SN Computer Science (2023) 4:531
https://doi.org/10.1007/s42979-023-02056-w

ORIGINAL RESEARCH

Text-Independent Speaker Recognition System Using Feature-Level Fusion for Audio Databases of Various Sizes

Neha Chauhan[1] · Tsuyoshi Isshiki[1] · Dongju Li[1]

Received: 30 August 2022 / Accepted: 13 June 2023

Abstract

To improve the speaker recognition rate, we propose a speaker recognition model based on the fusion of different kinds of speech features. A new type of feature aggregation methodology with a total of 18 features is proposed and includes mel frequency cepstral coefficient (MFCC), linear predictive coding (LPC), perceptual linear prediction (PLP), root mean square (RMS), centroid, and entropy features along with their delta (Δ) and delta–delta ($\Delta\Delta$) feature vectors. The proposed approach is tested on five different sizes of speech datasets, namely the NIST-2008, voxforge, ELSDSR, VCTK, and voxceleb1 speech corpora. The results are evaluated using the MATLAB classification learner application with the linear discriminant (LD), K nearest neighbor (KNN), and ensemble classifiers. For the NIST-2008 and voxforge datasets, the best SI accuracy of 96.9% and 100% and the lowest speaker verification (SV) equal error rate (EER) values of 0.2% and 0% are achieved with the LD and KNN classifiers, respectively. For the VCTK and ELSDSR datasets, the best SI accuracy of 100% and the lowest SV EER of 0% are achieved with all three classifiers using different feature-level fusion approaches, while the highest SI accuracy and lowest EER achieved on the voxceleb1 database are 90% and 4.07%, respectively, using the KNN classifier. From the experimental results, it is observed that the fusion of different features with their delta and delta–delta values shows an increase in speaker identification accuracy of 10–50%, and the EER value for SV is reduced compared to the value obtained with a single feature.

Keywords Speaker recognition · SV EER · SI accuracy · Feature-level fusion · Biometric system

Introduction

A speaker recognition system can be text-dependent or text-independent. For text-dependent systems, there is some restriction on the type of utterance that the speaker of the system can pronounce (for instance, a fixed pin or certain words in any order, etc.) while the text-independent speaker can say whatever they want. Text-independent speaker recognition is a method of verifying the identity of the speaker without restriction on the speech content. Compared to text-dependent speaker recognition, it is more suitable because the speaker can speak without any restrictions on the system. In the proposed work, all the speakers in the database have different utterances, and hence proposed work is based on text-independent speaker recognition.

Automatic speaker recognition (ASR) is a state-of-the-art technique [1, 2]. Feature extraction and feature mapping are two important processes in SR. Feature extraction extracts several feature vectors called descriptors, and feature matching is used to avoid redundancy present in speech signal features and is used to compare the feature vectors extracted from a signal belonging to an unknown speaker with those extracted from a signal belonging to a known speaker set [3–5]. In the 1980s, the Mel frequency cepstral coefficient (MFCC) was introduced; they use the mel frequency scale, which is a characteristic of popular speech [6]. A comparison of various features was done in Ref. [6]. It was concluded that among all these features, MFCC and LPCC allowed for better performance than other features [6]. The concept of dynamic features was introduced in 1981 by Furui [7]

✉ Neha Chauhan
chauhan.n.aa@m.titech.ac.jp

✉ Tsuyoshi Isshiki
isshiki@ict.e.titech.ac.jp

✉ Dongju Li
dongju@ict.e.titech.ac.jp

[1] Department of Information and Communication Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan

Published online: 18 July 2023

to detect the temporal variability in feature vectors. In addition, in short-term frame energy, the formed transitions and energy modulations also include useful speaker information [7]. The major problem is the deterioration of the performance of ASR systems in the presence of additive noise [8]. Overall, researchers have tried to ensure that recognition systems are noise-resistant in three main ways: (a) statistical models have been adapted to recognize noise (e.g., using parallel model combinations) [9], (b) methods for decreasing the noise in speech signals have been proposed [10, 11], and (c) noise-resistant features have been applied. Several techniques have been designed to address the sensitivity of cepstral features to noise, and various approaches, such as wiener filtering [12], spectral subtraction [13], RASTA [14], and lin-log RASTA [15], have been proposed. The improvement in the cepstral features themselves has not produced satisfactory results. Because of this limitation, more research has been carried out on new features that are more robust to noise in addition to cepstral features.

