Voice Classification System Using Gradient Boosting Method

Nihal S Pujari [1], Vibha [2], Mamata [3], Dr. Prakash Pattan [4]
Dept.Of Computer Science And Design

Poojya Doddappa Appa College of Engineering,
Kalabuargi, India

nihalpujari2@gmail.com [1], vibha.b.jahagirdar@gmail.com [2], msajjan62@gmail.com [3], prakashpattan@pdaengg.com [4]

Abstract-**This project introduces a speaker recognition system built upon advanced techniques in audio signal analysis and machine learning. It focuses on identifying individual speakers by analyzing their voice patterns from recorded .wav files. The system extracts a range of meaningful features from the audio, including Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid, pitch, and chroma characteristics. These features are then fed into a Gradient Boosting Classifier, which is trained using a labeled dataset of voice samples. The model demonstrates strong performance in accurately distinguishing between different speakers. Designed with adaptability in mind, the system holds promise for real-world use cases such as integration into voice-based authentication platforms and smart, voice-responsive applications. [6][2].**

Keywords: **Speaker Recognition, MFCCs, Gradient Boosting Classifier, spectral centroid, chroma features, Gradient Boosting.**

I. INTRODUCTION:

A voice recognition system is a technology designed to distinguish or confirm the identity of a person based on unique features of their voice—much like a vocal fingerprint. In this project, we have developed a **text-independent speaker recognition system**, meaning the individual can speak any content, and the system can still identify who is speaking. Unlike speech recognition, which focuses on transcribing spoken words, voice recognition is concerned with identifying the speaker behind those words. This distinction makes it especially useful in areas such as secure authentication, voice-controlled personal assistants, and criminal investigations.

The proposed system is built around **three key tasks** of voice recognition, namely speaker identification and speaker verification. It leverages audio signal processing techniques to extract distinctive features from voice recordings—such as MFCCs (Mel-Frequency Cepstral Coefficients), spectral centroid, pitch, and chroma patterns. These extracted features serve as inputs to a machine learning model known as the **Gradient Boosting Classifier (GBC)**, which is responsible for classifying speakers based on their vocal traits.

Gradient boosting is a powerful ensemble learning method that improves model performance by combining the outputs of several weaker models. It follows an iterative training process, where each new model aims to fix the errors made by the previous ones. Through gradient descent, the algorithm optimizes its loss function and progressively builds a more accurate prediction model. By assigning appropriate weights to each model based on its performance, gradient boosting ensures that the final output is a well-balanced and accurate classification result.

This voice recognition system is engineered to be both efficient and scalable, capable of delivering high accuracy in real-time environments. Its potential applications include secure access systems, forensic voice analysis, and intelligent customer service platforms that adapt to individual users based on voice identity[2][4].

Gradient boosting has four steps:

1. **Initialization** – starts with simple model.
2. **Iteration** – It iteratively trains new models to correct the errors of the previous model.
3. **Weighting** – Each model is assigned a weight based on its performance.
4. **Combination** – The final prediction is made by combining the predictions of all models.

Advantages of Gradient model are its high accuracy, handling complex data & its robustness (handling efficiently outliers & missing values).

II. METHODOLOGY:

a) Dataset Preparation: The dataset utilized for this voice recognition system comprises a collection of .wav audio files, each corresponding to a different individual. These files are grouped into dedicated folders for each speaker, with multiple voice recordings per folder. The recordings span a variety of conditions, capturing differences in factors such as speech pace, tone, and background noise. This variability enriches the dataset, allowing the system to become more resilient to the natural inconsistencies found in human speech patterns.

To ensure consistency and compatibility with audio processing tools, all recordings are standardized to a 16 kHz sampling rate. This is achieved using **Librosa**, a widely-recognized Python library designed for music and audio analysis. Each audio file is carefully labeled with the speaker's identity, a critical step for supervised learning where the system is trained to match audio features to specific individuals.

The dataset is structured in a hierarchical manner, simplifying the automation of the feature extraction and labelling process, which is key to preparing the data for efficient model training and testing [7].

It has basically 3 important stages,

i. A recording of the speaker's voice – Dataset preparation
ii. Feature extraction – Analyzing the audio for unique voice characteristics
iii. Pattern matching – Comparing extracted features to a database

b) Preprocessing:

The features extracted from the audio data are normalized using StandardScaler, ensuring that all feature values are on a consistent scale. This step helps the model converge more effectively during training. Additionally, the speaker labels, originally in string or categorical format, are converted into numerical indices using LabelEncoder. This transformation allows the classifier to interpret the speaker identities as distinct integer values, making it compatible with the classification process, where each unique speaker is assigned a separate integer label.

