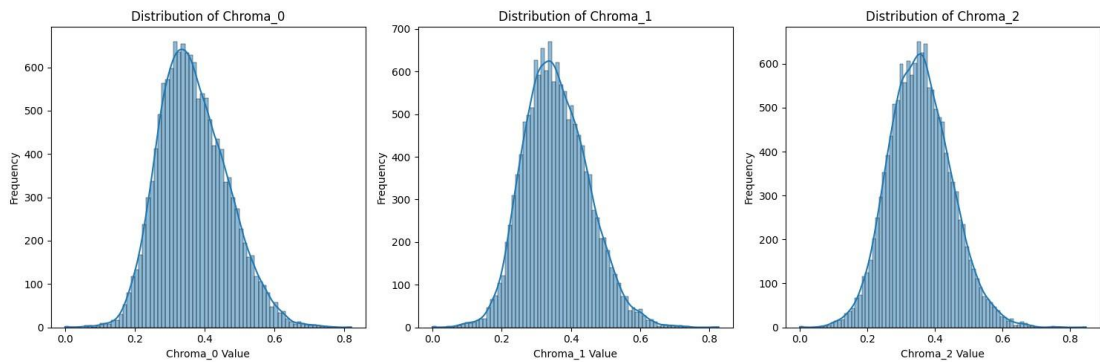


Figure 8. Chroma features

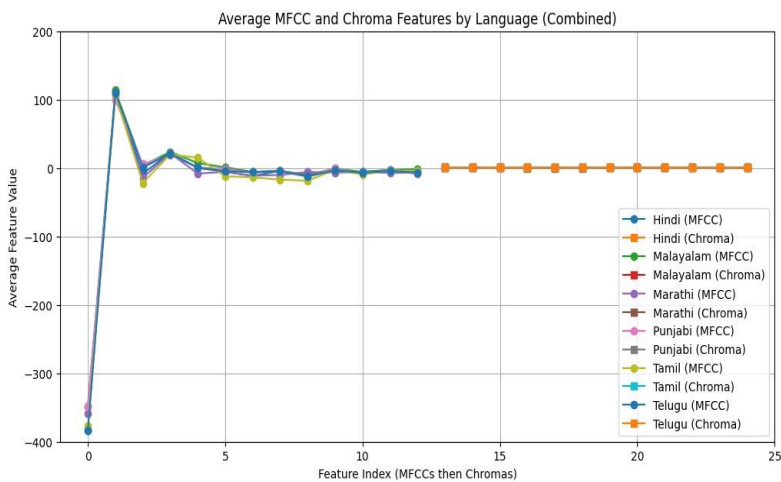


Figure 9. Graph of average Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features (combined) for Indian languages

Table 4. Comparative analysis of different classifiers with Mel-Frequency Cepstral Coefficients (MFCC), Chroma, and hybrid feature extraction method (MFCC + Chroma)

Feature	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MFCC	SVM	83.0	83.3	83.0	83.15
Chroma	SVM	53.4	53.6	53.4	53.50
Combined	SVM	87.6	87.7	87.6	87.65
MFCC	XGBoost	83.1	83.2	83.1	83.1
Chroma	XGBoost	83.9	84	84	84
Combined	XGBoost	87.1	87.2	87.1	87.1
MFCC	RF	82.5	82.7	82.5	82.60
Chroma	RF	53.1	53.2	53.1	53.15
Combined	RF	85.4	85.9	85.6	85.4
MFCC	LR	57.39	56.34	57.3	56.5
Chroma	LR	42.8	41.2	42.8	41.6
Combined	LR	65	64.2	65	64.3
MFCC	Stacking	85.4	85	85	85
Chroma	Stacking	57.9	58	58	58
Combined	Stacking	89.18	89.17	89.1	89.1

Note: MFCC = Mel-Frequency Cepstral Coefficients, SVM = Support Vector Machine, RF = Random Forest, XGBoost = Extreme Gradient Boosting, LR = Logistic Regression.

