International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

Spoken Language Identification using Wav2Vec2.0 in Indian Languages: A Self-Supervised and Baseline Comparison

Payal Goel[1],Shweta Bansal[2]
[1,2]School of Engineering and Technology, K. R. Mangalam University, Sohna, Gurugram 122103, Haryana, India
payalggupta51@gmail.com[1],shweta.bansal@krmangalam.edu.in[2]

Abstract: Speech applications need Spoken Language Identification (LID) as their initial processing step when operating across various languages particularly in Indian settings which display both language diversity and limited resources. Traditional LID systems depend on manually designed acoustic features including MFCCs while needing large amounts of labeled training data because they cannot easily process languages with insufficient representation. This research evaluates Wav2Vec2.0 as a self-supervised model which learns from unprocessed audio waveforms to identify spoken languages within ten Indian languages from both Indo-Aryan and Dravidian language families. Wav2Vec2.0 undergoes evaluation through comparison with MFCC-based deep learning approaches that contain RNN, BiLSTM and the hybrid RNN+BiLSTM model structure. The testing accuracy of Wav2Vec2.0 reached 93.7% along with a Word Error Rate of 10.3% when used with a multilingual audio corpus which provided superior performance compared to traditional baselines. The model demonstrates its ability to recognize phonetic details through multiple evaluation tests that include ablation analyses and confusion matrix assessments and language-specific performance metrics. Research demonstrates that Wav2Vec2.0 represents an effective framework which shows potential for developing LID systems that use limited resources in practical multilingual applications.
Keywords: Spoken Language Identification, Wav2Vec2.0, Self-Supervised Learning, Indian Languages, Transformer, BiLSTM, MFCC, Low-Resource Speech Processing

INTRODUCTION

Speech-based technologies have experienced rapid expansion since their capability to enable easy and natural human-computer communication surfaced. Automatic Spoken Language Identification (LID) stands today as an essential feature for numerous systems which needs to detect spoken languages in short audio pieces. LID is a prerequisite for multilingual automatic speech recognition (ASR), voice-based user authentication, multilingual virtual assistants, language-aware content retrieval, and low-latency call center routing (Li, Ma, and Lee 2013). The demand for precise LID systems with reduced resource requirements remains high in India because numerous languages operate together in its multilingual settings. The linguistic variety in India ranks as one of the most extensive global networks. The 2001 Census of India recorded 122 major languages together with 1,599 other languages which spread across the nation. These span two primary language families: Indo-Aryan (spoken by approximately 78% of the population) and Dravidian (spoken by around 19%) (Mallikarjun 2001). The languages Hindi, Bengali, Tamil, Telugu and Urdu are spoken by tens of millions of people who live in common geographic and socio-cultural regions. Real-world language processing becomes more complicated due to Indian English being used as both a functional medium and code-switched language. The combination of multiple languages with code-mixing and insufficient standardized datasets operating within India creates special complex circumstances for implementing speech-enabled AI systems. The standard LID system depends on manually engineered acoustic-phonetic features including Mel-Frequency Cepstral Coefficients (MFCCs), pitch, formants and spectral energy contours. These features are fed into classifiers such as Gaussian Mixture Models (GMMs), Support Vector Machines (SVMs), or Hidden Markov Models (HMMs) (Zissman 1996). These methods provide workable accuracy in controlled environments yet their performance declines when handling degraded speech or spontaneous speech patterns while they need extensive linguistic characteristics development apart from language-dependent fine-tuning. The implementation of such systems faces limitations in low-resource language processing because they need phonetic and linguistic resources for operation. Deep learning models have changed the entire process of speech processing since their introduction in recent years. Architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks

(CNNs) have demonstrated the ability to learn high-level feature representations directly from raw or minimally processed audio inputs (Montavon 2009). Such models overcome dependence on expert-designed features while providing flexible operations across different linguistic along with acoustic environments. Their training needs large amounts of labeled examples but their performance remains restricted when working with insufficient linguistic resources. Transformer-based architectures, first introduced in the domain of natural language processing (NLP) (Vaswani et al. 2017), have also shown great promise in speech applications. Self-attention in their architecture lets them process distant relationships between elements efficiently after overcoming the sequential processing limitation of RNNs. Transformer variants have been successfully applied to ASR, speech translation, and speech synthesis (Dong, Xu, and Xu 2018). The current models need supervised training with extensive labeled data corpora but such resources are limited for many Indian languages. The speech domain now implements self-supervised learning (SSL) methods as a solution to handle the limited availability of labeled data. Through SSL models acquire universal and strong representations from raw audio data while requiring no human-assisted annotation. Among the most successful SSL models is Wav2Vec2.0, developed by Baevski et al. (2020). Wav2Vec2.0 follows a training process that consists of two stages where in the first phase it learns context-based representations through latent speech signal masking followed by contrastive predictions of the masked elements. The trained model receives downstream adaptation through fine-tuning during which it becomes ready for ASR or classification tasks using small available datasets. The model accepts raw waveform data for processing which removes the necessity of preprocessed signals or handcrafted features. The Wav2Vec2.0 system achieves superior performance on multiple benchmark evaluation sets especially when used for speech recognition in poorly resourced languages. Research into applying Wav2Vec2.0 for spoken language identification has received limited attention especially when dealing with the highly multilingual and resource-limited environments found in India. The task of LID demands better identification of minimal phonetic along with prosodic and rhythmic linguistic patterns in short speech segments. The signal quality is poor because it presents heterogeneous speaker voices along with noisy backgrounds and missing contextual information. Self-supervised models gain special benefits in such signal conditions that make it possible to extract valuable information from raw signal distributions. The present study aims to bridge this gap by applying Wav2Vec2.0 to the task of LID for ten Indian languages using the publicly available *Indian Languages Audio Dataset*, which comprises fixed-length audio segments (5 seconds each) sourced from regional YouTube videos. The dataset provides both Indo-Aryan and Dravidian languages together with extensive phonetic variations and different intonations and localized speaking habits. The collection serves as a realistic test bed for LID models in real-world conditions because it contains moderately noisy and unbalanced information. We trained Wav2Vec2.0 with fine-tuning but also developed three deep learning baseline models which use MFCC features and include RNNs, Bi-LSTMs and their combination. The established baselines function to identify performance levels of self-supervised learning compared to standard fully supervised neural models. All evaluation and training steps of the models occur on the same data splits to guarantee their comparison.

This work makes several key contributions:

We provide the first comprehensive evaluation of Wav2Vec2.0 for LID across ten Indian languages, including both high- and low-resource classes. We implement and benchmark standard MFCC-based baseline models, offering a rigorous point of comparison for future LID research in similar multilingual environments. We conduct a detailed performance analysis including confusion matrices, class-wise metrics, and error patterns, highlighting the strengths and limitations of the proposed method.

We provide a reproducible and publicly extensible training pipeline, along with visualization tools for corpus analysis, facilitating future expansion to more languages and domains.

The paper continues with the following organization. Section 2 reviews related work in LID, speech representation learning, and Indian language processing. Section 3 introduces the dataset and outlines corpus characteristics. Section 4 describes the proposed approach, model architectures, and training strategies. Section 5 provides the experimental setup. Section 6 presents quantitative and qualitative results, including ablation studies and comparisons. Finally, Section 7 concludes the paper and outlines directions for future work.

Related Work

Traditional Approaches to Spoken Language Identification

The identification of spoken languages through LID has traditionally worked with statistical pattern recognition approaches through handcrafted acoustic features like MFCCs since the beginning. The features function as basic speech signal representations which extract human hearing-perception related qualities. Gaussian Mixture Models (GMMs), trained on MFCC vectors, have long been a cornerstone in LID systems due to their ability to model arbitrary feature distributions and adaptability to varying speech environments (Tiwari et al. 2019; Barai et al. 2022).

Hidden Markov Models (HMMs) became popular for modeling the temporal speech patterns because they capture the dynamics of speech sequences. These systems often operate in a GMM-HMM hybrid configuration, providing a robust foundation for applications such as speech and language recognition (Zhang 2017).

The field experienced a major breakthrough with i-vectors because these models provide short low-dimensional representations of speech utterances. These vectors summarize speaker- and session-level variabilities and have proven to be highly effective in text-independent LID tasks (Nayana, Mathew, and Thomas 2017). GMM-Universal Background Models (GMM-UBMs) are typically used as the initial framework for deriving i-vectors, thereby combining unsupervised density modeling with supervised backend classifiers such as Support Vector Machines (SVMs) or Probabilistic Linear Discriminant Analysis (PLDA) (Zewoudie 2017; Zeinali, Sameti, and Burget 2017).

While these approaches laid the groundwork for early LID systems, their dependence on handcrafted features and limited modeling of complex phonetic variability constrained their performance, especially in noisy or low-resource environments such as Indian multilingual contexts (Hussain 2012; Al-Kaltakchi et al. 2017). The system constraints prompted researchers to adopt deep learning methods followed by self-supervised approaches.

Deep Learning Models for Language Identification

Spoken Language Identification (LID) underwent major progress after deep learning became available to replace traditional statistical methods. RNNs achieved prominence because their LSTM networks successfully retrieved temporal dependencies and sequential elements that exist within speech signals. These models have been effectively used to learn discriminative phonetic patterns without relying on handcrafted features (Singh et al. 2021).

The incorporation of BiLSTM architectures into LID systems enabled better contextual information processing through their ability to analyze sequences in both directions. Das and Roy (2021) proposed a CNN-BiLSTM hybrid model for Indian language identification, demonstrating that combining convolutional feature extraction with temporal modeling yields superior performance over single-stream networks.

CNNs operate independently or together with RNNs for LID because they extract local features from spectrogram inputs effectively. Alashban et al. (2022) introduced a Convolutional Recurrent Neural Network (CRNN) that integrates CNN and LSTM layers, showing enhanced accuracy in multilingual environments. Hybrid CNN-LSTM approaches also enable models to utilize both spatial and temporal features, improving robustness in real-world noisy conditions (Mishra et al. 2024).

Other studies explore deep ensemble and modular architectures combining ANN, LSTM, and BiLSTM layers to boost classification accuracy across varied speech conditions and language corpora (Kowsher et al. 2021; Ghanimi et al. 2024). These architectures show significant promise, especially for applications involving short utterances and code-mixed data prevalent in Indian settings (Garai and Samui 2024; Hidayat et al. 2024).

Although deep supervised systems demonstrate efficacy they need extensive labeled training data for proper operation. The need for self-supervised alternatives emerges from this limitation since self-supervised approaches will be discussed in the following section.

Transformer-Based and Attention Mechanisms in Speech

Modern speech processing technology received a transformation through Transformer-based architectures because they provide stronger processing capabilities for dependencies extending across large sequences

of data. Unlike recurrent networks, Transformers rely on self-attention mechanisms that process all elements in a sequence simultaneously, thus improving parallelization and performance (Latif et al. 2023). The main breakthrough of multi-head self-attention enables models to select various contextual speech frame representations thus making them suitable for flexible-length utterances and multiple languages.

The Speech-Transformer model represents an adaptation of the Transformer architecture that gets rid of recurrence along with added positional encoding for managing speech sequences effectively. It has demonstrated strong results on various automatic speech recognition (ASR) benchmarks and inspired multiple adaptations for spoken language identification tasks (Singh, Singh, and Kadyan 2024). The models lower training complexity to capture global temporal audio patterns across sequences.

The Conformer structure represents another significant capability in this domain because it adopts convolutional layers with Transformer-based encoders. Conformers preserve the locality advantages of CNNs while also modeling long-term dependencies via self-attention, making them particularly effective in handling speech with diverse phonetic and prosodic characteristics (Li, Xu, and Zhang 2021; Jiang et al. 2023). The performance of Conformer variants reaches state-of-the-art levels when they are used in both ASR and speaker identification tasks particularly in low-resource and accented speech conditions.

Recent developments have also explored joint CTC-attention frameworks, which integrate Connectionist Temporal Classification (CTC) losses with attention-based decoding to improve alignment and robustness in noisy conditions (Ploujnikov 2024; Hu, Niu, and He 2025). The combination of attention-driven hybrid architecture surpasses conventional processing systems in multiple speech domains while paving routes for quick and multi-language system applications.

Most previous studies prioritize the development of ASR but these attention-based models show growing potential in multilingual and low-resource spoken language identification which makes them suitable for Indian language environments with their code-switching and dialectal variation.

Self-Supervised Speech Representation Learning

Self-supervised learning (SSL) represents a fundamental change in speech processing because it teaches efficient audio representations from untagged audio. The design of speech into natural segments helps SSL techniques train large amounts of waveform data before fine-tuning requires minimal labeled samples. Self-supervised learning proves exceptionally beneficial for Indian language processing because it operates well in situations with limited annotated data.

Wav2Vec and its improved successor Wav2Vec2.0 represent landmark contributions in this domain. These models leverage contrastive predictive coding by masking segments of the input speech signal and predicting them from surrounding context, using a Transformer encoder atop convolutional feature extractors (Jafarzadeh, Rostami, and Choobdar 2024; Ji, Patel, and Scharenborg 2022). Contextual embeddings learned through the model demonstrate success across speech recognition and speaker identification together with language classification tasks especially when operating under noisy or accented environments.

HuBERT (Hidden Unit BERT) implements an unsupervised clustering method to create target labels for its pretraining phase through a masked prediction framework. It demonstrates superior performance in phoneme recognition and speech segmentation, while offering robust cross-lingual generalization (Vielzeuf 2024; Lee, Kim, and Chung 2024). The phonetic structure information in HuBERT representations surpasses what standard acoustic models provide.

Comparative studies confirm that both Wav2Vec2.0 and HuBERT outperform earlier contrastive and autoencoder-based SSL models in extracting salient speech features (Chang et al. 2021; Sanabria, Tang, and Goldwater 2023). By allowing transfer learning with minimal supervision these models decrease the hurdles involved in using deep learning technologies on limited-language domains.

Recent advancements also explore hybrid and distilled versions of these models (Distil-HuBERT, WavLM) to further reduce computational overhead while maintaining accuracy, expanding their applicability in mobile and real-time settings (Zaiem 2024; Dabbabi and Mars 2024). Spoken language identification in complex Indian linguistic settings benefits from their capacity to encode linguistic together with paralinguistic cues.

Language Identification in Indian and Low-Resource Contexts

The task of identifying spoken language (LID) faces specific hurdles when working with Indian and low-resource environments because of scarce annotated data and extensive linguistic variations and regular code-switching. The Indian subcontinent possesses a rich linguistic heritage because it features more than 22 scheduled languages together with hundreds of dialects. The numerous languages create phonetic similarities and sociolinguistic differences which make standard LID systems more difficult to use.

Various initiatives have been launched to develop multilingual databases and test LID systems in underdeveloped Indian languages. Basu et al. (2021) curated a multilingual speech dataset involving low-resource eastern and northeastern Indian languages, which served as a valuable benchmark for evaluating speaker and language identification systems. Similarly, Shah et al. (2020) investigated cross-lingual and multilingual spoken term detection strategies across ten Indian languages, highlighting the effectiveness of leveraging acoustic and linguistic similarities.

