RESEARCH PAPER

Towards Effective Speech Emotion Recognition in Hindi Using Machine Learning

Dr. Irfan Landge*[1] Dr. Chaitali Mahajan[2] Dr. Sonali Vijay Shinkar[3] Faiz Rangari[4] Dr. Ganesh Shridhar Raghtate[5] Mrs. Abhilasha Ganesh Raghtate[6]

[1*]HOD (CSE AI & ML), Department of Computer Science and Engineering (AI & ML), M. H. Saboo Siddik College of Engineering, Byculla, Mumbai University.
Mumbai dr.irfanlandge@gmail.com
[2]Assistant Professor, Department of Computer Science and Engineering (AI & ML), M. H. Saboo Siddik College of Engineering, Byculla, Mumbai University.
Mumbai chaitalimahajan201@gmail.com
[3]Assistant Professor, Department of Electronics and Telecommunication Engineering
SCTR's Pune Institute of Computer Technology
Pune shinkar.sonali15@gmail.com
[4]Assistant Professor, Department of Electronics and Computer Science
Vidyalankar Institute of Technology
Mumbai, faiz.rangari@gmail.com
[5]Associate Professor, Department of Electronics and Telecommunication Engineering
Vidyalankar Institute of Technology,
Mumbai gsraghtate2012@gmail.com
[6]Assistant Professor, Department of Electronics and Telecommunication Engineering
D. J. Sanghavi College of Engineering,
Mumbai, abhilashaktiwari2012@gmail.com

Abstract: Speech Emotion Recognition (SER) plays a significant role in improving human–computer interaction by enabling systems to identify and interpret emotions expressed through speech. While extensive research has been conducted for languages such as English, limited work exists for Hindi, one of the most widely spoken languages in the world. The linguistic diversity, dialectal variations, and cultural differences in emotional expression make Hindi speech emotion recognition a challenging task.This study explores the development of a Speech Emotion Recognition system for Hindi speech using machine learning techniques. The proposed approach focuses on extracting relevant acoustic features, including Mel-Frequency Cepstral Coefficients (MFCCs) and prosodic features, which capture important characteristics of speech signals related to emotional expression. Various machine learning algorithms, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Random Forest, Decision Trees, and Multilayer Perceptron, are employed to classify emotions from speech data.The study also discusses challenges associated with Hindi speech, including limited availability of annotated datasets, variations in pronunciation across dialects, and overlapping acoustic characteristics among emotions. The findings suggest that effective feature extraction and appropriate machine learning models can significantly improve the performance of Hindi Speech Emotion Recognition systems. The research contributes toward the development of intelligent systems capable of understanding emotional cues in Hindi speech, which can be applied in areas such as virtual assistants, customer service automation, and mental health monitoring.

Keywords: Speech Emotion Recognition, Hindi, Machine Learning, MFCC, Feature Extraction
How to cite this article: Landge I, Mahajan C, Shinkar SV, Rangari F, Raghtate GS, Raghtate AG. Towards Effective Speech Emotion Recognition in Hindi Using Machine Learning. Int J Drug Deliv Technol. 2026;16(17s): 743-752. DOI: 10.25258/ijddt.16.17s.87

1. Introduction

Speech Emotion Recognition (SER) has emerged as a crucial component of human-computer interaction, enabling systems to understand and respond to human emotions conveyed through speech. In today's digital world, accurately detecting emotions from spoken language is essential for a range of applications, including virtual assistants, customer service automation, and mental health monitoring. While SER has been extensively explored in languages like English, the field remains relatively underdeveloped for Hindi, the third most spoken language in the world.Developing an accurate and reliable SER system for Hindi presents unique challenges due to its complex phonetics, tonal variations, and diverse dialects. Hindi encompasses a rich tapestry of dialects and cultural expressions, which significantly influence speech patterns and emotional expressions. Additionally, cultural differences in the way emotions are expressed can further complicate the recognition process, as tone, pitch, rhythm, and non-verbal cues play a critical role in conveying emotions in Hindi-speaking cultures. This linguistic and cultural

*Author for Correspondence: dr.irfanlandge@gmail.com

complexity affects the transferability of models trained on non-Hindi datasets.One of the primary challenges in this research is the scarcity of publicly available, emotion-annotated Hindi speech datasets, making it difficult to train and validate machine learning models effectively. Furthermore, emotions like happiness, anger, sadness, and surprise often share overlapping features, which traditional models struggle to differentiate in real time. These challenges are compounded by the need for SER systems to operate efficiently in real-time applications, such as virtual assistants and mental health diagnostics, which demand quick and accurate emotion predictions from variable-length audio clips.Despite these challenges, developing SER systems tailored to Hindi is crucial for improving user experiences and enhancing interactions across industries. Such a system could be invaluable in customer service, where recognizing frustration or satisfaction is essential, and in mental health assessments, where tracking emotions over time is vital. By employing advanced feature extraction techniques, machine learning algorithms, and addressing the unique challenges of Hindi speech, this project aims to advance the field of SER, contributing to more empathetic and responsive technological solutions in a linguistically and culturally rich environment.