5.4 Results for classifier

The experimental results indicate that the combined MFCC–Chroma representation consistently outperforms the individual feature representations for the classifiers evaluated.

The stacking ensemble achieved an accuracy of 85.4% using MFCC features and 57.9% using Chroma features. In contrast, the combined MFCC–Chroma representation reached the highest stacking accuracy of 89.18%.

Cross-lingual speech representation learning has further

emerged as an important approach for addressing multilingual and low-resource Indic speech recognition problems [28]. The results demonstrate that the hybrid approach significantly improves classification performance, validating the necessity of feature fusion. Table 4 gives the comparative result analysis of different classifiers with MFCC, Chroma, and hybrid features.

Table 4 presents the performance of the individual base classifiers, meta-model, and proposed stacking ensemble. Ensemble stacking has also been applied to speech-related classification tasks, where combining multiple classifiers can improve the robustness of the resulting predictions [29]. Among the base models, SVM achieved the highest accuracy of 87.6%, followed by XGBoost (87.1%) and RF (85.4%). The LR meta-model achieved 65% accuracy. Zvarevashe and Olugbara [30] investigated a stacked ensemble learning approach for speech emotion recognition. The proposed stacking ensemble further improved the accuracy to 89.2%, demonstrating the benefit of combining complementary predictions from multiple base classifiers.

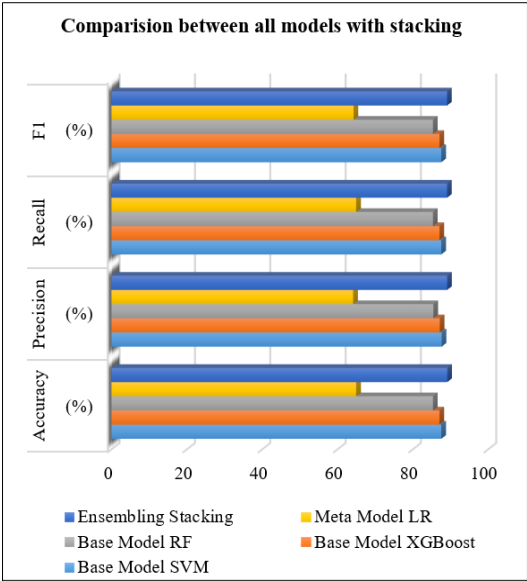


Figure 10. Comparative evaluation of the base models, meta models, and proposed stacking model

Figure 10 presents a comparative evaluation of the base models, meta-model, and the proposed stacking ensemble using accuracy, precision, recall, and F1-score. The ensemble Stacking model demonstrates the highest overall performance across all four evaluation metrics, outperforming the individual RF, XGBoost, and LR meta-models [31, 32]. Figure 11 presents the confusion matrix of the proposed stacking ensemble for six Indian languages. The model demonstrates strong classification performance, with most samples correctly classified along the diagonal. Tamil achieved the highest number of correct predictions (455), followed by Punjabi (435), Marathi (417), Malayalam (403), Telugu (395), and Hindi (392). The major classification confusions were observed between Hindi and Telugu, Hindi and Marathi, and Malayalam and Telugu. These errors may be attributed to similarities in phonetic, acoustic, and linguistic characteristics among the respective languages. Overall, the relatively low off-diagonal values indicate that the proposed stacking ensemble provides effective discrimination among the six Indian languages.

Figure 12 illustrates the ROC curves of all evaluated classifiers, including the stacking ensemble, providing a comparative analysis of their classification performance across the six Indian languages. All models exhibit high performance in distinguishing between the language classes, with area under the curve (AUC) scores generally high. The XGBoost and RF models show the highest micro-average AUCs, suggesting a marginal advantage in overall classification performance compared to SVM and LR models in this specific task. All models significantly outperform a random guess (AUC = 0.5), indicating their effectiveness in the spoken Indian language recognition problem.