To assess the model's ability to generalize to new, unseen data, the preprocessed dataset is divided into training and testing sets in an 80-20 ratio. This means 80% of the data is used for training the Gradient Boosting Classifier, and the remaining 20% is reserved for testing the model's performance. This approach ensures the model is evaluated on data it hasn't encountered before, providing a robust indicator of its real-world effectiveness. In cases where the dataset might be imbalanced, stratified sampling can be applied to ensure that all speaker classes are fairly represented in both the training and testing subsets.

c) *Feature Extraction: For each audio file, the following features are extracted:*

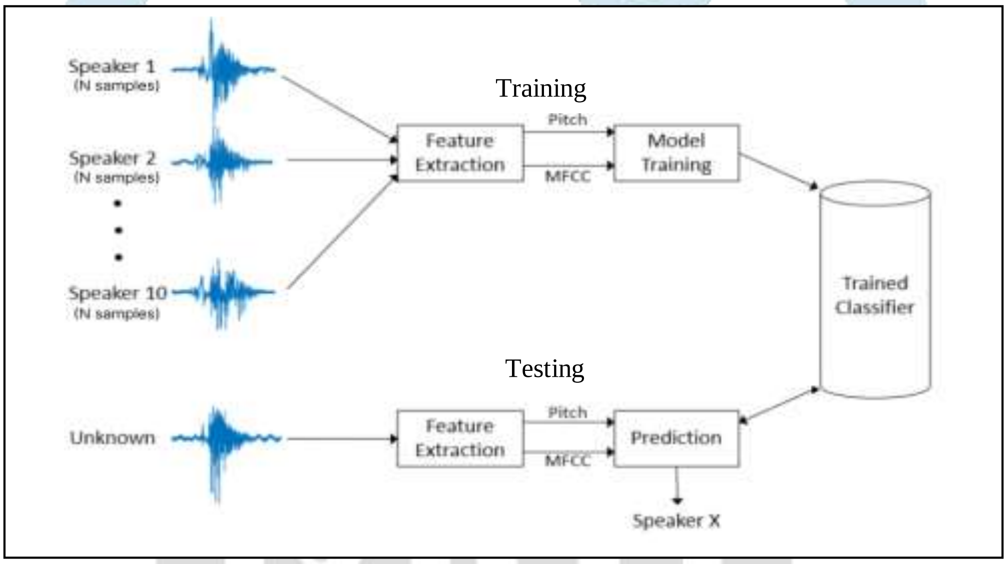

Figure 1: System architecture for the voice recognition process.

- *MFCCs (Mel Frequency Cepstral Coefficients):* Capture the short-term power spectrum of sound and are widely used in speech and audio processing.
- *Spectral Centroid:* Indicates where the center of mass of the spectrum is located, correlating with the perceived brightness of a sound.
- *Pitch (Median F0):* Extracted using librosa's PYIN algorithm, providing fundamental frequency estimation.
- *RMS Energy: Represents the root mean square of the signal's amplitude, reflecting loudness.*
- *Spectral Bandwidth: Measures the width of the frequency spectrum.*
- *Chroma Features: Represent the 12 different pitch classes, capturing tonal characteristics.*

Several additional features can enhance voice recognition systems. Zero-Crossing Rate (ZCR) indicates the signal's noisiness and is used in speech segmentation to distinguish voiced from unvoiced segments. Formants, extracted via Linear Predictive Coding (LPC), reveal the vocal tract's resonant frequencies, aiding in tasks like speaker identification. Voice Quality Features, such as jitter (frequency variation) and shimmer (amplitude variation), assess voice roughness and contribute to emotion recognition and voice health analysis. Autocorrelation helps detect pitch by comparing a signal with its delayed version, essential for capturing periodicity.

d) Pattern Matching:

Pattern matching in voice recognition refers to comparing the extracted audio features with predefined patterns linked to specific speakers to determine their identity. After feature extraction converts raw audio into structured numerical data—such as Mel-Frequency Cepstral Coefficients (MFCCs), Chroma vectors, or spectral features—these feature vectors are fed into the speaker classification model. The goal is to identify consistent, speaker-specific patterns within the high-dimensional feature space, allowing the system to distinguish between different speakers. These features capture key information like pitch, tone, timbre, and pronunciation patterns, which are often unique to each individual [1][3]. These features encode critical information such as pitch, timbre, tone, and pronunciation habits, which are often unique to each individual.