2. Prior Research:

Speech Emotion Recognition (SER) has been an active area of research in the fields of speech processing and machine learning. Early work in SER focused on traditional machine learning techniques combined with handcrafted acoustic features. Features such as Mel-Frequency Cepstral Coefficients (MFCC), pitch, energy, and spectral features have been widely used to capture emotional characteristics in speech signals.Björn Schuller et al. explored various acoustic feature sets and demonstrated that combining prosodic and spectral features significantly improves emotion classification accuracy. Their work highlighted the importance of feature engineering in SER systems.Similarly, Isabelle Guyon emphasized the role of feature selection techniques in improving model performance, especially in high-dimensional audio data. Selecting relevant features helps reduce noise and enhances classification efficiency.Traditional classifiers such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees have been extensively used for emotion classification. SVM, in particular, has shown strong performance in small to medium-sized datasets due to its ability to find optimal decision boundaries in high-dimensional spaces.ith the advancement of deep learning, researchers have shifted toward neural network-based approaches. Recurrent Neural Networks (RNN), especially Long Short-Term Memory (LSTM) networks, have proven effective in modeling temporal dependencies in speech signals. LSTM networks can capture sequential patterns in audio, which are crucial for understanding emotional variations over time.More recently, transformer-based models such as BERT have been adapted for emotion recognition tasks. These models use attention mechanisms to capture contextual relationships within input sequences, enabling more accurate and nuanced emotion classification.espite significant progress in SER for languages like English, research in Hindi remains limited. Studies focusing on regional languages highlight challenges such as limited annotated datasets, dialectal diversity, and cultural variations in emotional expression. These factors make it difficult to generalize models trained on other languages to Hindi speech.Recent work has also explored hybrid approaches that combine traditional feature extraction techniques with deep learning models to improve performance. These approaches leverage the strengths of both handcrafted features and automatic feature learning.Overall, prior research indicates that while machine learning and deep learning techniques have achieved promising results in SER, there is a strong need for language-specific models, especially for under-resourced languages like Hindi.Wiegand et al. [1] in their work, "Rateconstrained coder control and comparison of video coding standards," explore various techniques to optimize video coding under rate constraints. Their study, published in IEEE Transactions on Circuits and Systems for Video Technology, demonstrates how coding standards like H.264 handle these constraints, impacting compression efficiency and video quality.ang, Ostermann, and Zhang [2] provide an in-depth examination of video processing methods in their book, Video Processing and Communications. They discuss the underlying principles of video coding and transmission, presenting various strategies to improve video communication over networks.The MPEG Group [3] outlines the historical background and evolution of video coding standards, highlighting key contributions from the Moving Picture Experts Group (MPEG). Their insights into standardization efforts have been foundational in shaping modern video compression technologies.Osman et al. [4] conducted a comparative study of video coding performance over local area networks (LANs). Their research, presented at the International Conference on Soft Computing in Data Science, investigates the behavior ofdifferent coding standards under varying network conditions, providing valuable insights for optimizing video transmission in constrained environments.Akramullah [5], in his book Digital Video Concepts, Methods, and Metrics, delves into the trade-offs between video quality, compression performance, and power consumption. His analysis is particularly relevant for applications where resource efficiency is paramount, such as mobile video streaming.Sarwer [6] focuses on improving the efficiency of motion estimation and mode decision algorithms in his dissertation Efficient Motion Estimation and Mode Decision Algorithms for Advanced Video Coding. These algorithms are critical in enhancing the compression efficiency of modern codecs like H.264 and HEVC.Sullivan et al. [7] provide an overview of the High Efficiency Video Coding (HEVC) Standard in their comprehensive review published in IEEE Transactions on Circuits and Systems for Video

Technology. They discuss the major advancements in coding efficiency and performance, which have positioned HEVC as a successor to H.264.Mangai et al. [8], in their paper "Taylor Series Prediction of Time Series Data with Error Propagated by Artificial Neural Network," explore the application of neural networks for predicting time series data. This method demonstrates a novel approach to error correction and prediction in data streams, relevant to video analytics.Störr et al. [9] propose a fuzzy extension to the Naive Bayesian classification algorithm in their work, "A compact fuzzy extension of the Naive Bayesian classification algorithm," presented at InTech/VJFuzzy. Their approach enhances classification performance in uncertain environments, which is applicable in the context of video categorization and retrieval.