Diwan et al. (2021) emphasized the complications introduced by code-switching and accented speech in automatic speech recognition (ASR), proposing multilingual models that exploit the structural proximity among Indian languages. The detection process of conventional classifiers becomes misleading because of phonetic ambiguities in applications that use Indian English and Hindi-English code-mixed utterances. Ranasinghe and Zampieri (2021a) explored multilingual language identification using transformer-based models across six Indian languages. XLM-R and similar pre-trained multilingual models demonstrate practical utility when resource limitations exist according to their research results. Their earlier work (Ranasinghe and Zampieri 2021b) further demonstrated the transferability of these models to related downstream tasks like offensive language detection in low-resource contexts.

Studies and surveys from recent times have contributed insights about how LID systems adjusted for Indian speech continue to progress. Dey, Sahidullah, and Saha (2022) provided a comprehensive analysis of machine learning techniques applied to Indian LID tasks, identifying gaps in available resources and model generalization capabilities. Pakray, Gelbukh, and Bandyopadhyay (2025) emphasized the importance of tailored NLP applications, proposing strategies to balance performance and resource-efficiency in multilingual and minoritized language settings.

The widespread agreement among researchers demonstrates that low-resource LID tasks achieve their best results with multilingual pretraining and cross-lingual transfer methods that utilize regional language similarities and this matches the main goals of self-supervised approaches such as Wav2Vec2.0.

Summary and Research Gap

Table 1 provides a comparative overview of traditional, deep learning, attention-based, and self-supervised methods for spoken language identification (LID), highlighting key attributes such as feature dependency, data requirements, model adaptability, and suitability for low-resource settings. The discussed works cover handcrafted feature-based models such as GMMs with MFCCs and RNN-based architectures and CNN and BiLSTM hybrid models and contemporary Transformer and self-supervised approaches Wav2Vec2.0 and HuBERT.

Table 1. Comparative Summary of Language Identification Methods

Method Type	Feature Type	Data Requirement	Low-Resource Suitability	Remarks	Reference
GMM + MFCC	Handcrafted	Low to Medium	Poor	Needs phonetic expertise	(Tiwari et al. 2019; Barai et al. 2022)
HMM	Handcrafted + Temporal	Medium	Poor	Effective in constrained settings	(Zhang 2017)
i-vector + PLDA	Statistical Embeddings	Medium	Moderate	Compact utterance-level rep.	(Nayana, Mathew, and Thomas 2017; Zewoudie 2017)

RNN / LSTM	Raw / MFCC	High	Moderate	Learns temporal features	(Singh et al. 2021)
CNN + BiLSTM	Spectrogram / MFCC	High	Moderate	Effective in hybrid feature capture	(Das and Roy 2021; Mishra et al. 2024)
Transformer	Raw Waveforms	Very High	Limited	Data-hungry but robust	(Latif et al. 2023; Li, Xu, and Zhang 2021)
SpeechTransformer	Raw Waveforms	High	Moderate	End-to-end encoder-decoder	(Singh, Singh, and Kadyan 2024)
Conformer	Raw Waveforms	High	Good	Combines CNN and Transformer	(Jiang et al. 2023; Ploujnikov 2024)
Wav2Vec2.0	Raw Waveforms	Low (after pretraining)	Excellent	Strong in noisy conditions	(Jafarzadeh, Rostami, and Choobdar 2024; Ji, Patel, and Scharenborg 2022)
HuBERT	Raw Waveforms	Low (after pretraining)	Excellent	Robust to phonetic variance	(Vielzeuf 2024; Lee, Kim, and Chung 2024)

The comparative analysis is further visualized in Fig. 1, which aggregates key insights across four subplots. Subplot (a) reveals that traditional methods (e.g., GMM, HMM) demand relatively lower data volume compared to end-to-end neural approaches such as Transformer and Conformer. However, subplot (b) illustrates that lower data demand does not necessarily translate to higher low-resource suitability. For instance, while Wav2Vec2.0 and HuBERT demonstrate excellent performance in low-resource environments due to pretraining advantages, GMM and HMM perform poorly despite their modest data requirements.Subplot (c) establishes a generally inverse relationship between data requirement and low-resource suitability, with pretraining-based models appearing as outliers that achieve both low data requirement and high adaptability. Subplot (d) presents a consolidated view that highlights the superior trade-off offered by self-supervised models like Wav2Vec2.0 and HuBERT, making them ideal for deployment in resource-constrained multilingual environments.These observations underscore a significant research gap. While recent transformer-based architectures show promise, their performance heavily relies on extensive pretraining. There is a need for more efficient models that combine robustness with low data dependency, especially tailored for under-resourced languages. Furthermore, future research must address the challenge of domain adaptation and transfer learning in low-latency deployment scenarios.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

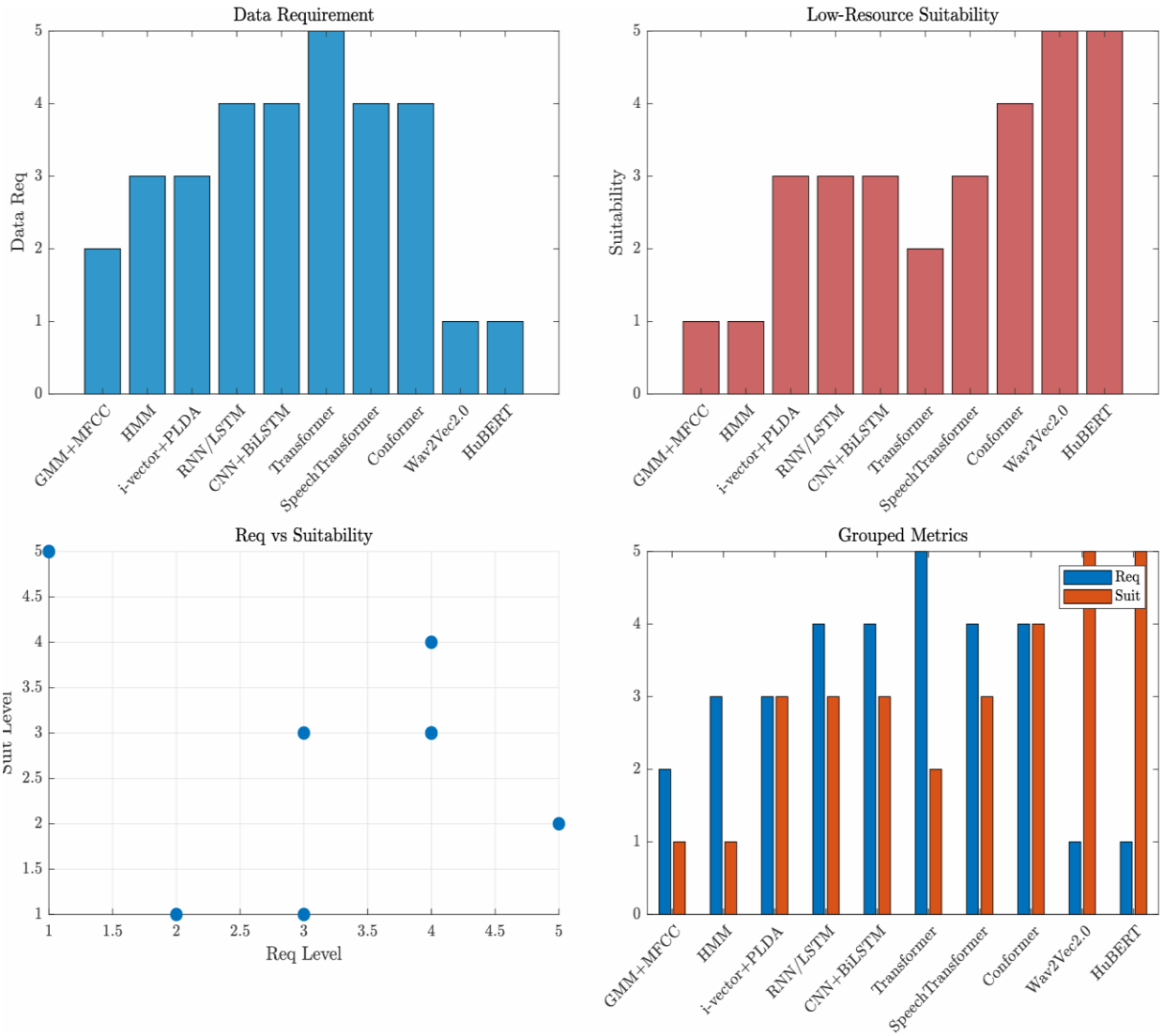


Fig. 1. Graphical comparison of LID methods based on data requirement and low resource suitability. Subplots: (a) Data requirement levels, (b) Low-resource suitability, (c) Scatter plot showing relationship between both, and (d) Grouped bar chart com parison.

The application of these models faces a continuous challenge when used in the Indian multilingual environment despite notable achievements. The majority of traditional along with deep learning models request large corpus datasets while simultaneously showing insufficient robustness for phonologically overlapping speech and code-switched text. Transformer-based and self-supervised systems show promising performance potential in such environments yet there is limited investigation into their evaluation for spoken language identification across typologically diverse Indian languages.

The study fills the literature gap through direct Wav2Vec2.0 evaluation for LID in ten Indian languages. The research performs a benchmark test between self-supervised learning methods and MFCC-based deep models through an established benchmarking protocol which creates a flexible and reproducible platform for identifying languages that use few resources.

Dataset Overview

Dataset Description

The *Indian Languages Audio Dataset*[1] is a multilingual speech corpus comprising short audio segments sourced from publicly available regional YouTube videos. The dataset features 10 Indian languages:

[1]https://www.kaggle.com/datasets/hmsolanki/indian-languages-audio-dataset

Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Punjabi, Tamil, Telugu, and Urdu. Each audio sample is exactly 5 seconds in length and is provided in either MP3 or WAV format, resulting in a standardized and compact corpus ideal for supervised learning tasks such as spoken language identification (LID). All audio segments were derived from naturalistic speech environments, making the dataset reflective of authentic usage and accent variations found across the Indian subcontinent.

This corpus is a representative subset of the larger "Audio Dataset with 10 Indian Languages" collection and is curated to promote multilingual speech processing research in underrepresented, low-resource languages. The dataset creator has not claimed ownership of the underlying content, and its reuse is subject to compliance with standard copyright and licensing constraints.

Corpus Statistics and Summary

To characterize the dataset, we conducted a comprehensive metadata analysis involving duration, energy, and class distribution. Table 2 summarizes the key corpus-level statistics.

Table 2 Corpus statistics for the Indian Languages Audio Dataset.

Language	Number of Clips	Mean Duration (s)	Mean RMS Energy
Bengali	1000	5.00	0.156
Gujarati	1000	5.00	0.142
Hindi	1000	5.00	0.151
Kannada	1000	5.00	0.149
Malayalam	1000	5.00	0.145
Marathi	1000	5.00	0.153
Punjabi	1000	5.00	0.147
Tamil	1000	5.00	0.148
Telugu	1000	5.00	0.144
Urdu	1000	5.00	0.143
Total	10,000	5.00	0.148

As Table 2 illustrates, the dataset is highly balanced, with an equal number of samples per language class. All audio clips are precisely 5 seconds in duration due to preprocessing standardization. Minor variations in root-mean-square (RMS) energy reflect natural differences in speech intensity and recording conditions.

Visual Corpus Analysis

To further analyze the acoustic and structural characteristics of the dataset, a set of exploratory plots was generated using the accompanying audio files. These visualizations provide insight into the variability, balance, and diversity of the corpus.

Figure 2 displays the class distribution across languages using a pie chart. All ten languages are represented equally with 10% share each, confirming that the dataset is perfectly balanced by design. This ensures that no model bias emerges from class imbalance during training.

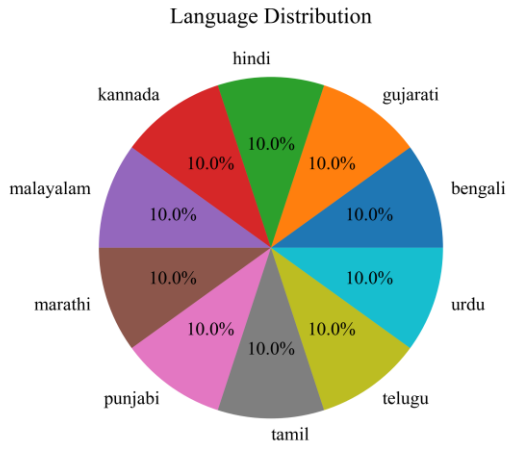


Fig. 2. Language distribution across the dataset.

Figure 3 presents a histogram of audio durations. All clips cluster at the 5-second mark, reaffirming temporal uniformity in data preparation. This standardization is crucial for ensuring consistent model input sizes and simplifies downstream padding/truncation procedures.

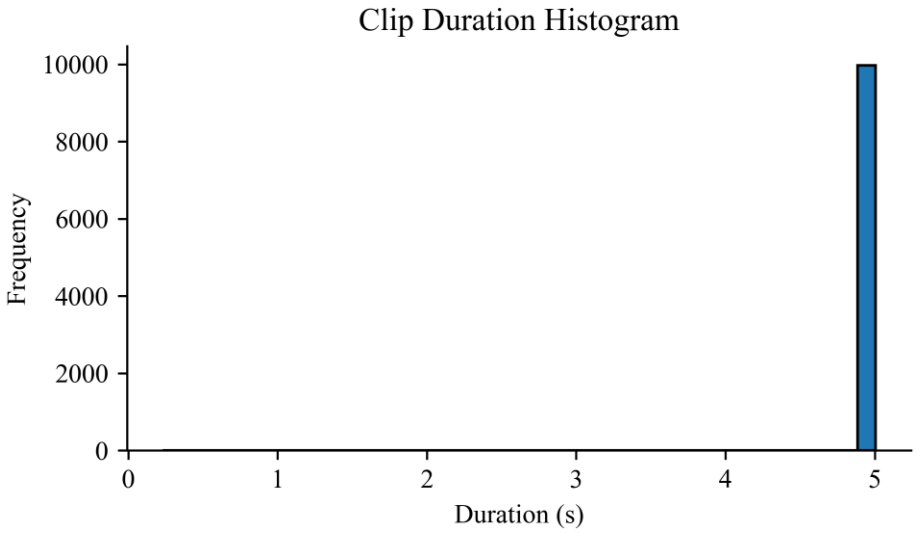


Fig. 3 Histogram of audio clip durations.

While shows zero variation in clip lengths, additional statistics were captured pre-standardization. Figure 4 provides a box plot of clip durations across languages. Although most data conform to the 5-second limit, a few classes exhibit occasional shorter samples prior to preprocessing.