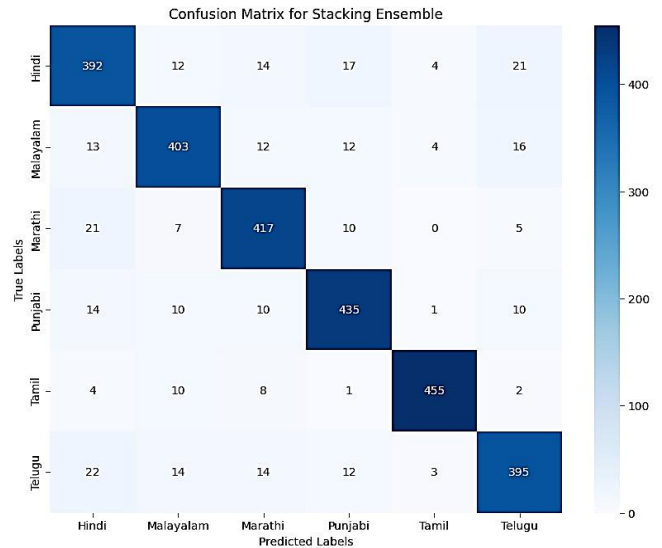


Figure 11. Confusion matrix for the stacking ensemble

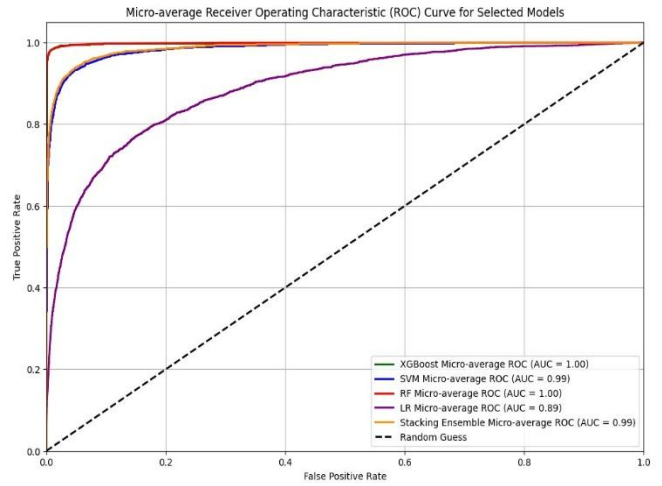


Figure 12. ROC for all classifiers with stacking ensemble

The proposed model achieved an overall classification accuracy of 89.18% with a 95% confidence interval (CI) of 87.97%–90.28%, indicating consistent classification performance across the evaluated languages across 2,800 test samples. The language-wise precision, recall, and F1-score achieved by the stacking ensemble are presented in Table 5. The macro-averaged precision, recall, and F1-score were also 89.1%, indicating consistent performance across the six language classes. Tamil achieved the highest F1-score of 96.1%, whereas Hindi exhibited the lowest F1-score of 84.7%. The relatively lower performance for Hindi and Telugu may

be attributed to greater acoustic similarity or overlapping phonetic characteristics with other Indian languages in the dataset. The micro-averaged precision, recall, and F1-score were 89.2%, which is consistent with the overall accuracy for this single-label multiclass classification task. The weighted-average results were also approximately 89.2%, indicating that the model maintained stable performance despite differences in the number of test samples across language classes.