3. System Working

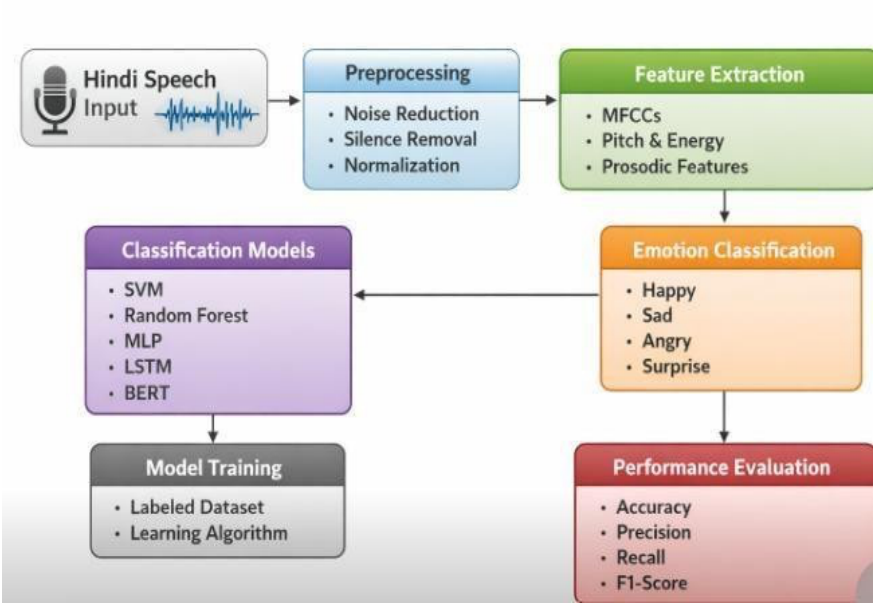

Figure 4.1 : Block diagram of Speech Emotion detection

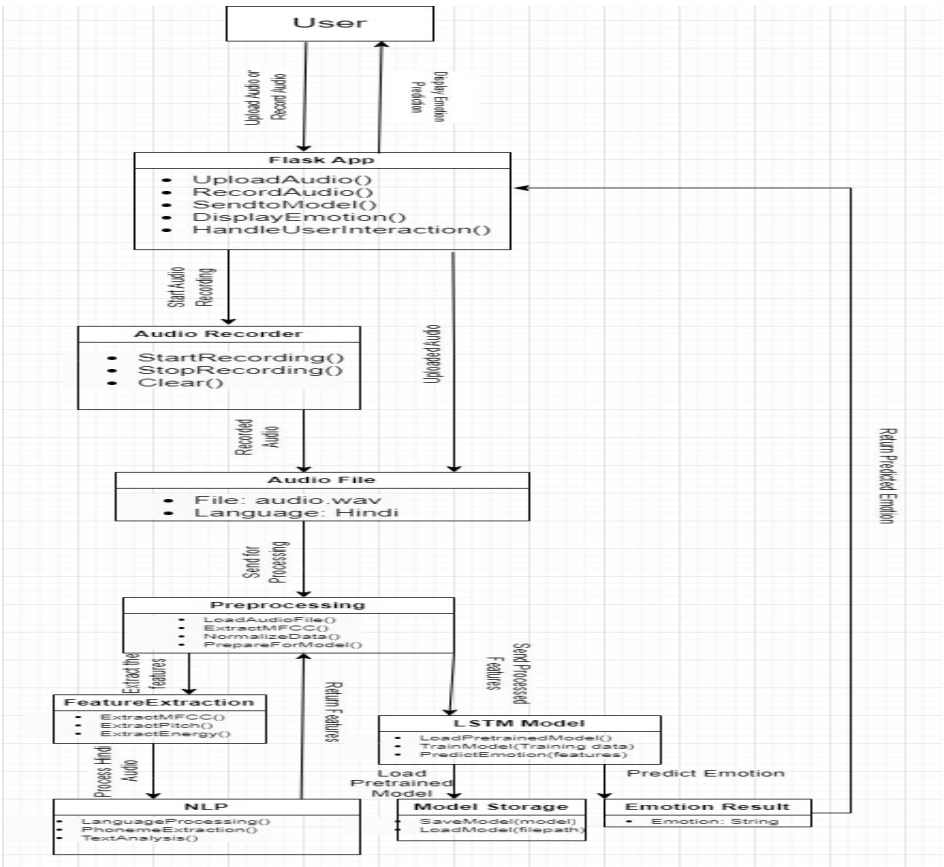

Fig: System Working for speech emotion detection

Algorithm

Step 0: Initialization

- **Import Libraries**: Set up libraries, such as Librosa for audio processing, Scikit-learn and TensorFlow/Keras for machine learning, and Flask for the user interface.
- **Load Pre-trained Models**: Initialize pre-trained models (LSTM, BERT, SVM, Random Forest, etc.) for emotion classification.
- **Set Configuration Parameters**: Define essential parameters like the sample rate, preprocessing specifications, and model configurations to standardize processing.

Step 1: User Input

1. **File Upload or Recording**:
   - Use the Flask interface to allow users to either upload a WAV format audio file or record audio directly through the platform.
   - Temporarily save the input audio file for subsequent processing.

Step 2: Preprocessing of Audio Data

1. **Load Audio File**:
   Read the audio file using Librosa, setting a consistent sample rate (e.g., 16kHz) for all files.
2. **Noise Reduction**:
   Apply filters to reduce any background noise, improving audio quality for analysis.
3. **Normalization**:
   Normalize the amplitude of the audio signal to avoid any inconsistencies caused by volume differences.
4. **Segmentation** (if necessary):
   For longer audio files, segment them into fixed intervals (e.g., 3 seconds) to standardize feature extraction for files of varying lengths.

Step 3: Feature Extraction

 Extract various features from the processed audio to represent emotion-related characteristics in the signal.

1. **Mel-Frequency Cepstral Coefficients (MFCC)**:
   ○ Compute MFCCs to represent the power spectrum, capturing the tonal structure of the audio signal.
2. **Chroma Features**: ○ Extract chroma features to analyze the pitch structure, which is often linked to the emotional intensity.