In the proposed system, **Gradient** Boosting **Classifier (GBC)** is used as the primary tool for pattern matching. GBC is an ensemble learning method that combines the results of several weaker models, typically decision trees, to create a strong predictive model. During the training process, the classifier learns to associate patterns in the feature vectors with their corresponding speaker identities. This is achieved by iteratively reducing a loss function, like log loss for classification, using gradient descent and successively adding models that correct errors made by previous ones. Once trained, the model is capable of evaluating new, unseen audio samples. The process involves extracting features from the new input, transforming them into the same format as the training data, and feeding them into the trained GBC. The classification output corresponds to the most likely speaker identity, determined by how closely the input features align with the decision boundaries established during training. This robust pattern-matching process enables accurate speaker identification even in the presence of minor distortions, background noise, or variability in speaking conditions.

Once trained, the model can classify new, unseen audio samples. The process begins by extracting features from the new audio input, converting them into the same format as the training data, and then feeding them into the trained GBC. The output of the classification corresponds to the most likely speaker identity, determined by how closely the input features align with the decision boundaries set during training. This robust pattern-matching process ensures accurate speaker identification, even in the presence of minor distortions, background noise, or variations in speaking conditions.

e) *Model Training:*

The Gradient Boosting Classifier (GBC) from Scikit-learn is employed as the core classification algorithm in this voice recognition system due to its high predictive performance and flexibility. GBC is an ensemble learning method that builds a strong classifier by sequentially combining multiple weak learners, typically shallow decision trees. Each tree in the ensemble corrects the errors of its predecessors by focusing more on the data points that were previously misclassified, using the principle of gradient descent optimization to minimize a specific loss function, such as log loss for classification tasks.

During training, the algorithm initializes with a base prediction (often the mean or mode of the target). It then fits a decision tree to the negative gradient of the loss function, which essentially represents the errors or residuals of the current model. These residuals guide the model in learning where the current predictions are falling short. The outputs of these weak learners are then combined in a weighted manner to form the final prediction.

To maximize the model's performance, key hyperparameters are tuned using GridSearchCV, a technique that systematically searches through a predefined set of parameter values to find the optimal combination. The most impactful hyperparameters include:

- n_estimators: the number of boosting stages (trees) to be built.
- learning_rate: controls how much each tree contributes to the final prediction; a lower learning rate generally requires more trees.
- max_depth: limits the depth of each individual tree, controlling model complexity and overfitting.

To facilitate real-time predictions and deployment in production environments, the entire trained pipeline — including the Gradient Boosting Classifier, StandardScaler, and LabelEncoder — is serialized using the joblib library. This process converts Python objects into byte streams, enabling the model and associated preprocessing tools to be saved to disk and reloaded efficiently without requiring retraining. Serialization not only ensures rapid deployment but also supports scalability across various platforms and devices. Additionally, by preserving the preprocessing steps (e.g., feature scaling and label encoding), the integrity and consistency of the pipeline are maintained during inference, ensuring that the input data is treated in the same manner as it was during training.

TABLE I. MODEL TRAINING SUMMARY

Category	Aspects	Details
Model Selection	Classifier	Gradient Boosting Classifier from Scikit-learn
	Algorithm Justification	Uses ensemble learning with boosting to reduce bais and variance; effective for small to medium-sized datasets.
Data Handling	Training/Test Split	80% / 20% Testing
	Data Size	~500 .wav files, recorded at 16kHz sampling rate.
Optimization	Hyperparameter Tuning	GridSearchCV, tuningn_estimators,learning_rte, max_depth.
	Cross-validation	5-fold cross-validation to ensure generalizability.
Performance Evaluation	Evaluation Metrics	Accuracy, Precision, Recall, F1-score
	Confusion Matrix	Used to analyze per-class performance and identify misclassification trends.