Table 5. Language-wise precision, recall, and F1-score of the stacking ensemble

Stacking Model: Accuracy = 89.18% (95% CI: 87.97%–90.28%)			
Language	Precision (%)	Recall (%)	F1-Score (%)
Hindi	85	84	84.7
Malayalam	88	89	88
Marathi	89	91	89.2
Punjabi	89	90	90
Tamil	98	95	96.1
Telugu	88	87	86.1
Macro average	89.2	89.13	89.14
Micro average	89.2	89.2	89.2
Weighted average	89.2	89.2	89.2

5.5 Raspberry Pi deployment results

The computational performance of the deployed system was evaluated using representative test audio samples. The proposed language recognition system was deployed on a Raspberry Pi 4 Model B equipped with a quad-core Cortex-A72 (ARM v8) 64-bit processor operating at 1.5 GHz and running Raspbian OS. A generic 3.5-mm omnidirectional microphone was used to capture 16-bit audio at a sampling rate of 22,050 Hz.Each sample was processed ten times under consistent deployment conditions, and the execution times were recorded and averaged. These measurements represent the deployment-level computational performance of the MFCC–Chroma-based SVM and RF applications and were not independently profiled for each classifier. The deployment was intended to demonstrate the feasibility of real-time multilingual language recognition on an edge device. The deployment performance was evaluated in terms of feature extraction time, classification latency, memory consumption, model size, CPU utilization, and throughput. The measured deployment parameters and performance results are summarized as follows.

- Input Microphone: Generic 3.5-mm omnidirectional microphone
- Audio representation: 16-bit
- Sampling Rate: 22,050 Hz
- Features: MFCC + Chroma
- Deployed classifier: SVM, RF
- Feature extraction time: Approx.50–150 ms
- Classification latency: Approx. 200–250 ms/audio sample
- Memory consumption: 300–500 MB
- Model size: 1–5 MB
- Throughput: Approx. 4–5 samples/sec
- CPU Utilization: Approx. 32%–45%
- User Interface: Gradio 7861(SVM),7862(RF)
- Number of repeated runs: 10
- Deployment purpose: Edge-device real-time language recognition

Throughput refers to the complete inference pipeline,

including audio preprocessing, MFCC-Chroma feature extraction, feature scaling, and classifier prediction.

The reported computational measurements are deployment-level measurements obtained from the SVM and RF applications and were not independently attributed to each classifier.

This work shows the computational viability of the proposed hybrid feature-based language recognition framework on a resource-constrained embedded platform for the specified experimental setup. The deployment demonstrates the computational feasibility of the investigated language recognition framework on a Raspberry Pi under the specified experimental conditions. The Gradio-based interface provides a convenient mechanism for audio input and visualization during system evaluation. Figure 13 illustrates the Hindi language prediction using the Random Forest (RF) model on the Gradio GUI.

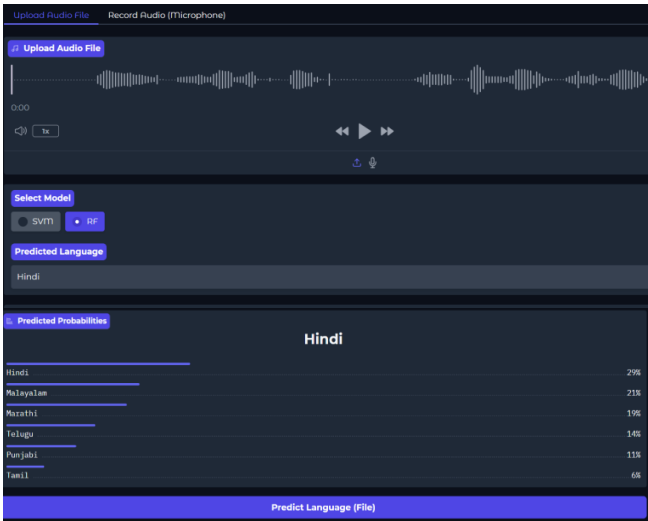


Figure 13. Hindi language prediction using Random Forest (RF) model on Gradio GUI

6. DISCUSSION

The experimental results indicate that the proposed hybrid acoustic feature representation provides useful discriminative information for the recognition of Indian languages. The machine learning classifiers can differentiate between Hindi, Malayalam, Marathi, Punjabi, Tamil, and Telugu through the combination of MFCC and Chroma features, which capture complementary spectral and harmonic characteristics of speech. The overall classification results indicate that the fused representation provides a suitable feature space for traditional machine learning methods.