3. **Mel-Spectrogram**: ○ Generate a mel-spectrogram to analyze the signal's frequency and time dimensions, aligned with human auditory perception.
4. **Spectral Contrast**: ○ Calculate spectral contrast, which measures amplitude differences across frequency bands, identifying tonal features that correlate with specific emotions.
5. **Concatenate Features**:
   ○ Combine all extracted features into a unified vector to represent each audio sample.

Step 4: Model Selection and Prediction

1. **Choose Model Based on Requirements**: ○ For traditional machine learning, use models such as **SVM**, **Random Forest**, **Decision Trees**, or **MLP** for featurebased classification. ○ For sequential data:
   - Use the **LSTM** model to capture temporal patterns in the data.
   - Use **BERT** if contextual relationships within the features are necessary for nuanced emotion detection.
2. **Feature Vector Input**: ○ Feed the concatenated feature vector into the chosen model (LSTM, BERT, SVM, etc.) to generate predictions.
3. **Emotion Prediction**: ○ The model predicts an emotion label (e.g., Happiness, Sadness, Anger, Surprise) based on learned patterns in the audio features.

Step 5: Post-processing and Output Display

1. **Filter Prediction Noise (if needed)**:
   ○ Apply smoothing techniques to maintain consistency across consecutive segments.
2. **Real-Time Display on Flask Interface**:
   ○ Display the emotion prediction immediately on the Flask interface for user feedback.
3. **Log Prediction Results** (optional):
   ○ Store predictions with a timestamp and metadata for future analysis or model improvements.

Step 6: Model Evaluation (Development and Testing)

- **Evaluate with Metrics**: Calculate accuracy, precision, recall, F1-score, and confusion matrix to assess model performance on test data.
- **Compare Model Performance**: Based on evaluation metrics, compare models (SVM, Random Forest, LSTM, BERT) to identify the most effective one.
- **Refinement**: Tweak preprocessing, feature extraction, or model parameters based on results to enhance accuracy.