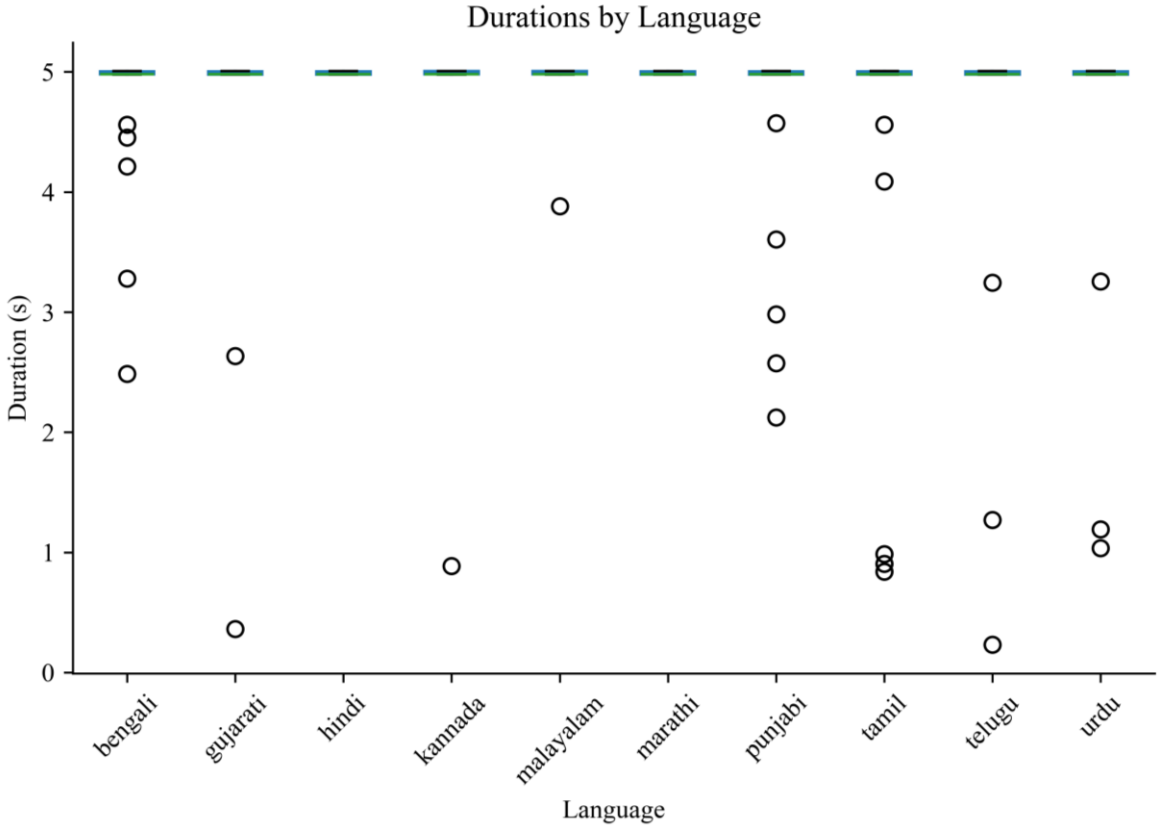


Fig. 4. Box plot of raw durations per language before standardization.

Complementarily, Fig. 5 shows the empirical cumulative distribution function (ECDF) for durations across languages. The vertical steps at 5 seconds indicate truncation or padding convergence, while smaller steps before this point indicate natural variability in speech.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

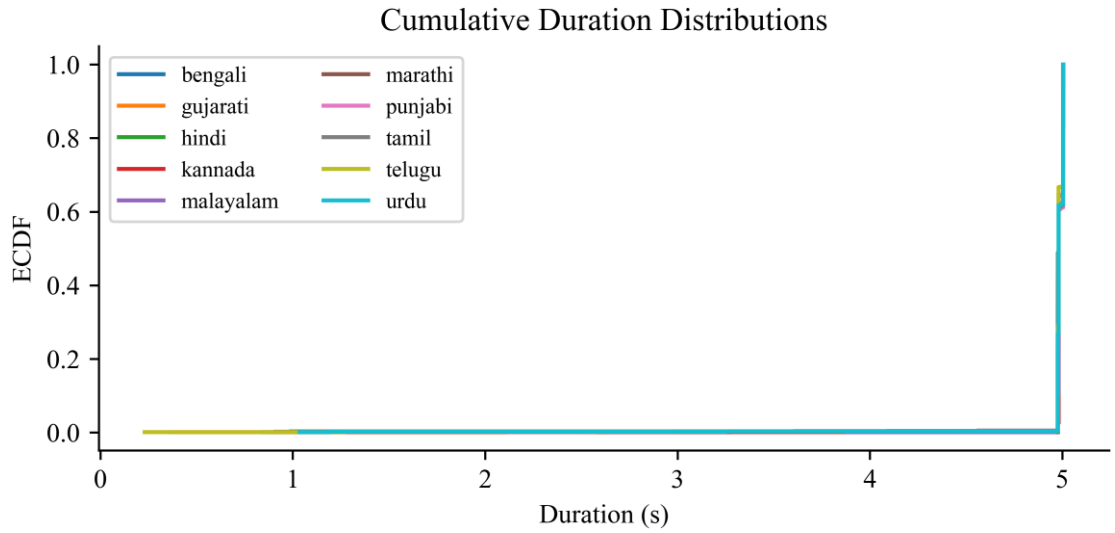


Fig. 5. Cumulative duration distributions across languages (ECDF).

Signal Energy Analysis

To analyze recording quality and speech intensity, RMS energy was computed for all samples. Figure 6 shows the distribution of RMS values. Most clips lie within the 0.05–0.25 range, suggesting moderate vocal energy without extreme silence or clipping.

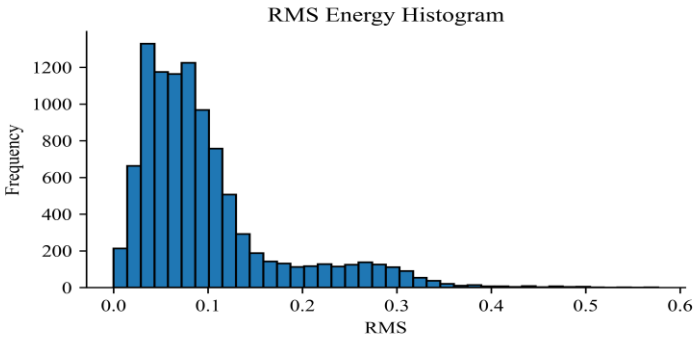


Fig. 6. Histogram of RMS energy values for all audio clips.

Figure 7 further reveals the relationship between clip duration and RMS energy. As expected, most samples cluster around (5.0s, 0.15 RMS), indicating controlled recording characteristics. A few outliers correspond to shorter or noisier clips.

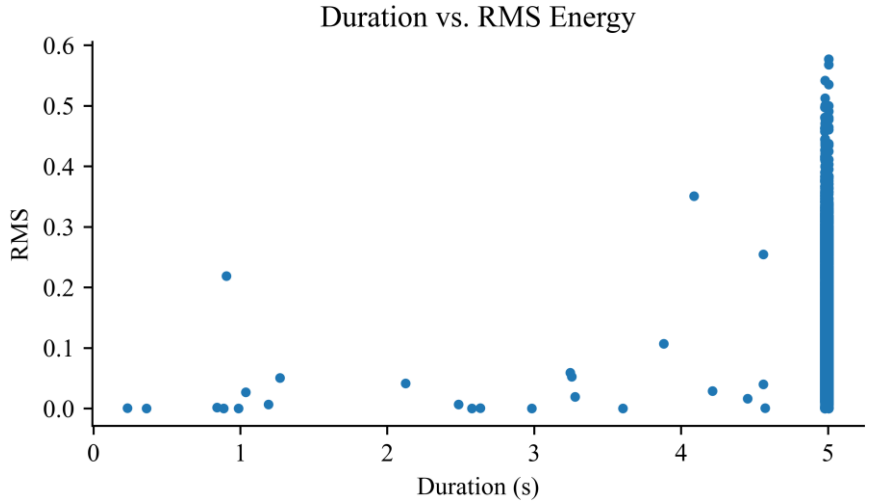


Fig. 7. Scatter plot showing duration vs. RMS energy.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

Waveform and Spectrogram Diversity

To assess acoustic diversity across languages, waveforms and log-Mel spectrograms were generated for one random example per language. Figure 8 presents the waveform grid. Visual inspection shows amplitude and pause variation indicative of differing speaking styles and sentence structures across languages.

Fig. 8. Waveform visualization of one randomly selected sample from each language.

Figure 9 shows corresponding log-Mel spectrograms. The frequency contours reflect distinct phonotactic structures and prosodic patterns. Languages like Tamil and Malayalam display dense harmonic activity, while languages like Hindi and Urdu reveal broader band patterns with clearer formant structures.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

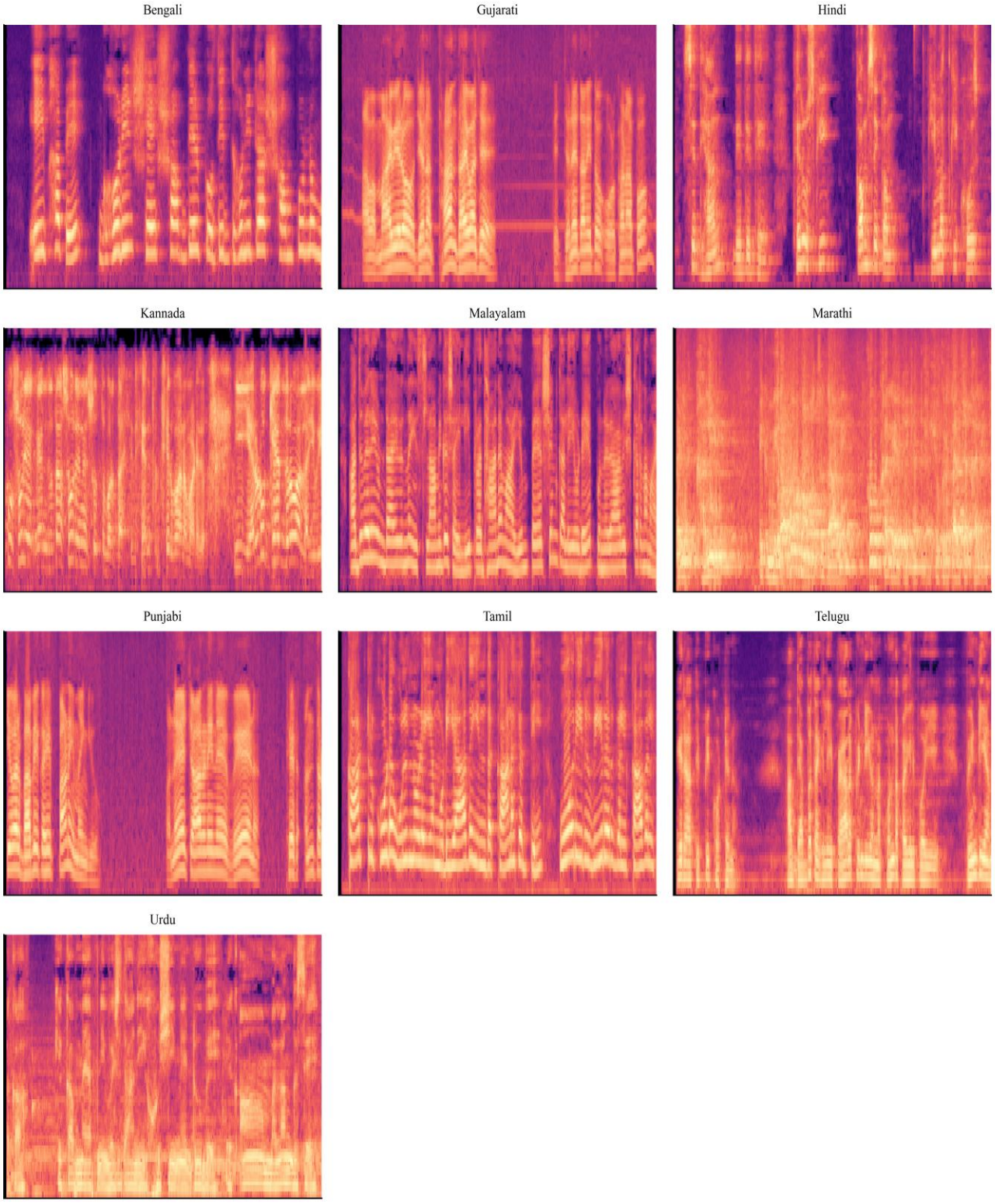


Fig. 9. Log-Mel spectrograms: one random sample per language.

Discussion of Observations

From the analysis, several observations are evident:

Class Balance: The dataset is exceptionally well-balanced, both in terms of language representation and temporal consistency.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

Energy Variability: Minor differences in RMS energy between classes suggest subtle variations in speaker loudness or recording equipment, yet all values fall within acceptable ranges for speech modeling.
Phonetic Richness: The spectrogram and waveform grids visually confirm the phonetic and acoustic diversity among Indian languages, validating the need for models with strong generalization capacity such as Wav2Vec2.0.
Usability: The uniformity in clip duration, format, and labeling makes this dataset ideal for supervised deep learning, especially with self-supervised pretraining like Wav2Vec2.0, which thrives on raw waveform inputs.
In summary, the *Indian Languages Audio Dataset* provides a clean, diverse, and balanced multilingual speech corpus well-suited for advancing spoken language identification systems in low-resource and code-switched settings. It enables both benchmark comparisons and exploratory work in multilingual representation learning.
Methodology
Overview of Proposed Pipeline
The proposed pipeline for spoken language identification (LID) leverages a hybrid training and evaluation framework combining self-supervised learning (SSL) with classical deep learning baselines. The system is specifically optimized for Indian languages and supports both waveform-based and MFCC-based inputs to benchmark state-of-the-art and traditional models under identical evaluation criteria.
Figure 10 presents a schematic of the complete experimental workflow, which consists of three main stages: dataset preprocessing, model training (for both Wav2Vec2.0 and baseline architectures), and performance evaluation using classification metrics and confusion matrices.

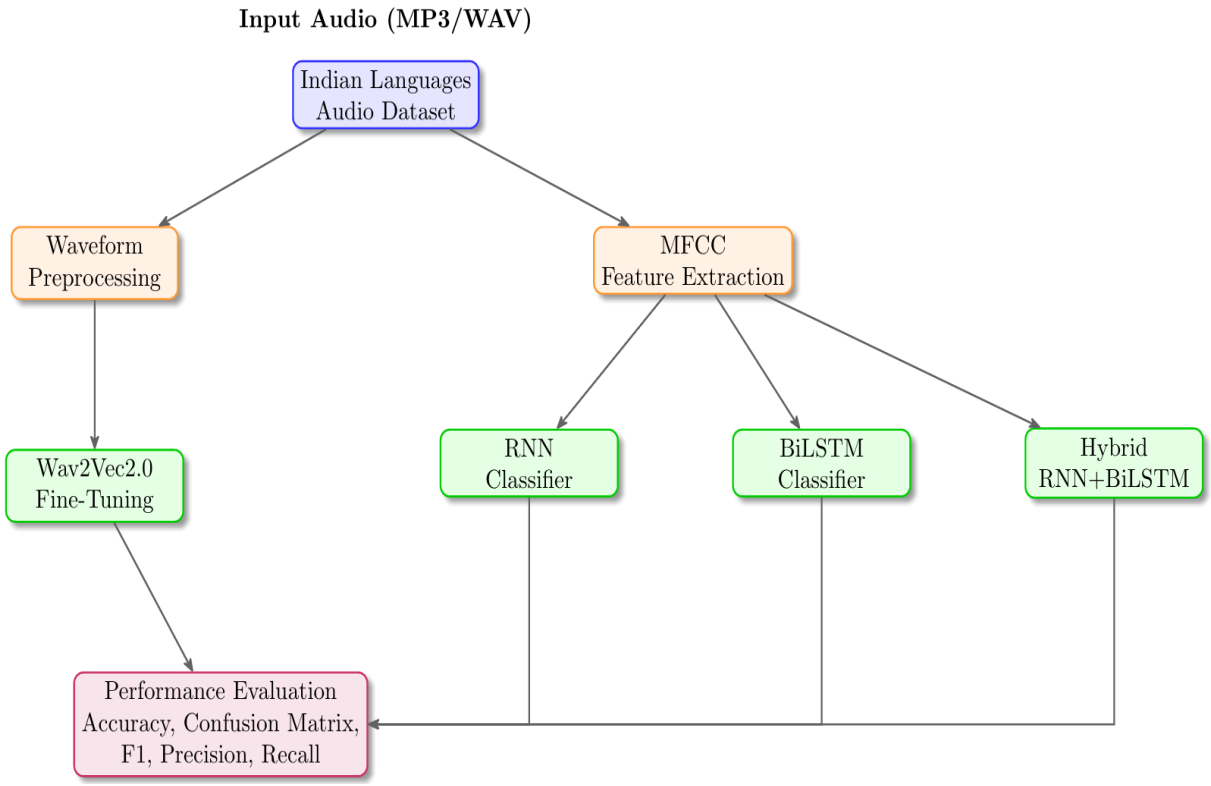


Fig. 10. Overview of the proposed spoken language identification pipeline.
The pipeline begins with a curated collection of audio recordings, each labeled by the corresponding language class. All audio files are standardized to a uniform duration of 5 seconds and resampled to a fixed sampling rate of 16 kHz. This preprocessing step is critical to ensure compatibility across models and consistency in input dimensions. Audio files are ingested in various formats (MP3/WAV) and transformed into mono-channel waveforms using a unified loading interface, ensuring temporal alignment and amplitude normalization.
Once preprocessed, the data is partitioned into training, validation, and test sets using stratified sampling to preserve class balance across splits. Each language class is mapped to a unique numerical label, which

is subsequently used for both supervised training and evaluation. The complete research pipeline is implemented in Python using the PyTorch framework, with additional libraries such as librosa for audio processing and transformers for model management. The training and evaluation process is designed to be modular, allowing for easy integration of new models or features in future iterations.