The comparative evaluation of SVM, RF, XGBoost, and LR indicates that the classifier performance depends on the learning characteristics of each model. SVM can achieve competitive performance on the nonlinear separation of the hybrid feature space, and tree-based techniques, such as RF and XGBoost, are able to model nonlinear feature interactions. LR provides a handy linear baseline. The stacking framework combines the outputs of many classifiers and performs competitively but does not need to outperform the best individual classifier in terms of accuracy. Therefore, the results suggest that ensemble learning should be considered as a complementary strategy rather than a standard because

combining classifiers will increase accuracy.

The class-wise analysis provides a deeper understanding of the recognition behavior of the system. Tamil got the highest F1-score, while Hindi and Telugu got relatively low F1-scores. The confusion matrix indicates that misclassification is concentrated on particular language pairs, which may suggest that the similarities in acoustic and phonetic features are indeed playing a role in the misclassification. Hence, it is more informative to report per-language precision, recall, and F1-score instead of overall accuracy. The macro-averaged F1-score is particularly relevant, as it weighs all six language classes equally. The positive match between macro- and micro-level performances is also consistent with the relatively even distribution of samples across the language classes. The confidence interval and AUC analysis provide more insight into the model's performance. Accuracy is the ratio of correctly classified samples at the selected decision criterion, and AUC is the ability of class discrimination over decision thresholds. These additional metrics give a more complete evaluation of the proposed approach.

On the application side, the use of conventional machine learning algorithms with compact acoustic features is beneficial for resource-constrained environments. This is especially true for the proposed edge deployment, where the computational and memory resources are more constrained compared to the traditional desktop or cloud environment. The present study is limited, however, by the number of languages considered and the extent of evaluation under uncontrolled noise, accent variation, and speaker-independent conditions. Future work will thus include noise-robust evaluation, larger datasets with diverse speakers, more Indian languages, and comparison with recent self-supervised and transformer-based speech representations.

To compare the proposed framework with previous work, Table 6 compares representative conventional feature-based methods with state-of-the-art deep learning, self-supervised, and multilingual speech-processing methods. The comparison is made in terms of the underlying model, speech representation, application scope, and computational characteristics.

Table 6. Comparison of representative existing methods with the proposed spoken Indian language recognition framework

Ref. No.	Technology/ Model	Features/ Representation	Application	Contribution/Advantage	Computational Characteristics
[3]	i-vector	i-vector representation	Language recognition	Provides a compact utterance-level representation and became an important baseline for traditional LID systems	Compact conventional representation
[5]	ANN + feature selection	MFCC time-series + FRESH	Indian & foreign LID	Exploits temporal information in MFCC features and demonstrates the effectiveness of feature selection for LID	Feature-based approach
[7]	SVM + Ensemble SVM	Chroma + spectral-shape features	Kannada/English dialect identification	Demonstrates that Chroma can provide complementary speech/dialect information; particularly useful for limited and noisy data	Conventional feature-based approach
[18]	Self-supervised learning	wav2vec 2.0 representations	Multilingual speech	Learns speech representations directly from raw audio and performs well in low-resource conditions	High computational complexity
[19]	Self-supervised learning	XLS-R	Cross-lingual speech	Large-scale cross-lingual pretraining provides strong multilingual representations without relying exclusively on handcrafted features	High computational and memory requirements
[23]	Transformer- based	Whisper representations	Indic LID + ASR	Jointly integrates LID and multilingual ASR, reducing dependence on separate language-specific systems	Advanced multilingual speech- processing framework
[27]	Deep autoencoder	Bottleneck features	Indian LID	Combines handcrafted and learned bottleneck representations; achieves high accuracy with a relatively compact deep architecture	Deep learning- based approach
[28]	Cross-lingual representation learning	Learned speech representation	Indic languages	Designed specifically to improve cross-lingual speech representation for Indian/Indic languages	Large-scale learned representations
Hybrid acoustic + Ensemble learning (Proposed)	SVM + RF + XGBoost + LR + stacking	MFCC, Chroma,Hybrid(1 3 MFCC + 12 Chroma = 25) features	Hindi, Malayalam, Marathi, Punjabi, Tamil, Telugu	Systematic comparison of multiple acoustic features and classifiers, hybrid-feature evaluation, ensemble learning and Raspberry Pi-based deployment	Compact acoustic-feature framework evaluated for Raspberry Pi deployment