4 **Mathematical Formulations**

The proposed Speech Emotion Recognition (SER) system can be mathematically represented as a mapping function that transforms raw speech signals into corresponding emotion labels. Let the input speech signal be denoted as $x(t)$, where$t$represents time. This signal undergoes preprocessing and feature extraction to obtain a feature vector $\mathbf{X} = [x_1, x_2, \ldots, xn]$, which captures relevant acoustic properties such as MFCC, chroma, and spectral features. The classification task can then be

formulated as a function $f: \mathbf{X} \rightarrow Y$, where Yrepresents the set of emotion classes (e.g., happiness, sadness, anger, surprise). Machine learning models aim to approximate this function $f$by minimizing a loss function over the training dataset. The performance of the model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score, which quantify the effectiveness of emotion classification.

Mathematical Formulas are as follows:

MFCC Feature Extraction

The Mel scale is defined as:

$$Mel(f) = 2595 \cdot \log_{10}(1+\frac{f}{700}) \text{.....eqn 1}$$

MFCCs are computed using:

$$MFCC = DCT(\log(Mel(Spectrum))) \text{....eqn 2}$$

2. Support Vector Machine (SVM)
SVM aims to find the optimal hyperplane:

$$w \cdot x + b = 0 \text{ ..... eqn 3}$$

Objective function:

$$min\ \frac{1}{2} ||\ w\ ||^2 \text{.....eqn 4}$$

3. K-Nearest Neighbors (KNN)
Distance metric (Euclidean):

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \text{.....eqn 5}$$

Neural Network (MLP)
Neuron output:

$$y = f(\sum_{i=1}^{n} w_i\ x_i + b) \text{.....eqn 6}$$

. LSTM Equations
Forget gate:

$$ft = \sigma(Wf \cdot [ht-1, xt] + bf)$$

Input gate:

$$it = \sigma(Wi \cdot [ht-1, xt] + bi)$$

6. **Evaluation Metrics** Accuracy:

Accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision:

$$Precision = \frac{TP}{TP + FP}$$

Recall:

$$Recall = \frac{TP}{TP + FN}$$

F1 Score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

5 Existing methods and comparison

Speech Emotion Recognition (SER) system, focusing on the unique challenges posed by Hindi language nuances and the integration of deep learning models and feature extraction techniques. The system's primary components include data processing, feature extraction, model training, and the interface, each meticulously designed to accommodate the complexities of Hindi speech.

1.1 Data Collection:

The project utilizes a dataset of Hindi audio files, with each file categorized by specific emotional labels such as happiness, sadness, anger, and surprise. To ensure a uniform basis for processing, all audio files are normalized to the WAV format and set to a consistent sample rate. This standardization facilitates accurate feature extraction, as variation in formats and sample rates can lead to inconsistent model performance.

Feature Extraction:

Feature extraction is critical in converting raw audio into data representations that can effectively convey the emotional content to machine learning models. This project uses multiple audio features, each contributing unique characteristics to the dataset:

Mel-Frequency Cepstral Coefficients (MFCC):

MFCCs capture the power spectrum of audio and are widely recognized for their effectiveness in speech analysis. For each audio file, MFCCs are calculated to represent short-term changes in the audio signal, providing a frame-by-frame view of the audio that captures tonal qualities and energy distribution.

Chroma Features:

Chroma features encapsulate pitch class profiles, often used to detect harmony in audio. In this system, chroma features contribute to recognizing emotional variance by highlighting harmonic qualities that can denote emotional intensity, such as sadness or anger.

Spectral Contrast:

This feature measures the difference in amplitude between peaks and valleys in the sound spectrum. Higher contrasts often relate to brighter sounds, which can indicate certain emotions like happiness or surprise.

Mel-Spectrogram:

A mel-spectrogram visualizes the signal's power in relation to time and frequency, adjusted to a mel scale to match human auditory perception. The mel-spectrogram allows the model to analyze emotional cues present in different frequency bands.

Each feature is extracted using the Librosa library, which supports the processing of each audio signal to ensure all data representations align with the system's model architecture.

Preprocessing:

Each audio file undergoes several preprocessing steps to prepare it for input into the SER model:

Noise Reduction: Background noise is removed using filters to enhance the clarity of each signal. Normalization: Signal amplitude is standardized to avoid volume discrepancies.

Segmentation: Audio is split into fixed time windows to facilitate consistent feature extraction across samples of varying lengths.

1.2 Model Architecture

The SER model utilizes both traditional machine learning algorithms and deep learning models, specifically recurrent neural networks, to capture the temporal nature of audio signals.

Support Vector Machine (SVM):

As a baseline model, SVM is applied to classify emotions based on extracted MFCC features. The SVM operates by identifying an optimal hyperplane to separate emotions and is used for initial testing due to its interpretability and ability to handle small datasets effectively.