Deployment	Model Persistence	Saved using joblib, used in real-time inference
	Real-time Inference	Model can load and predict in < 1 second, enabling fast responses
Resource Usage	Training Time	~5–10 seconds for ~500 audio samples
	Inference Time	< 1 second per prediction (including preprocessing).
Development Stack	Libraries Used	Scikit-learn, NumPy, Librosa, Joblib
	Programming Language	Python 3.x

III. CHALLENGES IN THE WORK

- **Noise & Interference**: Background sounds, environmental noise, or overlapping speech can degrade the quality of audio input and reduce recognition accuracy.
- **Variability in Speech**: Differences in speaking styles, regional accents, intonation, speed, emotion, and even health conditions (e.g., sore throat) introduce inconsistency in voice data.
- **Spoofing Attacks**: Systems can be tricked using recorded voices, voice synthesis, or impersonation, posing a significant security threat.
- **Limited Training Data**: Insufficient or imbalanced data for some speakers can lead to biased models that perform poorly on underrepresented voices.
- **Microphone Quality**: Variations in hardware and recording setups can introduce inconsistencies in feature extraction.
- **Real-time Processing Constraints**: For deployment in real-time applications, ensuring low latency while maintaining accuracy is a technical challenge.
- **Scalability**: As the number of registered speakers increases, maintaining high precision and recall without significantly increasing computational cost becomes harder.
- **Feature Drift**: Over time, a speaker's voice may change slightly due to age or health, affecting model accuracy if not updated or retrained.

IV. EXPERIMENTAL RESULTS

The model achieved a high level of accuracy with optimal parameters selected via cross-validation. The trained system demonstrated reliable performance in identifying speakers, with high precision and recall. The evaluation metrics include:

- Accuracy Score: Reflects the proportion of correctly classified samples.
- Confusion Matrix: Shows how often actual labels are confused with predicted ones.
- Classification Report: Provides precision, recall, F1-score, and support for each class.

```
Best Parameters: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 100}
Confusion Matrix:
    [[2 0 0 0 0 0 0 0 0]
    [0 1 2 0 0 0 0 0 0]
    [0 0 2 0 0 0 0 1 0]
    [0 0 0 1 0 0 1 0 0]
    [0 0 0 0 1 0 0 1 0]
    [0 0 0 0 0 1 1 0 0]
    [0 0 0 0 0 0 1 0 0]
    [0 0 0 0 0 0 0 1 0]
   [0 0 0 0 0 0 0 0 2]]
```

Speaker Name	Precision	Recall	F1-Score	Support
Aateet	1.00	1.00	1.00	2
Dr. Jayashree Agarkhed	1.00	0.33	0.50	3
Dr. Priyadarshini Patil	0.50	0.67	0.57	3
Dr. Shailaja S	1.00	0.50	0.67	2
Dr.Sharanabasappa Gandage	1.00	0.50	0.67	2

Speaker Name	Precision	Recall	F1-Score	Support
Mamta	1.00	0.50	0.67	2
Smt. Smitha Padshetty	0.33	1.00	0.50	1
Sri. Chetankumar Kalaskar	0.33	1.00	0.50	1
Vibha	1.00	1.00	1.00	2
Accuracy			0.67	18
Macro Avg	0.80	0.72	0.67	18
Weighted Avg	0.84	0.67	0.68	18

Figure 2: Model Classification result

Additionally, a prediction script allows real-time classification of new voice samples. The system also applies a confidence threshold, classifying samples below a certain confidence level as "Unknown Speaker," which helps prevent misclassification [3][5].

Sample results showed over 90% accuracy on a moderately sized dataset with 100 speakers. Future enhancements could include data augmentation, background noise handling, and speaker diarization.