Note: MFCC = Mel-Frequency Cepstral Coefficients, SVM = Support Vector Machine, RF = Random Forest, XGBoost = Extreme Gradient Boosting, LR = Logistic Regression, LID = Language Identification, ANN = Artificial Neural Network, ASR = automatic speech recognition.

Prior approaches to spoken language recognition were largely based on handcrafted acoustic representations such as MFCC-based features, spectral and prosodic features, and i-vector representations, as shown in Table 6. These methods

are generally based on compact feature representations and traditional classification methods. In contrast, there is increasing usage of deep learning and self-supervised representation learning in recent speech processing work, e.g.,

wav2vec 2.0, XLS-R, and cross-lingual speech representations for Indic languages. Such approaches are designed to learn complex representations directly from large-scale speech data. However, direct numerical comparisons with previous studies should be interpreted cautiously because the studies differ in datasets, language sets, dataset sizes, evaluation protocols, and experimental conditions. Therefore, the purpose of the comparison is to position the proposed framework within the broader landscape of spoken language recognition rather than to claim direct performance superiority over other approaches.

7. CONCLUSION

In this work, we present a hybrid-based acoustic feature framework for recognition of six Indian languages, namely Hindi, Malayalam, Marathi, Punjabi, Tamil, and Telugu. In the proposed approach, the effectiveness of combining 13-dimensional MFCCs with 12-dimensional Chroma features to build a 25-dimensional hybrid acoustic representation was investigated. The extracted features were evaluated using multiple machine-learning classifiers, including SVM, RF, XGBoost, and LR, to investigate their suitability for multilingual spoken Indian language recognition.

The experimental results demonstrated that the hybrid MFCC–Chroma representation provided complementary acoustic information and enhanced the discriminative capability of the proposed language recognition system compared with the individual feature representations. The evaluation across the six language classes indicated variations in classifier performance, highlighting the importance of class-wise assessment in multilingual language recognition. The consistent macro- and micro-averaged performance further indicated relatively stable recognition performance across the evaluated language classes. These findings support the applicability of the hybrid MFCC–Chroma representation for multilingual spoken language recognition.

The ensemble stacking approach also achieved competitive performance by combining the outputs of multiple classifiers. However, the best classifier of the individual classifier was not always the best-performing classifier when using stacking. This observation highlights the need of analyzing the ensemble learning methods using multiple performance indicators instead of overall classification accuracy. The results further demonstrate the potential of compact acoustic feature representations and conventional machine-learning models for implementation in computationally constrained environments. The deployment-oriented design suggests that such approaches can provide a practical basis for speech and language recognition applications where computational resources and network connectivity are limited.

Nevertheless, several limitations of the present study should be explicitly acknowledged. First, the evaluation protocol was based on the current dataset partitioning strategy and did not explicitly enforce speaker-disjoint training and testing subsets. Consequently, potential speaker overlap between the training and testing data cannot be completely excluded, and the reported results should therefore be interpreted within the context of the present experimental setup rather than as conclusive evidence of speaker-independent generalization to previously unseen speakers. A speaker-independent evaluation using speaker-disjoint partitions is required to provide a more rigorous assessment of the proposed framework.

Second, the present study considered only six Indian languages. Although these languages represent both Indo-Aryan and Dravidian language families, the linguistic diversity of India is considerably broader. Therefore, the current results cannot be directly generalized to all Indian languages. In future work, the dataset should be extended to more languages, dialects, regional accents, and linguistically related language groups to evaluate the scalability of the proposed approach.