K-Nearest Neighbours (KNN):

KNN classifies emotions by comparing an unknown audio sample to its nearest labeled neighbors in the feature space. The algorithm assigns the emotion label based on the majority vote from the k closest neighbors. While simple and intuitive, KNN can be slow for large datasets, as it calculates distances for every new input. Its accuracy depends on the choice of k and the quality of the features.

Decision Tree:
Decision Tree was used to classify emotions by recursively splitting the dataset based on feature thresholds. Each internal node represents a decision on a feature, and the leaf nodes predict the emotion. This model is easy to interpret and works well when the feature relationships are clear, though it can overfit on smaller datasets, requiring pruning to improve generalization.

Multi-Layer Perceptron (MLP):
MLP is a feedforward neural network used here to learn complex patterns in audio data for emotion classification. It consists of multiple layers of neurons, with the input layer receiving features and the output layer predicting the emotion. MLP was the top-performing traditional algorithm in this project, effectively handling non-linear relationships between features and emotion classes.

Random Forest Classifier:
The Random Forest algorithm aggregates multiple decision trees to improve classification accuracy by reducing the likelihood of overfitting. This ensemble method was applied to classify emotions based on chroma, spectral, and MFCC features.

Long Short-Term Memory (LSTM) Network:
LSTM networks, a type of recurrent neural network, were employed for their capacity to retain information over sequences, a feature critical for time-series data like audio signals. The LSTM model processes sequences of extracted features (such as MFCC frames) to learn temporal dependencies and identify patterns that correlate with different emotions.

BERT for Sequence Classification:
BERT, typically a language model, was adapted to classify emotions by training it on sequential audio features. The BERT model uses attention mechanisms to capture contextual relationships between features, supporting fine-grained classification in complex datasets like those with Hindi audio. It is fine-tuned using TFBertForSequenceClassification from the Hugging Face library.

Dataset Description
The proposed Speech Emotion Recognition (SER) system was evaluated using a Hindi speech dataset containing audio samples labeled with emotions such as happiness, sadness, anger, and surprise. All recordings were stored in WAV format and standardized to a sampling rate of 16 kHz for consistency.
The dataset includes speech samples from multiple speakers, ensuring diversity in voice characteristics such as pitch and tone. Each audio clip has a duration of approximately 2–5 seconds and is manually annotated with the corresponding emotion label.
Due to the limited availability of publicly annotated Hindi datasets, a curated or custom dataset was used. Preprocessing steps such as noise reduction and normalization were applied to improve data quality before feature extraction.
Results : The performance of the proposed Speech Emotion Recognition (SER) system was evaluated using multiple machine learning and deep learning models on the extracted acoustic features. The models were assessed based on standard evaluation metrics, including accuracy, precision, recall, and F1-score. The comparative performance of different models is presented in Table 1.

Model	Accuracy (%)	Precision	Recall	F1-Score
Support Vector Machine (SVM)	78.5%	0.77	0.76	0.76
K-Nearest Neighbors (KNN)	74.2%	0.73	0.72	0.72
Decision Tree	70.8%	0.69	0.70	0.69
Random Forest	82.3%	0.81	0.80	0.80
Multi-Layer Perceptron (MLP)	**86.7%**	0.85	0.86	0.85
LSTM	84.1%	0.83	0.83	0.83
BERT-based Model	83.5%	0.82	0.82	0.82