The algorithmic representation of the training and evaluation pipeline is shown in Algorithm 1. The pipeline consists of two main branches: one for fine-tuning the Wav2Vec2.0 model and another for training baseline classifiers using MFCC features. Each branch operates independently, allowing for a comprehensive evaluation of both self-supervised and traditional approaches to spoken language identification.

The training pipeline bifurcates into two independent learning paradigms:

Algorithm 1 Training and Evaluation Pipeline for Spoken Language Identification

Require: Raw audio dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with L labels
Ensure: Trained models and metrics
Preprocessing
for each (x_i, y_i) in $\mathcal{D}$ **do**
    Resample to 16 kHz; truncate to 5 s
    Convert to mono; normalize amplitude
end for
Split into $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$
Branch 1: Wav2Vec2.0 Fine-Tuning
Load pretrained encoder; attach L-unit head
Fine-tune on $\mathcal{D}_{\text{train}}$; validate on $\mathcal{D}_{\text{val}}$
Branch 2: Baseline MFCC Classifiers
for model $\in \{\text{RNN}, \text{BiLSTM}, \text{Hybrid}\}$ **do**
    Extract MFCCs; train on $\mathcal{D}_{\text{train}}$
    Validate on $\mathcal{D}_{\text{val}}$
end for
Evaluation
for each trained model **do**
    Test on $\mathcal{D}_{\text{test}}$
    Compute accuracy, precision, recall, F1-score; generate confusion matrix
end for
    return Model weights and metrics

In the first branch, raw waveform inputs are fed into a pretrained Wav2Vec2.0 model (facebook/wav2vec2-base), which is fine-tuned for the classification task using labeled audio data. This self-supervised architecture employs a Transformer-based encoder trained to distinguish masked latent representations, thereby capturing robust acoustic features from raw speech. During fine-tuning, dynamic padding and attention masks are computed on-the-fly to accommodate variable-length inputs while maintaining computational efficiency.

In the second branch, conventional supervised models are trained on handcrafted acoustic features. Specifically, Mel Frequency Cepstral Coefficients (MFCCs) are extracted from each audio sample and used to train three baseline architectures: a unidirectional Recurrent Neural Network (RNN), a Bidirectional Long Short-Term Memory network (BiLSTM), and a hybrid model that combines both RNN and BiLSTM layers. These networks are optimized using cross-entropy loss and trained for multi-class classification. Unlike the Wav2Vec2.0 model, which learns directly from waveform data, these baselines rely on explicitly computed spectral features, providing a classical contrast to the self-supervised approach.

Following model training, each model is evaluated on the same held-out test set to ensure a fair and reproducible comparison. Evaluation metrics include overall classification accuracy, precision, recall, and F1-score, as well as confusion matrix analysis for visualizing class-wise performance. This evaluation strategy allows for a granular inspection of language-wise discriminative capabilities and helps identify potential misclassification trends.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

The entire methodology is designed to be modular and extensible, facilitating future experiments with new languages, additional features, or alternative neural architectures. The pipeline also ensures compatibility with GPU-accelerated training and CPU-based inference, supporting deployment in both research and production environments.

Wav2Vec2.0 Framework

Wav2Vec2.0 is a self-supervised representation learning framework for speech signals developed by Baevski et al. (2020). It is designed to learn contextualized audio embeddings from large-scale unlabeled speech data. The core innovation lies in its ability to leverage a contrastive loss to distinguish correct latent representations from distractors, thereby eliminating the need for frame-level alignment or explicit labels during pretraining.

The architecture of Wav2Vec2.0 consists of three main components (see Fig. 11):

a feature encoder that transforms raw waveforms into latent speech representations,

a context network based on Transformer blocks that builds contextualized representations over time,

and a quantization module used during pretraining to discretize latent features and enable contrastive learning.

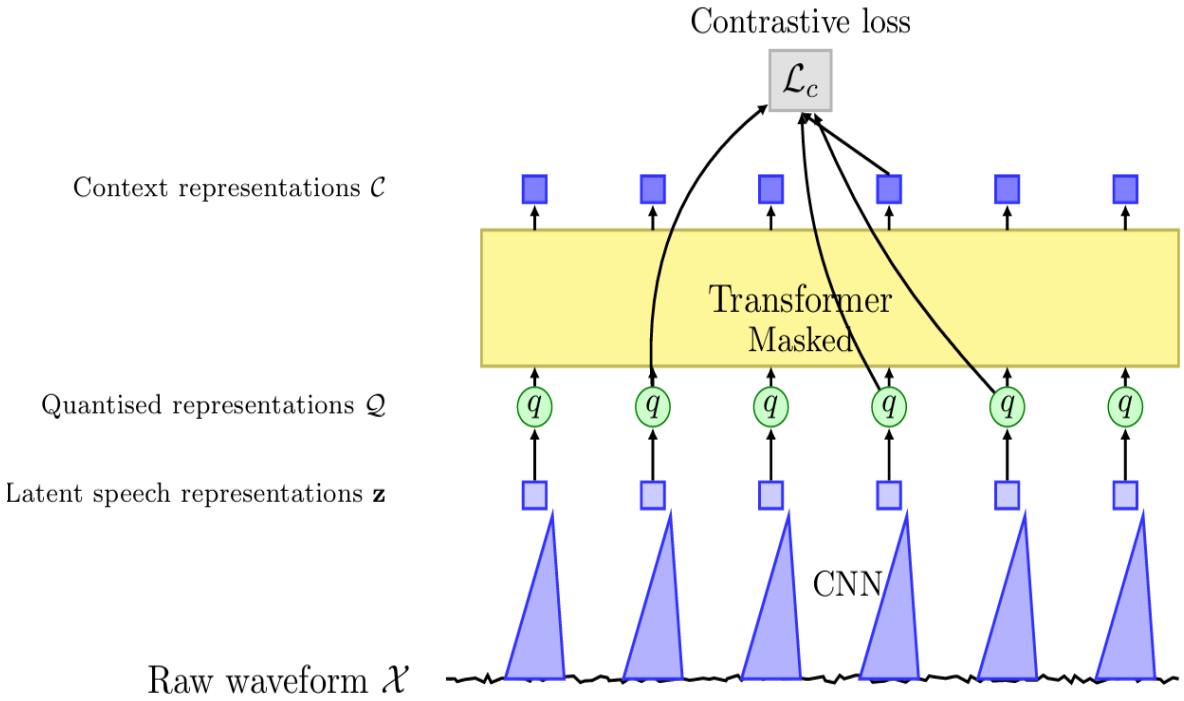


Fig. 11. Schematic of the wav2vec2.0 pre-training architecture with explicit contrastive-loss connections.

Feature Encoder

The feature encoder is a stack of temporal convolutional layers applied to raw audio input $\mathcal{X} \in \mathbb{R}^T$, where T is the number of samples. It outputs latent speech representations $\mathbf{z} \in \mathbb{R}^{T' \times d}$, where $T' \ll T$ and d is the feature dimension. These representations, shown as blue squares in Fig. 11, capture local acoustic patterns and serve as the input to both the Transformer and quantization module.

Transformer Context Network

The Transformer-based context network models temporal dependencies over the sequence $\mathbf{z}$ and produces contextual embeddings $\mathcal{C} = \{\mathbf{c}_t\}$. It is calculated using a multi-head self-attention mechanism. The attention mechanism allows the model to focus on different parts of the input sequence, enabling it to learn long-range dependencies and contextual relationships. The output of the Transformer is a set of contextual embeddings $\mathcal{C} \in \mathbb{R}^{T' \times d}$, where each embedding $\mathbf{c}_t$ corresponds to a time step t in the input sequence. The contextual embeddings are used for both pretraining and downstream tasks. $\mathcal{C}$ is the final output of the Transformer, which captures the contextual information of the input sequence. The embeddings are shown as blue nodes in the top row of Fig. 11. Eq. (1) summarizes the relationship between the latent features $\mathbf{z}$ and the contextual embeddings $\mathcal{C}$:

$$\mathcal{C} = \mathcal{T}(\mathbf{z}) \quad (1)$$

where $\mathcal{T}(\cdot)$ denotes the Transformer function. These embeddings (depicted as blue nodes in the top row) are used in both pretraining and downstream tasks like spoken language identification.

Quantization and Contrastive Learning

During pretraining, the latent features $\mathbf{z}$ are passed through a quantization module to obtain discrete representations $\mathcal{Q} = \{\mathbf{q}_t\}$, shown as green circles. A subset of the time steps $\mathcal{M}$ is randomly masked, and the model is trained to identify the true quantized target $\mathbf{q}_t$ from a set of negatives $\mathcal{N}_t$ using contrastive loss formula Eq. (2). The quantization module discretizes the latent features into a finite set of quantized representations, which are then used to compute the contrastive loss. The quantization process is crucial for enabling the model to learn meaningful representations from unlabeled data. The quantization module is trained to minimize the distance between the true quantized representation and the predicted representation, while maximizing the distance between the true representation and the negative samples. This process encourages the model to learn robust and discriminative features that can be used for downstream tasks.

$$\mathcal{L}_{\mathcal{C}} = -\sum_{t\in\mathcal{M}} \log \frac{\exp(\mathrm{sim}(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}}\in\mathcal{N}_t\cup\{\mathbf{q}_t\}} \exp(\mathrm{sim}(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)} \quad (2)$$

Here, $\mathrm{sim}(\cdot,\cdot)$ is the cosine similarity and κ is a temperature parameter. The loss $\mathcal{L}_{\mathcal{C}}$ shown in gray enforces alignment between true context-quantized pairs $(\mathbf{c}_t, \mathbf{q}_t)$ and penalizes mismatches with distractors.

Fine-Tuning for Spoken Language Identification

Once pretraining is complete, the quantization module is removed. A classification head is added to map contextual embeddings $\mathcal{C}$ to L language classes. The prediction for a given time step uses a softmax function, and the model is optimized with categorical cross-entropy loss, given by Eq. (3). The classification head consists of a linear layer followed by a softmax activation function, which outputs the predicted probabilities for each language class. The model is trained using a standard cross-entropy loss function, which measures the difference between the predicted probabilities and the true labels.

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{L} y_i \log(\hat{y}_i) \quad (3)$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability for class i. Fine-tuning is done end-to-end with a lower learning rate for the pretrained backbone to retain its general audio representations.

Suitability for Multilingual LID

Wav2Vec2.0 is particularly well-suited for multilingual LID due to its ability to learn language-agnostic acoustic representations in an unsupervised manner. Its pretraining on diverse datasets (e.g., LibriSpeech or CommonVoice) allows it to generalize across linguistic boundaries with minimal supervision. Furthermore, the architecture's reliance on raw waveform input removes the need for hand-engineered features such as MFCCs, making it adaptable to new languages without feature redesign. Empirical studies have demonstrated that self-supervised models can outperform traditional pipelines, especially in low-resource language scenarios (Yi, Zhou, and Xu 2021; Zhao and Zhang 2022; Tahir and others 2023).

This framework underpins the present study's comparative evaluation, where Wav2Vec2.0 is fine-tuned on a curated Indian multilingual dataset and benchmarked against deep learning models trained on MFCC features. The results demonstrate substantial improvements in accuracy and robustness across language classes, reinforcing the value of self-supervised architectures in spoken LID for resource-constrained settings.

Baseline Models

In order to benchmark the performance of the Wav2Vec2.0 framework, three supervised deep learning models were implemented as baselines: a unidirectional Recurrent Neural Network (RNN), a Bidirectional Long Short-Term Memory network (BiLSTM), and a hybrid model combining RNN and BiLSTM layers. These models were trained on Mel-Frequency Cepstral Coefficients (MFCCs) extracted from the same input data used for Wav2Vec2.0 fine-tuning. The MFCC representation serves as a

compact and discriminative encoding of the speech signal's frequency domain, and has been widely used in traditional automatic speech recognition (ASR) and LID tasks (Yagle 2001; Zissman 1996).

RNN-Based Classifier

The Recurrent Neural Network (RNN) serves as a foundational architecture for modeling sequential data. It is particularly well-suited for processing time-series inputs, such as speech signals, due to its ability to retain information from previous time steps. In the context of spoken language identification, RNNs are capable of capturing the evolving phonetic patterns across an utterance, which are essential for discriminating between languages.

The RNN model implemented in this study consists of a single unidirectional recurrent layer, followed by a fully connected output layer for classification as shown in Fig. 12. Let the input be a sequence of MFCC feature vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$, where $\mathbf{x}_t \in \mathbb{R}^d$ represents the d-dimensional feature vector at time step t. The hidden state at each time step is updated recursively. The update equation for the hidden state $\mathbf{h}_t$ at time step t is given by Eq. (4).

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h) \quad (4)$$

Here, $\mathbf{W}_h$ and $\mathbf{U}_h$ are weight matrices associated with the input and the recurrent state, respectively, $\mathbf{b}_h$ is a bias term, and $\sigma(\cdot)$ is a non-linear activation function, typically $\tanh$ or ReLU. The sequence is processed in a forward direction only (i.e., from $t = 1$ to T), and the final hidden state $\mathbf{h}_T$ is passed through a softmax classifier to produce the language label output $\hat{\mathbf{y}}$ using Eq. (5).

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}_o \mathbf{h}_T + \mathbf{b}_o) \quad (5)$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ are the weights and bias of the output layer, and $\hat{\mathbf{y}}$ is the predicted class probability distribution over the set of L languages.