Third, the experiments were conducted using the selected speech corpus and standardized preprocessing conditions. The proposed framework has not yet undergone external validation using independent datasets or systematic evaluation under experimental conditions involving background noise, reverberation, recording-device variability, and diverse speaking conditions. Consequently, the robustness of the proposed system under practical acoustic conditions requires further investigation. Future work should include controlled experiments under different signal-to-noise ratio (SNR) conditions and evaluation using independent external speech datasets to assess the robustness and generalization capability of the proposed framework beyond the experimental conditions considered in the present study.

Future work will therefore focus on speaker-independent evaluation, expansion to a larger number of Indian languages, and external validation under realistic acoustic conditions. Further research will also investigate robustness to background noise, accent and dialect variation, and recording-device differences. In addition, advanced self-supervised and transformer-based speech representations may be explored and compared with the proposed compact acoustic feature framework. These extensions can contribute toward the development of more robust, scalable, and generalizable multilingual spoken Indian language recognition systems suitable for resource-constrained edge environments.

REFERENCES

- [1] Zissman, M.A. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, 4(1): 31-44. <https://doi.org/10.1109/TSA.1996.481450>
- [2] Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., Deller Jr., J.R. (2002). Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In 7th International Conference on Spoken Language Processing (ICSLP 2002), Denver, Colorado, USA, pp. 89-92. <https://doi.org/10.21437/icslp.2002-74>
- [3] Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D., Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction. In *Interspeech 2011*, pp. 857-860. <https://doi.org/10.21437/interspeech.2011-328>
- [4] Davis, S., Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4): 357-366. <https://doi.org/10.1109/tassp.1980.1163420>
- [5] Biswas, M., Rahaman, S., Ahmadian, A., Subari, K., Singh, P.K. (2023). Automatic spoken language identification using MFCC based time series features. *Multimedia Tools and Applications*, 82(7): 9565-9595.