In order to build a better SR model, it is important to extract additional speaker-dependent information, such as entropy, energy, centroid and prosodic information, with square mean values (RMS). Prosodic features, such as pitch, energy, and RMS, are comparatively less disturbed by channel differences and noise. Although systems based on spectral features, such as MFCC, perform better than prosody-based systems, their combined performance can provide the robustness needed for recognition systems [16, 17]. Prosodic features are those features of speech that deal with the auditory properties of sound, such as stress and pitch [16–22]. One of the important problems related to the degradation of the performance of SR is that the MFCC, LPC, PLP, centroid, entropy, and RMS feature sets contain only static features. A static feature does not capture small changes in the speech signals because the speech signals change very frequently, and consequently, the values of the signals also change rapidly. Therefore, delta and delta–delta feature values are used to add more information and detect feature values over small intervals in speech signals [18]. Many research papers show that models with delta values drastically improve the performance of SI and SV systems [18, 23]. However, from the results of many published research papers [20, 24, 25] and their references, it is clear that prosodic information can also be used to improve SR performance. Usually, feature-level fusion contains more information than a single feature and thus improves the performance of the SR system [26, 27]. This concept of information theory for SR systems is explained clearly in Refs. [28, 29]. The main contributions of the proposed work are as follows:

1. A new methodology for feature aggregation using all 18 features is proposed to obtain the best SR results using various combinations of spectral and temporal speech features with their delta and delta–delta values.
2. The most effective common feature fusion model suitable for speech datasets of various sizes is proposed, for which various experiments are conducted using speech datasets of 5 different sizes.
3. A total of 315 unique feature fusion models (a large number of feature models) are tested on the NIST-2008 database, which contains 18 feature fusion steps. The best 35 models are selected (the best 2 models at each feature fusion step) and tested on the remaining 4 datasets for fast computation to obtain the best feature fusion model.
4. The factors affecting the SR performance are investigated. Feature fusion models suitable for small, medium, and large size voice datasets are proposed.

This paper is mainly divided into related work (Sect. "Related Work"), which defines the previous research done on SI and SV systems using mainly the ELSDSR, voxforge, VCTK, NIST-2008 and voxceleb1 audio databases. The proposed work and methodology section (Sect. "Proposed Work and Methodology") explains the theoretical and practical description of the proposed research. Sect. "Evaluation" consists of database descriptions and a discussion of the results generated. The conclusion and future work are described in Sect. "Conclusion and Future Work".

Related Work

Here, we give a detailed overview of the popular methods used for SI and SV systems, mainly using the ELSDSR, VCTK, voxforge, NIST-2008 and voxceleb1 speech databases. Furui first used the feature concatenation approach for the joint use of cepstral and polynomial features in the form of delta and delta–delta coefficients [7]. The concatenation of MFCC and spectral features enhances the performance of the SR system [30]. In Ref. [31], it is shown that the concatenation of phase information with MFCC enhances speaker performance [31]. In another work, the authors jointly used the statistical pH feature and the concatenation characteristics of MFCC and achieved better performance under noise conditions [32].

In Ref. [33], SI implementation of score-level fusion and feature-level fusion is performed with ELSDSR audio data. The score-level fusion-based system gave a better identification rate of 100% than all other systems using a support vector machine (SVM) [33]. Score-level fusion and feature-level fusion were used in Ref. [34] to calculate SI accuracy using ELSDSR speech data, and the SI accuracy was increased to 95.22% when score-level fusion was used in random forest (RF) and multiclass SVM classification. In Ref. [35], the authors show the potential of deep belief networks (DBNs) in the extraction of short-term spectral features. An accuracy

of 95% is achieved by combining MFCC and DBN features with the Gaussian mixture model-universal background model (GMM-UBM) on the ELSDSR database. In Ref. [36], a two-step approach using the gender and voice information of speakers from ELSDSR was proposed, and the model obtained an improved accuracy of 99.9% with the GMM classifier. Paper [37] presented a simulation study on a transformation-based fusion algorithm for a multimodal biometric authentication system using an ensemble classifier with face scores and voice recognition modules. Using score fusion, true positive rates of 99% and an accuracy of 99.22% are achieved on the ELSDSR voice dataset. The authors in Ref. [38] used the score fusion method with SVM, linear discriminant analysis (LDA) classifier and MFCC, delta MFCC, and delta–delta MFCC features for GMM-UBM modeling. The best EER of 0.02 is obtained with LDA and cosine distance scoring using ELSDSR data. In Ref. [39], a new type of pipeline architecture was proposed, and the fusion of the Gabor filter (GF) and convolutional neural network (CNN) features with RF, SVM, and deep neural network (DNN) classifiers on the ELSDSR database is used. The best accuracy of 94.87% is obtained using the RF classifier for 22 speakers.