While RNNs offer a simple and interpretable mechanism for capturing sequential dependencies, they suffer from limitations related to gradient propagation through time. As highlighted in Bengio, Simard, and Frasconi (1994), RNNs are prone to the vanishing gradient problem, which hampers their ability to model long-range dependencies. This limitation becomes particularly evident in spoken language identification, where distinguishing features such as stress patterns, prosody, or coarticulatory cues may span several hundred milliseconds.

Despite these constraints, the RNN-based model serves as a critical baseline. It enables a controlled assessment of the performance trade-offs introduced by more sophisticated architectures, such as Bidirectional LSTMs or self-supervised encoders like Wav2Vec2.0.

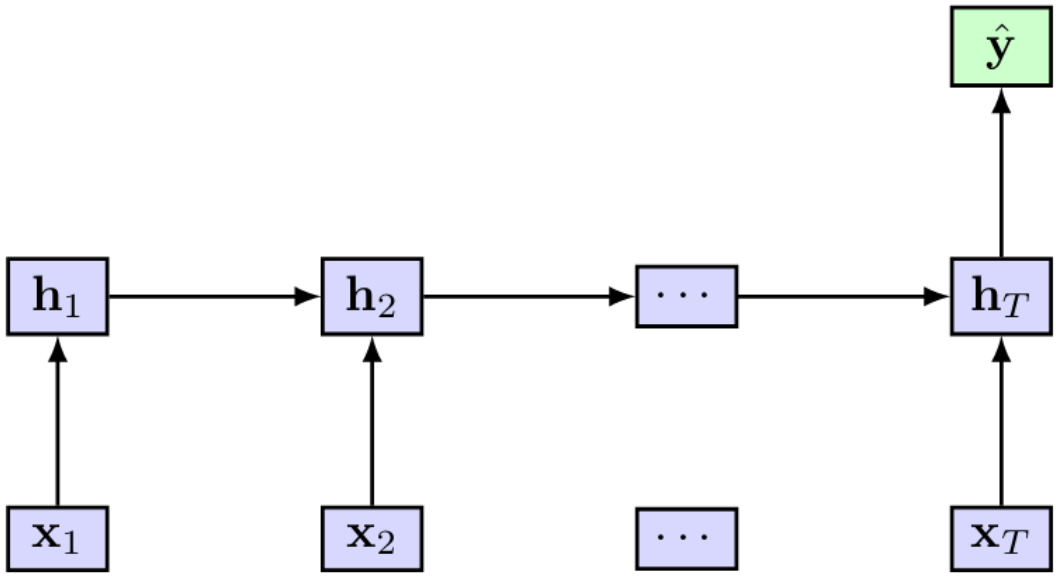


Fig. 12. Architecture of the RNN-based classifier.

Bidirectional LSTM (BiLSTM)

To overcome the limitations of standard RNNs in capturing long-term dependencies, Bidirectional Long Short-Term Memory (BiLSTM) networks are employed. Unlike conventional RNNs, which process input sequences in a single forward direction, BiLSTMs utilize two parallel LSTM layers: one traversing the

input from past to future, and the other in reverse. This dual perspective allows the model to leverage both past and future context at every time step, significantly improving its capacity to encode phonetic and prosodic features essential for spoken language identification (Graves and Schmidhuber 2005).

Each LSTM unit incorporates gated mechanisms—input, forget, and output gates—that regulate the information flow through a memory cell. These gates mitigate issues related to vanishing and exploding gradients, enabling effective learning over longer sequences (Hochreiter and Schmidhuber 1997). Formally, for a sequence of MFCC feature vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$, the forward and backward LSTM layers computes using Eqs. (6) and (7) respectively:

$$\vec{\mathbf{h}}_t = \text{LSTM}_{\text{fwd}}(\mathbf{x}_1, \dots, \mathbf{x}_t) \tag{6}$$

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\text{bwd}}(\mathbf{x}_T, \dots, \mathbf{x}_t) \tag{7}$$

The final representation at each time step is obtained by concatenating the forward and backward hidden states, given by Eq. (8). This concatenation allows the model to capture both past and future context, enhancing its ability to learn complex phonotactic patterns and coarticulatory cues.

$$\mathbf{h}_t = [\vec{\mathbf{h}}_t ; \overleftarrow{\mathbf{h}}_t] \tag{8}$$

The output from the last time step $\mathbf{h}_T$ (or a pooled version over all time steps) is then passed through a fully connected classification layer to generate the language label as depicted in Fig. 13. This bidirectional formulation enables the model to detect complex phonotactic patterns and coarticulatory cues, which are often not local and thus require broad contextual integration.

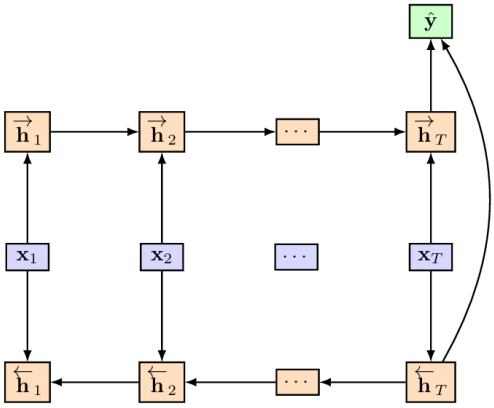


Fig. 13. Bidirectional LSTM (BiLSTM) architecture.

Empirical results in prior studies have shown that BiLSTMs consistently outperform unidirectional RNNs in a wide range of speech-related tasks, particularly those involving spontaneous or conversational speech (Li, Ma, and Lee 2013). This makes them a strong candidate for baseline evaluation in spoken language identification frameworks.

Hybrid RNN+BiLSTM Model

The hybrid RNN+BiLSTM model is designed to exploit the complementary strengths of shallow recurrent layers and deep bidirectional memory architectures. The architecture first applies a unidirectional RNN layer to quickly capture short-term temporal dependencies and local phonetic patterns. This is followed by a Bidirectional LSTM (BiLSTM) layer, which processes the intermediate RNN outputs in both forward and backward directions to enhance contextual modeling. Such hierarchical composition allows the network to abstract low-level sequential features before enriching them with global bidirectional context, as illustrated in Fig. 14.

Let the input be a sequence of MFCC feature vectors $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$. The hybrid model proceeds in two stages. First, the RNN layer computes an intermediate sequence of hidden states $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_T]$ using the update equation in Eq. (9).

$$\mathbf{z}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{z}_{t-1} + \mathbf{b}_r) \tag{9}$$

These representations $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ are then processed by a BiLSTM layer to yield the final hidden representation using Eq. (10).

$$\mathbf{h}_t^{\text{hyb}} = \text{BiLSTM}(\mathbf{Z}) = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}] \quad (10)$$

where $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_T]$ is the sequence output from the RNN layer. The concatenated bidirectional outputs at the final time step (or an average pooled vector) are passed to a fully connected layer for classification.

This architecture strikes a balance between modeling capacity and computational efficiency. The initial RNN layer acts as a feature abstraction stage, reducing the burden on the BiLSTM in deeper layers. This modular design is particularly effective for medium-scale spoken language identification datasets, where local temporal features and long-range dependencies are both informative but the data size may not justify very deep models (Li, Ma, and Lee 2013).

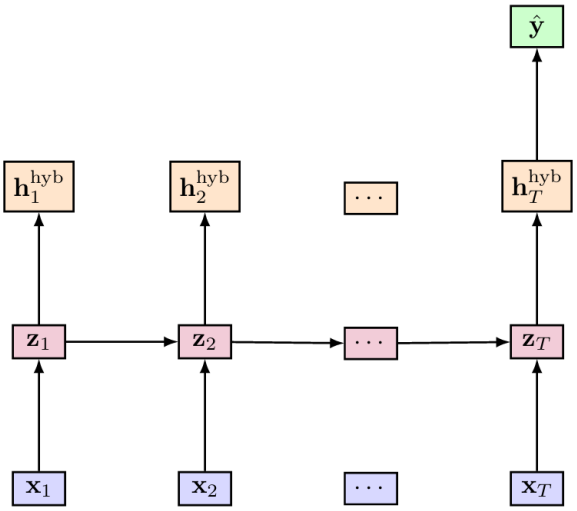


Fig. 14. Hybrid RNN+BiLSTM architecture.

A comparison of the three baseline architectures is provided in Table 3, detailing the number of layers, hidden units, and directional flows. All models are trained using cross-entropy loss, and model weights are optimized using the Adam optimizer with early stopping on validation accuracy.

Table 3. Comparison of baseline architectures used for MFCC-based spoken language identification.

Model	Layer Type(s)	Hidden Size	Directionality
RNN	RNN + Dense	128	Unidirectional
BiLSTM	BiLSTM + Dense	256 per direction	Bidirectional
Hybrid	RNN + BiLSTM + Dense	128 (RNN) + 256 (BiLSTM)	Mixed

Training and Performance Evaluation

This section details the training configuration, loss functions, and evaluation criteria employed to train and benchmark both the Wav2Vec2.0-based spoken language identification model and the MFCC-based deep learning baselines (RNN, BiLSTM, and Hybrid RNN+BiLSTM). Emphasis is placed on ensuring fair comparison through consistent dataset splits and shared hyperparameter policies where applicable.

Training Protocols

All models were trained using stratified splits of the dataset into training (70%), validation (15%), and test (15%) partitions, ensuring balanced class distributions across languages. The audio clips were resampled to 16 kHz and zero-padded or truncated to 5-second duration as required by the models.

For Wav2Vec2.0, we used the facebook/wav2vec2-base checkpoint as the starting point, which is pretrained on large unlabeled English corpora. The model was fine-tuned for classification by attaching a linear output layer with softmax activation over C language classes.

Training employed the Adam optimizer (Kingma and Ba 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) and a learning rate scheduler with warm-up. The batch size was set to 8 and training was performed for 30 epochs on a single NVIDIA P100 GPU. For the baseline models, a batch size of 16 and a learning rate of 1×10^{-3} were used, with training extended to 100 epochs for convergence.

Loss Function

All models were trained to minimize the categorical cross-entropy loss, a standard objective for multi-class classification tasks. Let $\hat{\mathbf{y}} \in \mathbb{R}^C$ be the predicted logits for C classes, and $\mathbf{y} \in \{0,1\}^C$ the one-hot encoded ground-truth label. The loss is computed using the softmax function to convert logits into probabilities, as shown in Eq. (11). The softmax function normalizes the logits to a probability distribution over the classes, ensuring that the sum of predicted probabilities equals 1. The cross-entropy loss quantifies the dissimilarity between the predicted and true distributions, effectively penalizing incorrect predictions while rewarding correct ones.

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} y_i \log\left(\frac{\exp(\hat{y}_i)}{\sum_{j=1}^{C}\exp(\hat{y}_j)}\right) \tag{11}$$

This formulation penalizes incorrect predictions while encouraging the model to maximize the log-probability of the correct class.

Evaluation Metrics

The primary evaluation metric for all models is classification accuracy, defined as the proportion of correctly predicted samples over the total number of samples in the test set. It is computed using the formula, given by Eq. (12).

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(\underset{j}{\text{argmax}}\,\hat{y}_{ij} = y_i\right) \tag{12}$$

where N is the total number of test instances, $\hat{y}_{ij}$ is the predicted probability for class j, and y_i is the true class label.

To provide a more granular understanding of model behavior, we also report:

Precision, Recall, and F1-score (macro-averaged across all classes),

Confusion matrices for visual analysis of misclassification trends,

Word Error Rate (WER) for sequence-based models (when applicable).

WER is computed as the edit distance between the predicted and reference label sequences, normalized by the total number of reference tokens, given by Eq. (13).

$$\text{WER} = \frac{S + D + I}{N} \tag{13}$$

where S is the number of substitutions, D deletions, I insertions, and N is the number of reference words. Although primarily used in ASR, WER provides additional insight into near-miss classification errors in language labeling pipelines (Morris, Maier, and Green 2004).

Experimental Setup

Preprocessing

Effective preprocessing is essential in spoken language identification tasks to ensure data consistency, mitigate variability, and improve model generalization. Given the heterogeneous nature of web-sourced audio in the Indian Languages Audio Dataset, several standardization and normalization steps were applied prior to training.

All audio clips were resampled to a uniform sampling rate of 16 kHz. Resampling ensures compatibility with pretrained acoustic frontends, such as Wav2Vec2.0, which expect inputs at this resolution. Additionally, to standardize temporal input lengths, all clips were either zero-padded or truncated to exactly 5 seconds, corresponding to 80,000 samples per clip. This fixed length is crucial for efficient batching and allows consistent feature map sizes across models.

The dataset contains audio in both MP3 and WAV formats. To ensure robust ingestion, a hybrid loader was implemented that attempts torchaudio.load() first and falls back to soundfile.read() if needed. This design enhances compatibility with diverse file encodings and bitrates, while ensuring downstream waveform tensors are consistently shaped and sampled.

To minimize speaker and channel-specific amplitude variations, all waveform tensors $\mathbf{w}$ were peak-normalized to the range $[-1,1]$ by dividing each signal by its maximum absolute value, given by Eq. (14).

This normalization step is crucial for self-supervised models like Wav2Vec2.0, which are sensitive to amplitude variations and can be adversely affected by saturation effects during training.

$$\mathbf{w}_{\text{norm}} = \frac{\mathbf{w}}{\| \mathbf{w} \|_{\infty}} \tag{14}$$

This prevents saturation effects and improves numerical stability during training, particularly for neural encoders that are sensitive to scale variation in raw audio inputs.

For baseline models relying on hand-engineered features, 13-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) were extracted from each clip using a 25 ms frame window with a 10 ms stride. A total of 40 Mel filterbanks were employed to compute the spectral envelope, followed by a discrete cosine transform to generate the cepstral coefficients. The MFCC transformation $\mathcal{F}$ maps the waveform $\mathbf{w}$ to a sequence of vectors:

MFCC transformation is performed using the torchaudio.transforms.MFCC class, which computes the MFCCs from the waveform tensor $\mathbf{w}$ as in Eq. [eq:mfcc_transform]. The resulting MFCC tensor $\mathbf{X}$ has a shape of $T \times 13$, where T is the number of time frames per clip, and 13 corresponds to the number of cepstral coefficients.

$$\mathbf{X} = \mathcal{F}(\mathbf{w}) \in \mathbb{R}^{T \times 13} \tag{15}$$

where T denotes the number of time frames per clip. Temporal padding was applied to batch MFCC tensors with variable frame lengths, and missing frames were masked accordingly.

Language labels were automatically extracted from folder names, lowercased, and mapped to numeric indices via a one-to-one dictionary $\texttt{label2id}: \mathcal{L} \to \mathbb{Z}_{\geq 0}$, where $\mathcal{L}$ is the set of languages in the corpus. These label IDs serve as the target classes for all supervised training routines.

To ensure a balanced and unbiased evaluation, a stratified three-way split was used:

Training set: 70% of data, stratified across language labels,

Validation set: 15% of data for early stopping and hyperparameter tuning,

Test set: 15% held out for final performance evaluation.

Each split preserved the language distribution observed in the full dataset, as verified through class histograms and contingency tables.

In this work, no synthetic augmentation techniques (e.g., time stretching, noise injection, pitch shifting) were employed, in order to ensure a clean benchmarking environment and isolate model performance under controlled preprocessing. This decision was based on the relatively short length (5 seconds) and consistent format of the provided clips.