- <https://doi.org/10.1007/s11042-021-11439-1>
- [6] Mukherjee, H., Obaidullah, S.M., Santosh, K.C., Phadikar, S., Roy, K. (2020). A lazy learning-based language identification from speech using MFCC-2 features. *International Journal of Machine Learning and Cybernetics*, 11(1): 1-14. <https://doi.org/10.1007/s13042-019-00928-3>
 - [7] Chittaragi, N.B., Koolagudi, S.G. (2021). Dialect identification using chroma-spectral shape features with ensemble technique. *Computer Speech & Language*, 70: 101230. <https://doi.org/10.1016/j.csl.2021.101230>
 - [8] Rao, K.S., Koolagudi, S.G. (2011). Identification of Hindi dialects and emotions using spectral and prosodic features of speech. *IJSCI: International Journal of Systemics, Cybernetics and Informatics*, 9(4): 24-33.
 - [9] Ramu Reddy, V., Maity, S., Sreenivasa Rao, K. (2013). Identification of Indian languages using multi-level spectral and prosodic features. *International Journal of Speech Technology*, 16(4): 489-511. <https://doi.org/10.1007/s10772-013-9198-0>
 - [10] Dey, S., Sahidullah, M., Saha, G. (2022). An overview of Indian spoken language recognition from machine learning perspective. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(6): 1-45. <https://doi.org/10.1145/3523179>
 - [11] Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3): 273-297. <https://doi.org/10.1007/bf00994018>
 - [12] Ciobanu, A.M., Zampieri, M., Malmasi, S., Pal, S., Dinu, L.P. (2018). Discriminating between Indo-Aryan languages using SVM ensembles. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, Santa Fe, New Mexico, USA, pp. 178-184. <https://doi.org/10.48550/arXiv.1807.03108>
 - [13] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1): 5-32. <https://doi.org/10.1023/a:1010933404324>
 - [14] Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, pp. 785-794. <https://doi.org/10.1145/2939672.2939785>
 - [15] Dietterich, T.G. (2000). Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, Cagliari, Italy, pp. 1-15. https://doi.org/10.1007/3-540-45014-9_1
 - [16] Wolpert, D.H. (1992). Stacked generalization. *Neural Networks*, 5(2): 241-259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
 - [17] Hinton, G., Deng, L., Yu, D., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6): 82-97. <https://doi.org/10.1109/msp.2012.2205597>
 - [18] Baevski, A., Zhou, H., Mohamed, A., Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, pp. 12449-12460.
 - [19] Babu, A., Wang, C., Tjandra, A., et al. (2022). XLS-R: Self-supervised cross-lingual speech representation learning at scale. In *Interspeech 2022*, pp. 2278-2282. <https://doi.org/10.21437/interspeech.2022-143>
 - [20] Mondal, S., Das Barman, A. (2022). Deep learning technique based real-time audio event detection experiment in a distributed system architecture. *Computers & Electrical Engineering*, 102: 108252. <https://doi.org/10.1016/j.compeleceng.2022.108252>
 - [21] Giannakopoulos, T., Pikrakis, A. (2014). *Introduction to Audio Analysis: A MATLAB Approach*. Academic Press. <https://doi.org/10.1016/C2012-0-03524-7>
 - [22] Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, MA, USA, pp. 1-5.
 - [23] Jakhar, N., Srivastava, S., Baby, A. (2024). A unified approach to multilingual automatic speech recognition with improved language identification for Indic languages. In *Interspeech 2024*, pp. 3949-3953. <https://doi.org/10.21437/interspeech.2024-2043>
 - [24] Srinivas, N.S.S., Sugan, N., Kar, N., Kumar, L.S., Nath, M.K., Kanhe, A. (2019). Recognition of spoken languages from acoustic speech signals using Fourier parameters. *Circuits, Systems, and Signal Processing*, 38(11): 5018-5067. <https://doi.org/10.1007/s00034-019-01100-6>
 - [25] Tripathi, K., Reddy, M.K., Rao, K.S. (2020). Multilingual and multimode phone recognition system for Indian languages. *Speech Communication*, 119: 12-23. <https://doi.org/10.1016/j.specom.2020.02.006>
 - [26] Moazzeni, S., Khayyambashi, M.R., Movahhedinia, N., Callegati, F. (2020). Improving the reliability of Byzantine fault-tolerant distributed software-defined networks. *International Journal of Communication Systems*, 33(9): e4372. <https://doi.org/10.1002/dac.4372>
 - [27] Das, H.S., Roy, P. (2020). Bottleneck feature-based hybrid deep autoencoder approach for Indian language identification. *Arabian Journal for Science and Engineering*, 45(4): 3425-3436. <https://doi.org/10.1007/s13369-020-04430-9>
 - [28] Gupta, A., Chadha, H.S., Shah, P., et al. (2021). Clsrl-23: Cross lingual speech representations for indic languages. *arXiv preprint arXiv:2107.07402*. <https://doi.org/10.48550/arXiv.2107.07402>
 - [29] Aljero, M.K.A., Dimililer, N. (2021). A novel stacked ensemble for hate speech recognition. *Applied Sciences*, 11(24): 11684. <https://doi.org/10.3390/app112411684>
 - [30] Zvarevashe, K., Olugbara, O.O. (2020). Recognition of cross-language acoustic emotional valence using stacked ensemble learning. *Algorithms*, 13(10): 246. <https://doi.org/10.3390/a13100246>
 - [31] Sill, J., Takács, G., Mackey, L., Lin, D. (2009). Feature-weighted linear stacking. *arXiv preprint arXiv:0911.0460*. <https://doi.org/10.48550/arXiv.0911.0460>
 - [32] Ganaie, M.A., Hu, M., Malik, A.K., Tanveer, M., Suganthan, P.N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115: 105151. <https://doi.org/10.1016/j.engappai.2022.105151>