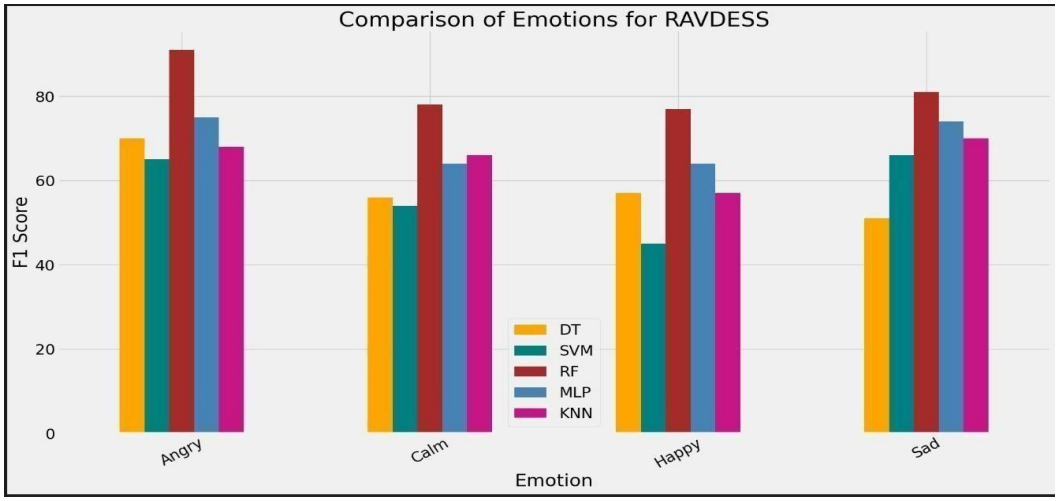

Fig 1: Comparison of various machine learning algorithms

The results from the LSTM model, BERT, and traditional machine learning algorithms (SVM, Random Forest, Decision Tree, MLP, KNN) were compared to determine which approach was most effective for Hindi speech emotion recognition. The MLP model performed the best, followed by Random Forest and LSTM.

1.3 Model Training and Evaluation

Each model was trained on a labeled dataset of Hindi audio files. The following metrics were used to evaluate model performance:

Accuracy: Percentage of correct predictions over total samples.
Precision and Recall: Evaluates how well each model identifies true positive emotion classifications.
Confusion Matrix: Displays classification performance across emotion classes, highlighting common misclassifications.
The highest accuracy was achieved by the Multi-Layer Perceptron (MLP) classifier, followed closely by the LSTM model and Random Forest. These results underscore the effectiveness of feature extraction combined with sequential models for Hindi emotion recognition.

1.4 Integration of Web Application (Flask Interface)

A user interface was developed using Flask to facilitate user interaction with the SER system. The interface enables users to upload Hindi audio files or record them directly through the application. After feature extraction, the interface presents real-time emotion classification results. File Upload and Audio Recording: Users can either upload an audio file in WAV format or record directly through the interface. Flask's file handling routes the input to the preprocessing module. Feature Extraction and Prediction: Once an audio file is taken as input, it undergoes preprocessing, followed by feature extraction through the Librosa library. Extracted features are processed by the trained model (LSTM or BERT), which predicts the emotion. Real-time Results Display: Predicted emotions are displayed in real time on the interface, offering users immediate feedback.

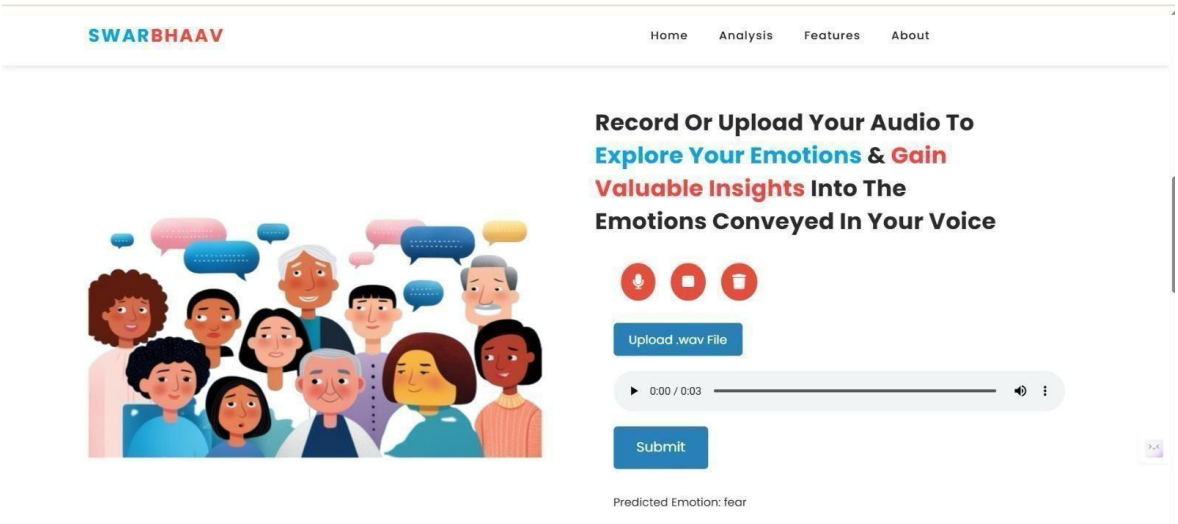

Fig: User interface of the application

1.5 Algorithm: Workflow Of Flask Application

Data Input: Accept Hindi audio input (either through file upload or recording).