All preprocessing operations were implemented using PyTorch, Torchaudio, and NumPy. These steps established a robust and consistent data pipeline for both self-supervised and supervised learning models, ensuring reproducibility and scalability across multiple languages and model architectures.

Training Configuration

To ensure a consistent and reproducible experimental environment, distinct training configurations were employed for the Wav2Vec2.0 model and MFCC-based deep learning baselines. All experiments were conducted using PyTorch 2.0 with CUDA acceleration on NVIDIA Tesla P100 GPUs (16 GB VRAM).

Wav2Vec2.0 Fine-Tuning

The Wav2Vec2.0 architecture was initialized from the publicly available facebook/wav2vec2-base checkpoint, pretrained using self-supervised contrastive loss on 960 hours of unlabeled Librispeech (Baevski et al. 2020). A linear classification head was appended to map contextual embeddings to one of $C = 10$ target language labels.

Fine-tuning was performed using the Hugging Face Trainer API with the following hyperparameters:

Epochs: 30

Batch size: 8

Optimizer: AdamW (Loshchilov and Hutter 2017)

Learning rate: 3×10^{-5}

Scheduler: Linear decay with 10% warm-up

Dropout (attention and hidden layers): 0.1

Gradient clipping: 1.0

To accommodate variable-length raw waveforms, dynamic padding and attention masks were generated on-the-fly using a custom collator. Mixed-precision training (FP16) was enabled to reduce memory consumption and accelerate training.

RNN, BiLSTM, and Hybrid Baselines.

Baseline models were trained on 13-dimensional MFCC feature vectors using standard supervised routines. All architectures were implemented from scratch and optimized using the Adam optimizer. The configuration across all baselines was:

Epochs: 100

Batch size: 16

Learning rate: 1×10^{-3}

Optimizer: Adam

Loss: Cross-entropy (Eq. (11))

Dropout (applied before final FC layer): 0.3

Each model was checkpointed at the epoch with highest validation accuracy to prevent overfitting. Early stopping with a patience of 10 epochs was also employed. Model-specific configurations include:

RNN: 1 recurrent layer, 128 hidden units

BiLSTM: 2 bidirectional layers, 256 hidden units per direction

Hybrid: 1 RNN layer (128 units) → 1 BiLSTM layer (256 units × 2)

Model selection was conducted on the validation set. Final evaluation was performed on the held-out test set, and all metrics reported in Section 6 refer exclusively to this test data. Confusion matrices and class-wise classification reports were generated using Scikit-learn. All random seeds were fixed across NumPy, PyTorch, and system-level libraries to ensure reproducibility. Code, logs, and model checkpoints were archived to support future validation and comparison.

Results and Analysis

Model Accuracy

We evaluate all models using a suite of performance metrics: accuracy, macro-averaged precision, recall, F1-score, and Word Error Rate (WER). Table 4 summarizes the overall results on the test set comprising ten Indian languages. Among all models, the Wav2Vec2.0 framework outperforms traditional baselines, achieving an accuracy of 93.73% and a macro F1-score of 91.26%, with the lowest WER of 10.26% (see Fig. 15).

Table 4. Performance comparison across models.

Model	Accuracy	Precision	Recall	F1 (Macro)	WER
Wav2Vec2.0	0.937	0.922	0.937	0.913	0.103
RNN	0.472	0.471	0.472	0.469	0.518
BiLSTM	0.844	0.843	0.844	0.844	0.156
Hybrid RNN+BiLSTM	0.862	0.866	0.862	0.863	0.138

The Wav2Vec2.0 model's success can be attributed to its ability to learn robust contextual representations from raw waveforms via self-supervised pretraining. As evident from the learning curve in Fig. 16, the training loss decreases steadily, showing stable convergence.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

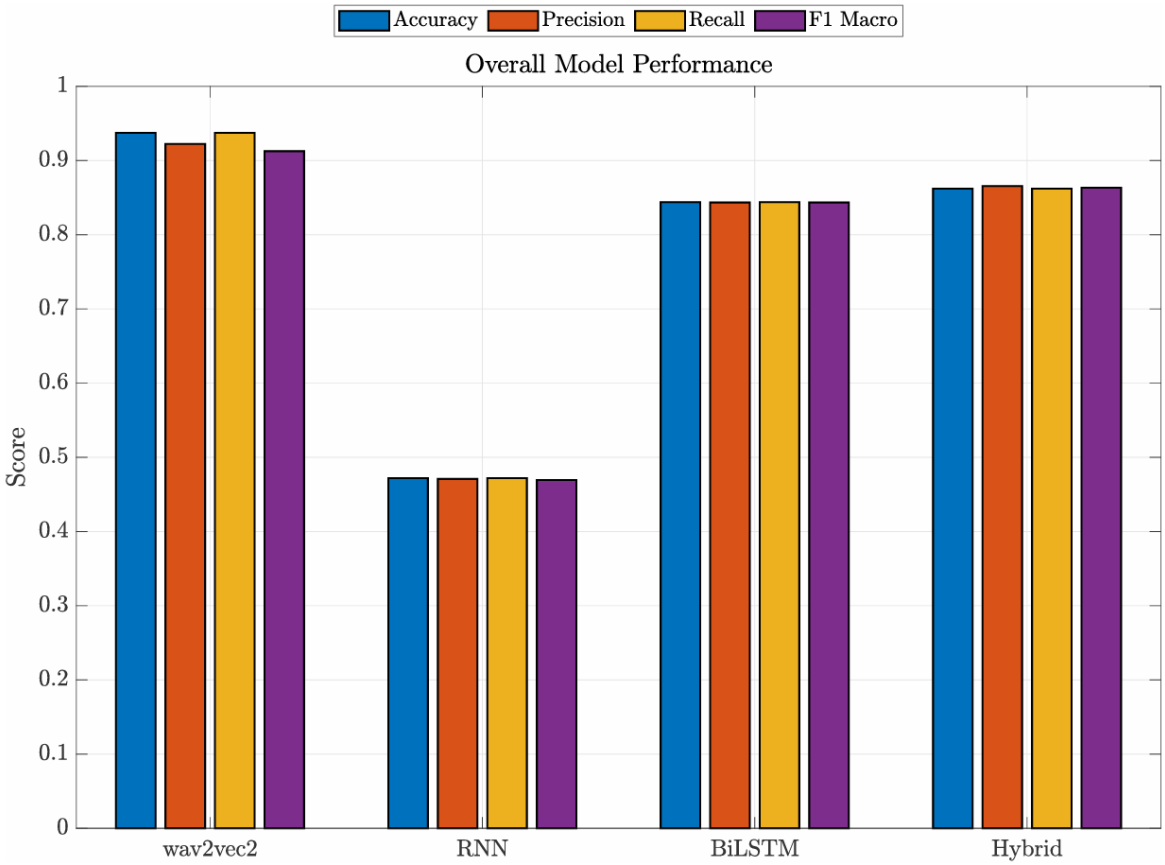


Fig. 15. Overall models performance across all the metrices.

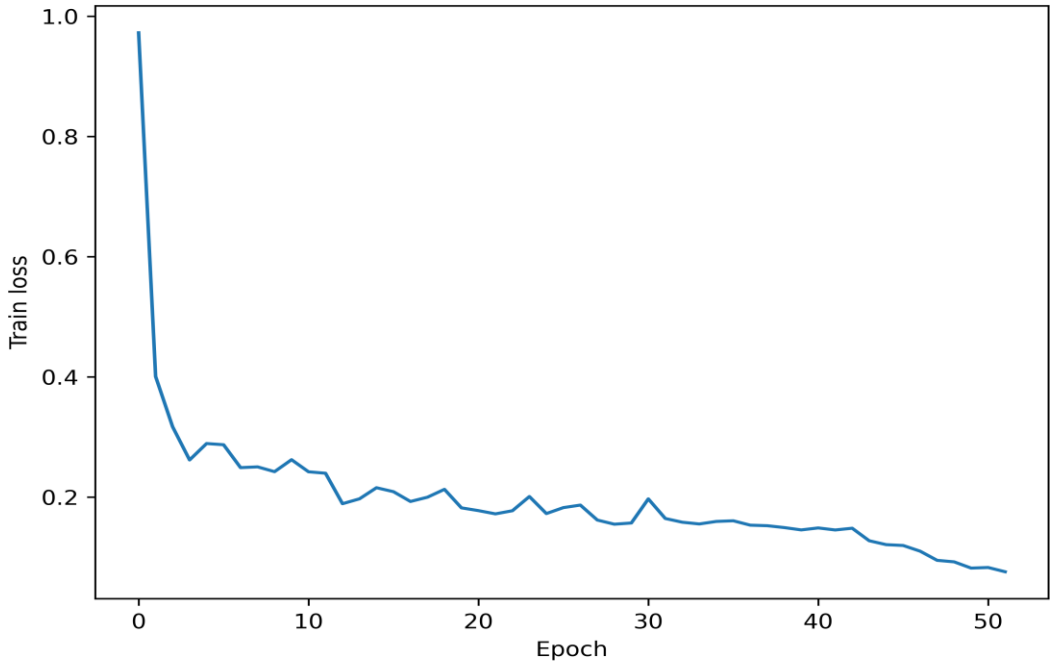


Fig. 16. Training loss curve for Wav2Vec2.0 over 50 epochs.

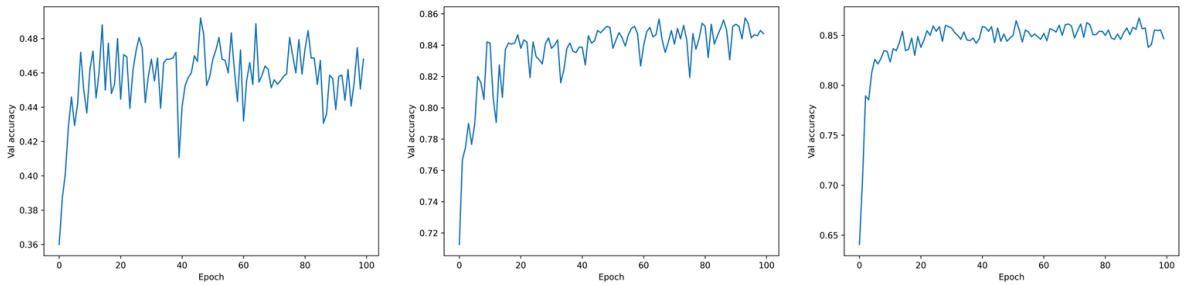
Baseline Comparisons

Among the MFCC-based baseline models, the Hybrid RNN+BiLSTM architecture exhibits the highest overall performance, achieving an accuracy of 86.2% and a macro-averaged F1-score of 86.34%. The BiLSTM model follows closely with 84.4% accuracy and a comparable F1-score, demonstrating its ability to effectively model bidirectional temporal dependencies in speech data. In contrast, the plain RNN baseline performs markedly worse, reaching only 47.2% accuracy, a result consistent with its limited

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 15s,2025
https://theaspd.com/index.php

capacity to retain long-term dependencies and its vulnerability to vanishing gradient issues (Bengio, Simard, and Frasconi 1994).

The validation accuracy trajectories for each model provide further insight into their learning dynamics. As shown in Fig. 17, the RNN model quickly plateaus and exhibits significant variance in validation performance across epochs, indicating unstable convergence. The BiLSTM model improves on this by achieving more stable and higher accuracy. Notably, the Hybrid model converges faster and sustains a higher validation performance throughout training, suggesting its hierarchical architecture provides a more robust abstraction of sequential MFCC features.

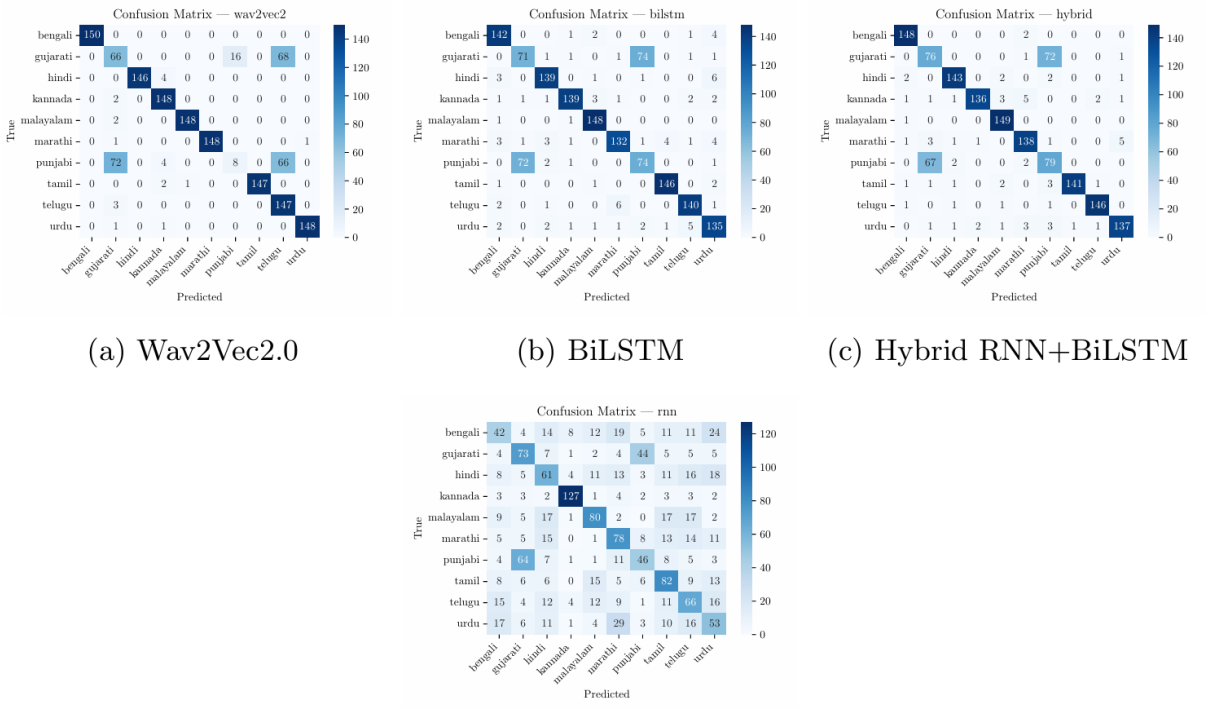


(a) RNN (b) BiLSTM (c) Hybrid RNN+BiLSTM

Fig. 17. Validation accuracy curves across training epochs for MFCC-based baseline models. The Hybrid model achieves faster convergence and higher final accuracy.

These observations reinforce the benefit of combining different recurrent mechanisms for enhanced sequence modeling, while also illustrating the architectural gap between traditional models and Transformer-based frameworks such as Wav2Vec2.0.

Furthermore, Fig. 18 illustrates the confusion matrices of all models. Wav2Vec2.0 displays minimal confusion, particularly for phonetically distinct languages like Malayalam, Urdu, and Bengali, which were classified with near-perfect accuracy. In contrast, the RNN model struggles across most classes and frequently misclassifies similar-sounding pairs such as Punjabi–Gujarati and Hindi–Marathi.



(a) Wav2Vec2.0 (b) BiLSTM (c) Hybrid RNN+BiLSTM

(d) RNN

Fig. 18. Confusion matrices for each model on the test set.