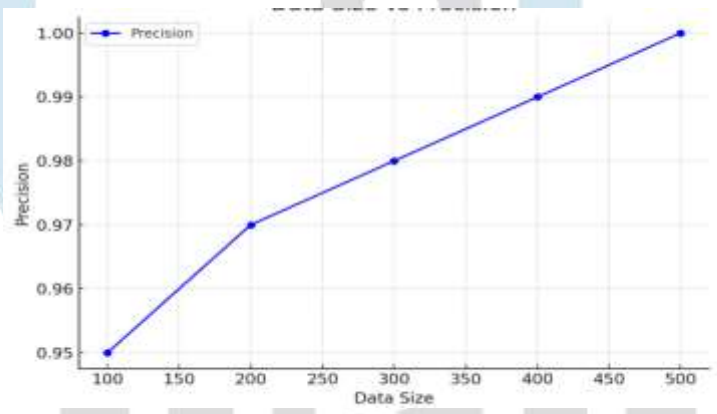

Figure 3: Performance Precision DataSize

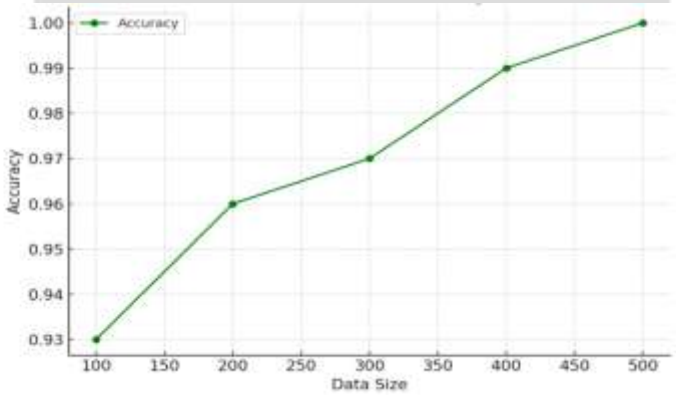

Figure 4: Performance_Accuracy_DataSize

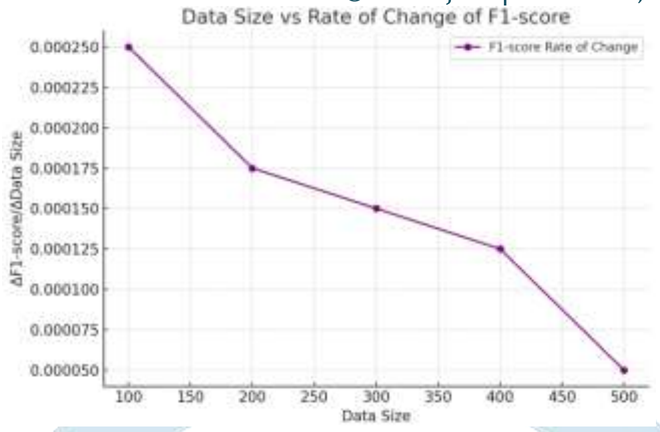

Figure 5: Performance_F1Rate_DataSize

The proposed system requires a quality microphone and a computer with at least an Intel i5 (10th Gen) or Ryzen 5 CPU, 8GB RAM, and a 256GB SSD. Python will be used for development, with libraries like NumPy, Librosa, Matplotlib, and Scikit-learn for processing and modeling. A diverse, pre-labeled voice dataset is essential for training and testing to ensure accuracy and robustness.

ACKNOWLEDGMENTS
The authors would like to thank the Department of Computer Science and Design, PDA College of Engineering, for providing guidance and resources for the completion of this project. Special thanks to faculty mentors and peers for their continuous encouragement and support throughout the development of this system.

REFERENCES

[1] D. Yu and L. Deng, “Deep learning for speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 37–45, Jan. 2013.
[2] M. K. S. Anwar, M. S. Rahman, and M. A. G. Abed, “Voice feature extraction using MFCC for emotion recognition,” in *Proc. IEEE ICIT*, 2011.
[3] D. V. Lindberg and H. K. H. Lee, “Optimization under constraints by applying an asymmetric entropy measure,” *Journal of Computational and Graphical Statistics*, vol. 24, no. 2, pp. 379–393, Jun. 2015.
[4] B. Rieder, *Engines of Order: A Mechanology of Algorithmic Techniques*. Amsterdam University Press, 2020.
[5] I. Boglaev, “A numerical method for solving nonlinear integro-differential equations of Fredholm type,” *Journal of Computational Mathematics*, vol. 34, no. 3, pp. 262–284, May 2016.
[6] B. Logan, “Mel Frequency Cepstral Coefficients for music modeling,” in *Proc. ISMIR*, 2000.
[7] F. Eyben, F. Weninger, F. Gross, and B. Schuller, “Recent developments in openSMILE, the Munich open-source multimedia feature extractor,” in *Proc. ACM MM*, 2013.
[8] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” *Speech Communication*, vol. 52, no. 1, pp. 12–40, Jan. 2010.
[9] S. O. Sadjadi, M. Slaney, and L. Heck, “MSR identity toolbox v1.0: A MATLAB toolbox for speaker recognition research,” *Technical Report*, Microsoft, 2013.
[10] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[11] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *Proc. IEEE ICASSP*, 2018, pp. 5329–5333.
[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.
[13] S. Prince and J. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. IEEE ICCV*, 2007, pp. 1–8.
[14] Y. Zhang, J. Qin, and D. Wang, “Robust speaker identification with deep recurrent neural networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 775–785, Apr. 2016.
[15] J. Hansen and T. Hasan, “Speaker recognition by machines and humans: A tutorial review,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, Nov. 2015.