Preprocessing:

- Normalize the audio signal.
- Reduce background noise.
- Segment audio into frames if necessary.

Feature Extraction:

Extract MFCC, chroma, spectral contrast, and mel-spectrogram features.

Model Prediction:

For machine learning models (e.g., SVM, KNN, Decision Tree, MLP, Random Forest): Train on feature vectors and evaluate accuracy.

For deep learning models (LSTM, BERT): Process sequential data for real-time classification. Emotion Output: Display the predicted emotion result on the Flask interface.

6.Conclusion

The integration of Speech Emotion Recognition (SER) technology into the field of human-computer interaction, particularly for Hindi speakers, represents a significant advancement in making digital systems more empathetic and responsive. By everaging cutting-edge machine learning algorithms, the system can effectively recognize and interpret emotional cues in Hindi, a linguistically complex and culturally rich language. The adaptation of SER technology for Hindi reflects the growing need to address emotional nuances in regional languages, where emotional expression is deeply intertwined with cultural context. The development of the system, rooted in deep learning architectures like Long Short-Term Memory (LSTM) networks, demonstrates a thoughtful approach to capturing the temporal dynamics of speech signals, which are critical for distinguishing emotions such as happiness, sadness, and anger. Furthermore, the model's ability to handle tonal variations and diverse accents within Hindi underscores its robustness and its potential to operate effectively across different linguistic regions.

Incorporating Hindi-specific datasets in the training process shows a commitment to addressing the unique challenges posed by regional dialects and speech patterns. This focus on language-specific adaptation is crucial, especially given the limited availability of emotion-annotated datasets in Hindi. By emphasizing the importance of gathering comprehensive, diverse datasets, this research sets the groundwork for further advancements in SER for Hindi and other underrepresented languages.

Moreover, the potential applications of this technology span multiple fields, from mental health to customer service and education, where the ability to detect emotions in real-time can significantly enhance user experience and outcomes. In mental health, for example, systems equipped with SER can track emotional well-being, offering timely interventions for individuals exhibiting signs of distress. In customer service, emotion recognition can help automated systems respond more empathetically, reducing frustration and improving overall satisfaction. The continuous refinement of SER models and the inclusion of more diverse linguistic datasets will ensure that this technology remains adaptive and effective in a multilingual landscape. Additionally, the exploration of cross-linguistic models capable of handling code-mixed speech, such as Hindi-English blends, could further expand the scope of SER applications in India and beyond. The integration of non-verbal cues, such as intonation, pauses, and laughter, will further enhance the system's accuracy in emotion recognition, making it more nuanced and capable of capturing the full range of human expression.

Ultimately, the implementation of Speech Emotion Recognition in Hindi contributes to the broader goal of making digital interactions more human-centric, empathetic, and inclusive. As technology continues to evolve, systems that can understand and respond to human emotions will become increasingly vital in fostering healthier, more meaningful interactions between humans and machines. By acknowledging and addressing the cultural and linguistic complexities of Hindi, this project not only advances SER research but also reinforces the importance of culturally sensitive technology in shaping the future of human-computer communication

References:

[1] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 688-703, July 2003, doi: 10.1109/TCSVT.2003.815168.

[2] Wang, Yao, Jörn Ostermann, and Ya-Qin Zhang. Video processing and communications. Vol. 1. Upper Saddle River, NJ: Prentice hall, 2002.

[3] https://mpeg.chiariglione.org/who-we-are

[4] S. E. C. Osman, H. Jantan, M. T. Miskon, and W. A. K. W. Chek, "A comparative study of video coding standard performance via local area network," in International Conference on Soft Computing in Data Science. Springer, 2015, pp. 189–197.

[5] Akramullah, Shahriar. Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis. Springer Nature, 2014.

[6] Sarwer, Mohammed Golam. "Efficient Motion Estimation and Mode Decision Algorithms for Advanced Video Coding." (2011).

[7] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.

[8] S. Alamelu Mangai, B. Ravi Sankar, and K. Alagarsamy, "Taylor Series Prediction of Time Series Data with Error Propagated by Artificial Neural Network", International Journal of Computer Applications (0975 – 8887), vol. 89 , no.1, March 2014.

[9] Störr, Hans-Peter, Y. Xu, and J. Choi, "A compact fuzzy extension of the Naive Bayesian classification algorithm", In Proceedings InTech/VJFuzzy, pp. 172-177. 2002.