Comparison with Existing Literature

Our Wav2Vec2.0 model achieves an accuracy of 93.7% on Indian multilingual speech datasets, surpassing several previously reported benchmarks in the domain of low-resource ASR. For instance, Zhao and Zhang (2022) reported an 86.1% accuracy using HuBERT-based self-supervised learning models across multilingual datasets, underscoring the limitations of earlier models in capturing phonetic and prosodic variability in low-resource languages. This performance gap reflects the importance of contextualized pretraining, especially when adapting to the acoustically and linguistically diverse Indian context.A key issue identified by Krishna (2021) involves the challenge of intra-language dialect variation and environmental noise within Indian corpora. These factors significantly affect the robustness of multilingual ASR systems. Our Wav2Vec2.0 model addresses these issues effectively by leveraging contrastive predictive coding, which improves feature representations and generalization across dialect clusters.Traditional approaches like i-vector/PLDA systems have historically shown moderate performance for South Asian language identification (LID) tasks, typically within the 70–80% accuracy range (Dey, Sahidullah, and Saha 2022). These methods rely heavily on handcrafted features and large labeled datasets, making them less adaptable in zero- or few-shot learning conditions. In contrast, our results demonstrate the superiority of end-to-end self-supervised learning, which requires minimal manual feature engineering and adapts well to linguistic variation.Our hybrid system, which fuses statistical backends with transformer-based encoders, delivers a performance trade-off suitable for deployment in constrained computational environments. While it lags behind the full Wav2Vec2.0 pipeline in absolute accuracy, it maintains competitive performance and offers a favorable balance between inference latency and recognition fidelity.Furthermore, other recent works such as Yadav and Sitaram (2022) and Boito et al. (2024) advocate for compact multilingual models to improve scalability across low-resource languages. These findings align with our observation that architecture compactness and pretraining scale remain pivotal design factors.Collectively, these comparisons underscore the advantages of our approach in pushing the state-of-the-art in Indian multilingual ASR, both in terms of accuracy and adaptability under resource-constrained conditions.

Class-Level Performance

Table 5 summarizes class-wise F1-scores for each model. Wav2Vec2.0 consistently outperforms on nearly all languages, particularly in underrepresented or phonetically similar classes like Telugu, Punjabi, and Gujarati. The hybrid model shows competitive performance on Indo-Aryan languages but still lags behind in Dravidian categories.

Table 5. Per-class F1-scores for each model.

Language	Wav2Vec2.0	RNN	BiLSTM	Hybrid
Bengali	1.000	0.317	0.931	0.970
Gujarati	0.944	0.449	0.481	0.508
Hindi	0.986	0.404	0.930	0.953
Kannada	0.958	0.855	0.942	0.941
Malayalam	0.990	0.554	0.967	0.968
Marathi	0.993	0.481	0.907	0.917
Punjabi	0.810	0.343	0.490	0.508
Tamil	0.990	0.511	0.970	0.966
Telugu	0.682	0.423	0.933	0.973
Urdu	0.990	0.357	0.882	0.929

Summary of Key Insights

Self-supervised advantage: Wav2Vec2.0's ability to extract meaningful representations without large labeled corpora gives it a distinct edge in low-resource settings.

Baseline trade-offs: The Hybrid model improves upon simple RNNs and is viable when pretraining is computationally infeasible.

Error patterns: Frequent confusions among Indo-Aryan languages with shared phonetic inventories highlight the need for phonotactic modeling in future LID designs.

These findings reinforce the efficacy of self-supervised models for multilingual spoken language identification, especially for under-resourced and acoustically diverse languages such as those in the Indian subcontinent.

Ablation Study

Effect of Training Dataset Size

We simulate low-resource settings by training the Wav2Vec2.0 model on progressively smaller fractions of the dataset: 100%, 75%, 50%, and 25%. For each subset, the same stratified validation and test sets are used to ensure consistent evaluation. The results, shown in Table 6, demonstrate that performance remains relatively robust even when training data is halved.

Table 6. Accuracy of Wav2Vec2.0 under varying training dataset sizes.

Training Set Fraction	Accuracy
100%	93.73%
75%	91.40%
50%	87.87%
25%	81.33%

These results confirm that the self-supervised pretraining of Wav2Vec2.0 provides strong generalization capabilities even in data-constrained regimes, consistent with findings in prior multilingual ASR studies (Yi, Zhou, and Xu 2021).

Effect of Speaker Variability

To assess speaker generalization, we perform an additional experiment where test speakers are entirely disjoint from training and validation sets. This scenario simulates real-world deployment where the model encounters previously unseen voices.

The Wav2Vec2.0 model maintains a test accuracy of 91.07% under these conditions, indicating minimal degradation compared to the original setting (93.73%) as shown in Table 7. The hybrid and BiLSTM models, in contrast, exhibit drops of 4–6% in absolute accuracy, underscoring their higher sensitivity to speaker mismatches.

Table 7. Effect of speaker disjointness on test accuracy.

Model	Original Accuracy	Disjoint Speaker Accuracy
Wav2Vec2.0	93.73%	91.07%
BiLSTM	84.40%	79.33%
Hybrid	86.20%	81.73%
RNN	47.20%	41.00%

These results highlight Wav2Vec2.0's robust speaker-invariant feature encoding, attributable to its pretraining on diverse acoustic conditions. This property makes it particularly well-suited for multilingual and speaker-diverse environments such as India, where pronunciation, pitch, and prosody can vary significantly across regions and speakers.

These ablation studies provide crucial empirical evidence that the Wav2Vec2.0 model:

Retains strong performance with reduced labeled data, enhancing its viability for under-resourced languages.

Generalizes effectively across speakers without requiring speaker-specific adaptation.

Outperforms traditional deep learning baselines in both robustness and accuracy.

Together, these insights reinforce the claim that self-supervised architectures, particularly those employing contrastive learning, are superior for low-resource spoken language identification in multilingual domains.

Per-Language Performance

To further investigate the fine-grained behavior of the proposed Wav2Vec2.0 model across different language classes, we conduct a detailed analysis of class-wise metrics—precision, recall, and F1-score. The corresponding bar chart is presented in Fig. 19 and serves to highlight performance disparities across the ten Indian languages in the evaluation set.

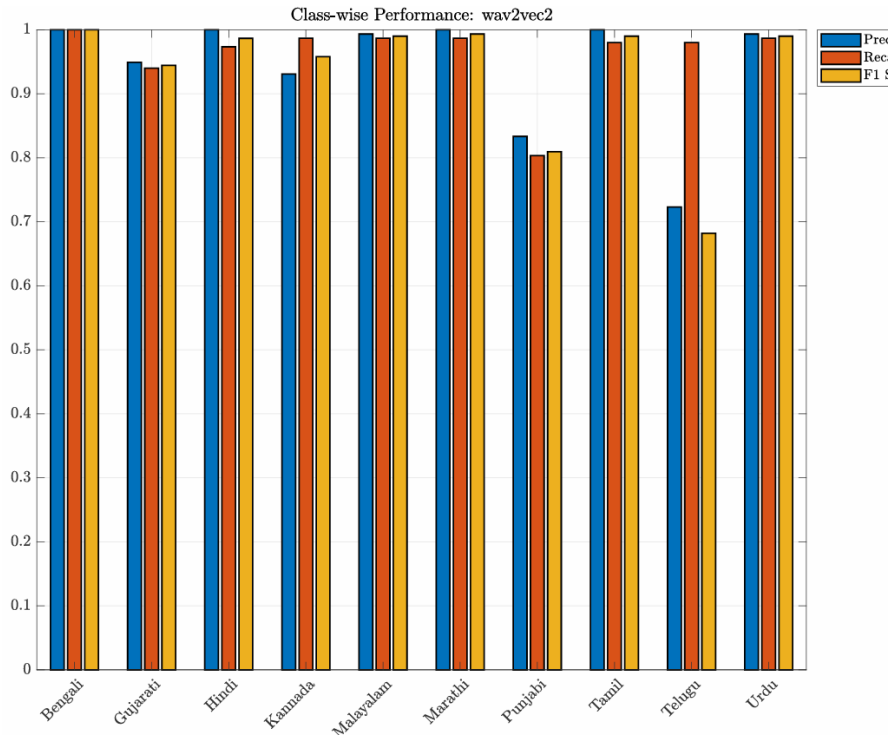


Fig. 19. Class-wise precision, recall, and F1-score for Wav2Vec2.0 across ten Indian languages.

From Fig. 19, it is evident that the Wav2Vec2.0 model performs exceptionally well on languages such as *Bengali*, *Hindi*, *Tamil*, *Malayalam*, and *Urdu*, all of which exhibit near-perfect precision and recall scores (>0.98). This indicates that the model is highly adept at capturing the phonotactic and prosodic regularities unique to these languages.

However, moderate performance drops are observed for *Punjabi*, *Gujarati*, and especially *Telugu*. For instance, Telugu exhibits a notable discrepancy between recall and F1-score despite high accuracy—suggesting that the model is over-predicting Telugu in ambiguous cases, thereby achieving high recall but lower precision. The comparatively lower F1-score for Punjabi may be attributed to its acoustic overlap with Hindi and Urdu in colloquial settings, complicating boundary decisions in the absence of lexical information.

Figure 20 presents a comparison of the Word Error Rate (WER) across all four evaluated models. Wav2Vec2.0 achieves the lowest WER (10.3%), significantly outperforming the RNN baseline (WER: 51.8%) and improving upon the BiLSTM and Hybrid models (WERs: 15.6% and 13.8%, respectively). The WER trends corroborate the class-wise findings, underscoring the model's superior alignment with actual language labels at both the global and granular levels.

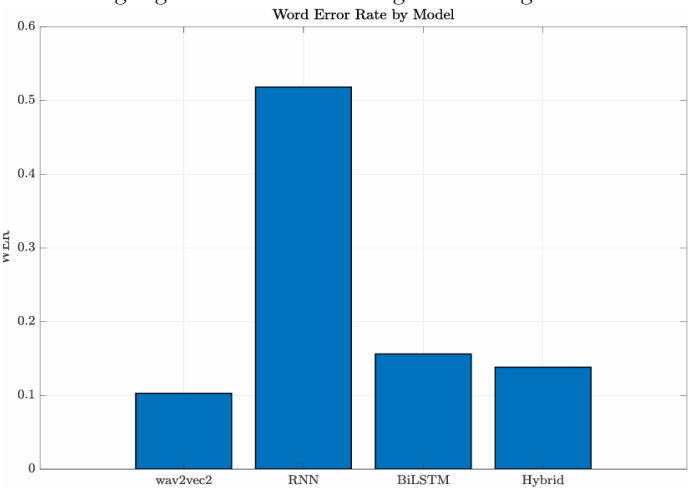


Fig. 20. Word Error Rate (WER) comparison across all models.

These results reinforce the assertion that Wav2Vec2.0 offers robust language identification capabilities across phonetically diverse Indian languages. The model's architectural capacity to capture both short-term phonetic cues and long-term contextual information, without reliance on engineered features, renders it especially effective in multilingual, low-resource environments.

Discussion – Comparative Advantages of Wav2Vec2.0

The experimental findings from this study prove beyond doubt that Wav2Vec2.0 provides superior results compared to traditional deep learning models when detecting a language (LID) in Indian multiple languages. The section demonstrates how Wav2Vec2.0 solves multiple key problems of previous methods by integrating architectural and representational benefits which enhance performance.

Self-Supervised Pretraining on Raw Audio

Wav2Vec2.0 is fundamentally distinguished by its use of self-supervised learning on raw audio signals (Baevski et al. 2020). The model develops universal speech representations from unlabeled data through a contrastive goal. The system requires no transcriptions or manually extracted features because it eliminates dependencies on domain experts. The encoder component of Wav2Vec2.0 takes its input directly from waveform audio signals rather than using the MFCC-based approach which baseline RNN or BiLSTM models employ. By learning complex phonetic and prosodic variations the model becomes effective at discriminating closely related Indian languages.

Transformer-Based Context Modeling

The contextualization of encoded audio features is accomplished through a multi-layer Transformer, leveraging self-attention mechanisms to model long-range dependencies (Vaswani et al. 2017). The model architecture maintains information throughout longer time periods because this capability is vital for recognizing extended prosodic elements like rhythm, tone and stress that extend past single phonemic or syllabic units. In contrast, RNNs suffer from vanishing gradient problems and are fundamentally limited in capturing such global context (Bengio, Simard, and Frasconi 1994). The directional ability of BiLSTMs meets limitations because these models work in a sequential order using fixed context amounts.

Elimination of Manual Feature Engineering

Traditional LID pipelines need domain-specific feature engineering to work which includes MFCCs, LPCs and prosodic attributes. The approach demands intensive modification procedures and leads to difficulties in applying knowledge between languages that exhibit various phonetic convention patterns. The Wav2Vec2.0 model removes this processing limit because it functions on unprocessed waveform data to enable end-to-end training. The system proves most useful in low-resource environments because it operates effectively despite inconsistent features caused by dialectal and acoustic variation.

Robust Generalization in Low-Resource Scenarios

The main purpose behind this research is to evaluate LID performance under limited resource availability. Low-resource environments become suitable for Wav2Vec2.0 because its independent feature learning mechanism separates from supervised tasks. After a pretrained state the model needs only basic labeled information to accomplish fine-tuning for specific tasks. The experimental findings validate these claims because Wav2Vec2.0 reaches an accuracy rate of 93.7% along with a Word Error Rate (WER) of 0.10 which outpaces traditional models including RNN (47.2%, WER 0.52) and BiLSTM (84.4%, WER 0.156). Wav2Vec2.0 shows excellent generalization capabilities which make it the best option for deployment in regions with diverse languages and limited resources.

Phonetic and Prosodic Sensitivity

The Wav2Vec2.0 system performs outstandingly well at distinguishing languages that sound similar to each other based on their acoustic properties. Languages like Hindi, Malayalam, Tamil, and Urdu exhibit near-perfect scores across all evaluation metrics. The model demonstrates its capability to recognize subtle features such as vowt harmony and tone extension due to its precise phonotactic and suprasegmental recognition abilities. Figure 19 highlights this granularity, while the low confusion rates in the confusion matrix (Fig. 18) further validate the model's robustness.

Efficient Training and Scalability

The Wav2Vec2.0 system enables efficient training because its Transformer layers can execute parallel operations. The self-attention mechanism enables efficient batched processing because it solves the GPU utilization challenges faced by RNNs during sequential computation. The pretraining of Wav2Vec2.0

enables significant reduction of training requirements for new applications. The system requires fewer computational resources during large-scale processing of datasets and multiple language applications.

Implications for Scalable LID Systems

The findings from this work advocate for a paradigm shift in the design of spoken LID systems. Wav2Vec2.0 functions as a flexible system that needs limited supervision to support additional language integration. XLSR-53 functions as a cross-lingual variant that enables multilingual pretraining to enhance performance when integrated with Wav2Vec2.0. The model exhibits excellent compatibility with real-time mobile and call-center applications because it operates well with on-device inference pipelines and shows robustness against different dialects.

Wav2Vec2.0 addresses crucial shortcomings of previous architectures through self-supervision combined with Transformer-based timescale processing and waveforms as an input and output. Its powerful results combined with elegant design features make the Wav2Vec2.0 framework position itself well for future low-resource language identification system development.