In Ref. [40], the authors proposed a new type of feature extraction technique called the twofold information set (TFIS) for a text-independent SR system on three voice datasets, i.e., NIST-2003, voxforge (2015), and VCTK. On the voxforge 2014 database, for the clean voice dataset, the best performance accuracy of 100% and an EER of 0.02 were achieved. On the VCTK dataset, the best SI accuracy of 98.9% and an EER of 0.05 were achieved using TFIS features, and a genuine acceptance rate (GAR) of 0.1% was achieved. In Ref. [41], the authors have proposed a prototypical network loss (PNL)-based speaker embedding model, and a comparison is made with popular triplet loss-based models (TL). The best SI test accuracy and EER achieved using the VCTK database with 90 speakers are 95.63% and 4.08%, respectively, using the PNL technique. Reference [42] showed how the accuracy of SI improves when the fusion of delta and delta–delta features with non-delta features is performed. The best SI accuracy of 94% was achieved after the fusion of MFCC, delta MFCC and delta–delta MFCC with 18 feature vectors using voxforge database. While Ref. [43] showed how the accuracy of SI is increased by fusing models, a new type of generalized fuzzy model (GFM) was implemented and combined with GMM and the hidden Markov model (HMM). The HMM-GFM combination achieves an accuracy of 93% using voxforge data. In Ref. [44], the authors have proposed a speaker verification approach that learns speaker discriminative information directly from the raw speech signal using CNNs in an end-to-end manner. On the Voxforge corpus, the proposed approach yielded a system that outperformed systems based on state-of-the-art approaches. In Ref. [45], a three-step score fusion method was proposed and an SI system was tested with and without adding white Gaussian noise (AWGN) and nonstationary noise (NSN) using MFCC, the power normalized cepstral coefficient (PNCC) and GMM-UBM acoustic modeling. The best SI accuracy of 95.83% was obtained when testing the clean speech data in NIST-2008 [45]. In Ref. [46], a comparison was made between the i-vector model and GMM-UBM on clean and noisy speech of 120 speakers from the TIMIT and NIST-2008 speech datasets with 7 types of score fusion techniques. The highest SI accuracy of 96.67% was achieved using the i-vector approach, while an SI accuracy of 95.83% was achieved using GMM-UBM on clean NIST-2008 data. In reference to [47], the authors have proposed bottleneck (BN) features based on multilingual deep neural networks. Experiments are done on the NIST SRE 2008 female short2-short3 telephone task (multilingual) and the NIST SRE 2010 female core-extended telephone task (English) audio datasets. Tian et al. [47] show that compared to the deep neural network (DNN)-based approach, the BN features based model provides better results for the speaker verification system.

Research paper [48] used the *i*-vector and *x*-vector approaches and proposed attentive pooling for deep speaker embedding for a text-independent speaker verification (SV) system using the voxceleb1 and NIST-2012 voice datasets. Mainly four pooling techniques, including (i) simple average pooling, (ii) statistics pooling, (iii) attentive average pooling, and (iv) attentive statistics pooling, were used in Ref. [48]. From the experimental results, it was observed that the best EER of 3.85% was achieved using attentive statistics pooling (*x*-vector), and with the *i*-vector, the best EER of 5.39% was achieved when the voxceleb1 dataset was used for training and evaluation. In Ref. [49], a fully automated pipeline based on computer vision techniques was used on the voxceleb1 dataset from open-source media. Research paper [49] showed that a CNN-based architecture obtained the best result for an SI accuracy of 80.5% using the top1 classification accuracy, and the best EER of 7.8% was obtained for SV. Table 1 shows the summary of all the related work in ASR using mainly ELSDSR, voxforge, VCTK, NIST-2008 and voxceleb1 database.