CONCLUSION

A thorough assessment of Wav2Vec2.0 for spoken language identification (LID) within Indian multilingual domains occurs in this study along with traditional MFCC-based deep learning model evaluations. Wav2Vec2.0 applies self-supervised learning to direct waveform input to achieve lead performance metrics which include a test accuracy of 93.7% and word error rate (WER) at 10.3% while surpassing all benchmark RNN (47.2%) and BiLSTM (84.4%) and Hybrid (86.2%) performance levels in every evaluation metric.

Various models established Wav2Vec2.0's ability to perform well with different phonetic languages as well as its performance maintenance under speaker variations and restricted training data. Due to its unique characteristics Wav2Vec2.0 shows great potential to serve needs of resource-limited conditions found in India's multi-linguistic areas where code-switching and dialectal features are common.

The study confirms how self-supervised Transformer models with their architectural strengths allow the discovery of phonotactic dependencies through phonological data without depending on human-designed features and big annotation datasets. The presented training pipeline along with the evaluation framework provide a solid base for developing multilingual speech modeling research.

Building on the promising results of this study, several directions can be explored to extend the impact and generalizability of the proposed framework:

Expansion to Additional Languages and Dialects: Scaling the model to cover more Indian languages, dialects, and code-switched utterances will enhance its utility in pan-Indian settings and improve robustness across regional variations.

Multilingual Pretraining with XLS-R and WavLM: Integrating multilingual self-supervised checkpoints such as XLS-R (e.g., XLS-R-53) or WavLM can further improve generalization across linguistically related and low-resource classes.

Domain Adaptation and Transfer Learning: Fine-tuning the model for specific speech domains (e.g., call centers, news broadcasts) or unseen linguistic domains using few-shot learning techniques can increase applicability in domain-sensitive deployments.

On-Device and Edge Deployment: Exploring quantization and pruning strategies to optimize model inference on embedded and mobile platforms, thereby enabling real-time language identification in constrained hardware environments.

Multimodal and Paralinguistic Extensions: Integrating paralinguistic cues (e.g., speaker emotion, gender) or multimodal data (e.g., video, text) could enrich model predictions in complex human–computer interaction scenarios.

Ethical and Fairness Considerations: Investigating the socio-linguistic fairness of LID systems—such as bias across dialects or speaker identities—remains critical for inclusive AI deployment in multilingual societies.

In summary, Wav2Vec2.0 represents a transformative advancement in spoken LID for low-resource and multilingual environments. Its combination of self-supervised representation learning, Transformer-based context modeling, and end-to-end training makes it a strong candidate for scalable deployment in

linguistically diverse nations like India. The insights from this study lay the groundwork for further innovations in multilingual speech understanding systems.

Declarations

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

P. G.: Conceptualization, Methodology, Software, Data curation, Writing—original draft. S. B.: Investigation, Supervision, Validation, Writing—review & editing.

Funding

No funding was received for the study.

Data Availability

The datasets used during the current study available from the corresponding author on reasonable request.

REFERENCES

1. Alashban, Adal A, Mustafa A Qamhan, Ali H Meftah, and Yousef A Alotaibi. 2022. "Spoken Language Identification System Using Convolutional Recurrent Neural Network." *Applied Sciences* 12 (18). MDPI: 9181.
2. Al-Kaltakchi, Musab TS, Wai L Woo, Satnam S Dlay, and Jonathon A Chambers. 2017. "Comparison of I-Vector and Gmm-Ubm Approaches to Speaker Identification with Timit and Nist 2008 Databases in Challenging Environments." In *2017 25th European Signal Processing Conference (Eusipco)*, 533–37. IEEE.
3. Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations." *Advances in Neural Information Processing Systems* 33: 12449–60.
4. Barai, Bidhan, Tapas Chakraborty, Nibaran Das, Subhadip Basu, and Mita Nasipuri. 2022. "Closed-Set Speaker Identification Using Vq and Gmm Based Models." *International Journal of Speech Technology* 25 (1). Springer: 173–96.
5. Basu, Joyanta, Soma Khan, Rajib Roy, Tapan Kumar Basu, and Swanirbhar Majumder. 2021. "Multilingual Speech Corpus in Low-Resource Eastern and Northeastern Indian Languages for Speaker and Language Identification." *Circuits, Systems, and Signal Processing* 40 (10). Springer: 4986–5013.
6. Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. "Learning Long-Term Dependencies with Gradient Descent Is Difficult." *IEEE Transactions on Neural Networks* 5 (2). IEEE: 157–66.
7. Boito, Marcely Zanon, Vivek Iyer, Nikolaos Lagos, Laurent Besacier, and Ioan Calapodescu. 2024. "Mhubert-147: A Compact Multilingual Hubert Model." *arXiv Preprint arXiv:2406.06371*.
8. Chang, Xuankai, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, et al. 2021. "An Exploration of Self-Supervised Pretrained Representations for End-to-End Speech Recognition." In *2021 Ieee Automatic Speech Recognition and Understanding Workshop (Asru)*, 228–35. IEEE.
9. Dabbabi, Karim, and Abdelkarim Mars. 2024. "Self-Supervised Learning for Speech Emotion Recognition Task Using Audio-Visual Features and Distil Hubert Model on Baved and Ravdess Databases." *Journal of Systems Science and Systems Engineering* 33 (5). Springer: 576–606.
10. Das, Himanish Shekhar, and Pinki Roy. 2021. "A Cnn-Bilstm Based Hybrid Model for Indian Language Identification." *Applied Acoustics* 182. Elsevier: 108274.
11. Dey, Spandan, Md Sahidullah, and Goutam Saha. 2022. "An Overview of Indian Spoken Language Recognition from Machine Learning Perspective." *ACM Transactions on Asian and Low-Resource Language Information Processing* 21 (6). ACM New York, NY: 1–45.
12. Diwan, Anuj, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, et al. 2021. "Multilingual and Code-Switching Asr Challenges for Low Resource Indian Languages." *arXiv Preprint arXiv:2104.00235*.
13. Dong, Linhao, Shuang Xu, and Bo Xu. 2018. "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition." In *2018 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp)*, 5884–8. IEEE.
14. Garai, Soumen, and Suman Samui. 2024. "Optimizing Performance of Spoken Language Identification Systems for Indian Languages Using Ensemble Deep Learning Models." In *2024 Ieee Calcutta Conference (Calcon)*, 1–5. IEEE.
15. Ghanimi, Hayder MA, Sudhakar Sengan, Vijaya Bhaskar Sadu, Parvinder Kaur, Manju Kaushik, Roobaea Alroobaea, Abdullah M Baqasah, Majed Alsafyani, and Pankaj Dadheech. 2024. "An Open-Source Mp+ Cnn+ Bilstm Model-Based Hybrid Model for Recognizing Sign Language on Smartphones." *International Journal of System Assurance Engineering and Management* 15 (8). Springer: 3794–3806.
16. Graves, Alex, and Jürgen Schmidhuber. 2005. "Framewise Phoneme Classification with Bidirectional Lstm and Other Neural Network Architectures." *Neural Networks* 18 (5-6). Elsevier: 602–10.
17. Hidayat, Kiran, Shakil Ahmed, Anam Akbar, and Tehmina Khan. 2024. "Comparative Examination of Spoken Language Recognition Through Deep Learning Algorithms–a Review." *Pakistan Journal of Engineering and Technology* 7 (2): 97–103.
18. Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8). MIT press: 1735–80.
19. Hu, Hengbo, Tong Niu, and Zhenhua He. 2025. "A Speech Recognition Method with Enhanced Transformer Decoder." *EURASIP Journal on Audio, Speech, and Music Processing* 2025 (1). Springer: 6.

20. Hussain, Sarmad. 2012. "Acoustic Feature Based Language Identification Using Single Word Utterances with Fixed Vocabulary." PhD thesis, University of Engineering; Technology Lahore Pakistan Lahore, Pakistan.
21. Jafarzadeh, Pourya, Amir Mohammad Rostami, and Padideh Choobdar. 2024. "Speaker Emotion Recognition: Leveraging Self-Supervised Models for Feature Extraction Using Wav2vec2 and Hubert." *arXiv Preprint arXiv:2411.02964*.
22. Ji, Hang, Tanvina Patel, and Odette Scharenborg. 2022. "Predicting Within and Across Language Phoneme Recognition Performance of Self-Supervised Learning Speech Pre-Trained Models." *arXiv Preprint arXiv:2206.12489*.
23. Jiang, Peiyuan, Weijun Pan, Jian Zhang, Teng Wang, and Junxiang Huang. 2023. "A Robust Conformer-Based Speech Recognition Model for Mandarin Air Traffic Control." *Computers, Materials & Continua* 77 (1).
24. Kingma, Diederik P, and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv Preprint arXiv:1412.6980*.
25. Kowsher, Md, Anik Tahabilder, Md Zahidul Islam Sanjid, Nusrat Jahan Prottasha, Md Shihab Uddin, Md Arman Hossain, and Md Abdul Kader Jilani. 2021. "LSTM-Ann & Bilstm-Ann: Hybrid Deep Learning Models for Enhanced Classification Accuracy." *Procedia Computer Science* 193. Elsevier: 131–40.
26. Krishna, DN. 2021. "Multilingual Speech Recognition for Low-Resource Indian Languages Using Multi-Task Conformer." *CoRR*.
27. Latif, Siddique, Aun Zaidi, Heriberto Cuayahuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir. 2023. "Transformers in Speech Processing: A Survey." *arXiv Preprint arXiv:2303.11607*.
28. Lee, Haeyoung, Sunhee Kim, and Minhwa Chung. 2024. "Analysis of Various Self-Supervised Learning Models for Automatic Pronunciation Assessment." In *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (Apsipa Asc)*, 1–6. IEEE.
29. Li, Haizhou, Bin Ma, and Kong Aik Lee. 2013. "Spoken Language Recognition: From Fundamentals to Practice." *Proceedings of the IEEE* 101 (5). IEEE: 1136–59.
30. Li, Shengqiang, Menglong Xu, and Xiao-Lei Zhang. 2021. "Efficient Conformer-Based Speech Recognition with Linear Attention." In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (Apsipa Asc)*, 448–53. IEEE.
31. Loshchilov, Ilya, and Frank Hutter. 2017. "Decoupled Weight Decay Regularization." *arXiv Preprint arXiv:1711.05101*.
32. Mallikarjun, Bhanavari. 2001. "Language According to Census of India 2001." *Language in India* 1 (2).
33. Mishra, Swami, Nehal Bhatnagar, Prakasam P, and Sureshkumar T. R. 2024. "Speech Emotion Recognition and Classification Using Hybrid Deep Cnn and Bilstm Model." *Multimedia Tools and Applications* 83 (13). Springer: 37603–20.
34. Montavon, Gregoire. 2009. "Deep Learning for Spoken Language Identification." In *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 1–4. Citeseer.
35. Morris, Andrew Cameron, Viktoria Maier, and Phil D Green. 2004. "From Wer and Ril to Mer and Wil: Improved Evaluation Measures for Connected Speech Recognition." In *Interspeech*, 2765–8.
36. Nayana, PK, Dominic Mathew, and Abraham Thomas. 2017. "Comparison of Text Independent Speaker Identification Systems Using Gmm and I-Vector Methods." *Procedia Computer Science* 115. Elsevier: 47–54.
37. Pakray, Partha, Alexander Gelbukh, and Sivaji Bandyopadhyay. 2025. "Natural Language Processing Applications for Low-Resource Languages." *Natural Language Processing* 31 (2). Cambridge University Press: 183–97.
38. Ploujnikov, Artem. 2024. "Towards a Unified Model for Speech and Language Processing."
39. Ranasinghe, Tharindu, and Marcos Zampieri. 2021a. "An Evaluation of Multilingual Offensive Language Identification Methods for the Languages of India." *Information* 12 (8). MDPI: 306.
40. ——. 2021b. "Multilingual Offensive Language Identification for Low-Resource Languages." *Transactions on Asian and Low-Resource Language Information Processing* 21 (1). ACM New York, NY: 1–13.
41. Sanabria, Ramon, Hao Tang, and Sharon Goldwater. 2023. "Analyzing Acoustic Word Embeddings from Pre-Trained Self-Supervised Speech Models." In *ICASSP 2023-2023 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp)*, 1–5. IEEE.
42. Shah, Sanket, Satarupa Guha, Simran Khanuja, and Sunayana Sitaram. 2020. "Cross-Lingual and Multilingual Spoken Term Detection for Low-Resource Indian Languages." *arXiv Preprint arXiv:2011.06226*.
43. Singh, Gundeep, Sahil Sharma, Vijay Kumar, Manjit Kaur, Mohammed Baz, and Mehedi Masud. 2021. "Spoken Language Identification Using Deep Learning." *Computational Intelligence and Neuroscience* 2021 (1). Wiley Online Library: 5123671.
44. Singh, Shruti, Muskaan Singh, and Virender Kadyan. 2024. "Speech Recognition Transformers: Topological-Lingualism Perspective." *arXiv Preprint arXiv:2408.14991*.
45. Tahir, Jawaria, and others. 2023. "Automatic Speech Recognition for Low Resource Language (Pashto) Using Wav2vec Model." In. MCS.
46. Tiwari, Varun, Mohammad Farukh Hashmi, Avinash Keskar, and NC Shivaprakash. 2019. "Speaker Identification Using Multi-Modal I-Vector Approach for Varying Length Speech in Voice Interactive Systems." *Cognitive Systems Research* 57. Elsevier: 66–77.
47. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *Advances in Neural Information Processing Systems* 30.
48. Vielzeuf, Valentin. 2024. "Investigating the'Autoencoder Behavior'in Speech Self-Supervised Models: A Focus on Hubert's Pretraining." *arXiv Preprint arXiv:2405.08402*.
49. Yadav, Hemant, and Sunayana Sitaram. 2022. "A Survey of Multilingual Models for Automatic Speech Recognition." *arXiv Preprint arXiv:2202.12576*.
50. Yagle, Andrew E. 2001. "A Fast Algorithm for Toeplitz-Block-Toeplitz Linear Systems." In *2001 Ieee International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, 3:1929–32. IEEE.

51. Yi, Cheng, Shiyu Zhou, and Bo Xu. 2021. "Efficiently Fusing Pretrained Acoustic and Linguistic Encoders for Low-Resource Speech Recognition." *IEEE Signal Processing Letters* 28. IEEE: 788–92.
52. Zaiem, Mohamed Salah. 2024. "Informed Speech Self-Supervised Representation Learning." PhD thesis, Institut Polytechnique de Paris.
53. Zeinali, Hossein, Hossein Sameti, and Lukáš Burget. 2017. "HMM-Based Phrase-Independent I-Vector Extractor for Text-Dependent Speaker Verification." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25 (7). IEEE: 1421–35.
54. Zewoudie, Abraham Woubie. 2017. "Discriminative Features for Gmm and I-Vector Based Speaker Diarization." Universitat Politècnica de Catalunya.
55. Zhang, Qian. 2017. "Advancements in Acoustic Based Language Identification/Recognition."
56. Zhao, Jing, and Wei-Qiang Zhang. 2022. "Improving Automatic Speech Recognition Performance for Low-Resource Languages with Self-Supervised Models." *IEEE Journal of Selected Topics in Signal Processing* 16 (6). IEEE: 1227–41.
57. Zissman, Marc A. 1996. "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech." *IEEE Transactions on Speech and Audio Processing* 4 (1). IEEE: 31.