Proposed Work and Methodology

Motivation

An analysis of other SR approaches was performed [33–49], and the explanations of these approaches are given in Sect. "Related Work". It is found that there is still room for improvement although the proposed work used various feature fusion approaches to achieve the best SR results. The

Table 1 Related work on speaker recognition using mainly ELSDSR, voxforge, VCTK, NIST-2008 and voxceleb1 database

References	Database	Technique/classifier used	SI accuracy (%)	SV EER (%)
[33]	ELSDSR	Score level fusion, SVM	100	–
[34]	ELSDSR	Score level fusion, random forest (RF) + SVM	95.2	–
[35]	ELSDSR	Feature-level fusion, deep belief network, GMM-UBM	95	–
[36]	ELSDSR	Two step approach (gender + voice), GMM	99.9	–
[37]	ELSDSR	Score-level fusion (face + voice), Ensemble classifier	99.2	–
[38]	ELSDSR	LDA, cosine distance scoring	–	0.02
[39]	ELSDSR	Novel pipeline, Gabor filter, CNN, DNN	94.8	–
[40]	VCTK	Feature-level fusion, improved Hanman classifier (IHC)	98.9	5
[41]	VCTK	DNN, prototype network loss (PNL)	95.63	4.08
[40]	Voxforge	Feature-level fusion, improved Hanman classifier (IHC)	100	2
[42]	Voxforge	Feature-level fusion, probabilistic neural network (PNN)	94	–
[43]	Voxforge	Model fusion, generalized fuzzy model (GFM)	93	–
[44]	Voxforge	Convolutional neural network (CNN)	–	1.18
[45]	NIST-2008	Score-level fusion, log-likelihood ratio, GMM-UBM	95.83	–
[46]	NIST-2008	Score-level fusion, i-vector	96.67	–
[46]	NIST-2008	Score-level fusion, GMM-UBM	95.83	–
[47]	NIST-2008	Deep neural network (DNN)	–	7.26
[47]	NIST-2008	Bottleneck feature, i-vector	–	5.86
[48]	Voxceleb1	Score-level fusion, DNN, x-vector	–	3.85
[48]	Voxceleb1	Score-level fusion, i-vector	–	5.39
[49]	Voxceleb1	Automated pipeline, CNN	80.5	7.8

author believes that the SR result of Ref. [31] can be further improved if more features are tested with more classification methods; hence, this is performed in the proposed work. For a better comparison of the SR performance, all the feature combinations in Ref. [34] should be considered. The proposed methods are tested on 315 different feature combinations with linear discriminant (LD), K nearest neighbor (KNN) and ensemble classifiers using 18 features in total with NIST-2008 data, and the 35 best feature combinations are selected for testing on ELSDSR, voxforge, VCTK and voxceleb1 data to achieve the best SR performance. Banerjee et al. [35] showed that the performance of an SR system can be improved if more features are fused with MFCC. The authors believe that to build a better SR model, it is essential to extract additional speaker-dependent information such as entropy, energy, centroid and prosodic information, such as RMS values (proposed).

Prosodic features contain useful information that is different from the information in cepstral features. Thus, an increasing number of researchers from the SR area have shown interest in prosodic features [36].

Many studies have been performed on feature-level fusion [28, 32, 40–42], but they mainly involve the fusion of MFCC features with other features and MFCC delta and delta–delta values. The proposed method, which uses fusion MFCC, LPC, PLP, RMS, centroid, and entropy information and combinations of their delta and delta–delta values to further improve the SR performance and to find a unique feature fusion model suitable for speech databases of various sizes, is implemented. The main advantage of using feature-level fusion is the recognition of correlated feature values produced by different biometric algorithms, thereby determining a compact set of relevant features that can enhance SR accuracy and remove redundant features to improve the SR results [48].

In addition, it is observed that other research on SR systems includes speech datasets of similar sizes; for example, Refs. [31, 32, 39] included only small-size speech datasets; Refs. [40] and [42–47] performed SR evaluation using only medium-size speech data (more than 100 speakers); Refs. [48, 49] used large speech datasets, while the proposed method is tested on various size speech datasets to find one common model suitable for different sizes of speech datasets. Another problem in the SR system is how different speech features should be combined, and this problem can be solved by the proposed methodology of feature aggregation. Furthermore, the best results obtained by the proposed method are compared with those of some famous SR methods, such as the x-vector, i-vector, and DNN approaches.

Feature Extraction

For the proposed work, the following features and their delta and delta–delta values are used. Mirtoolbox [50] in MATLAB is used to compute the feature vectors of MFCC, centroid, RMS and entropy.

Mel Frequency Cepstral Coefficient (MFCC)

MFCC is the most important and effective aspect of speech-related applications. Since the mid-1980s, MFCCs have been the most popular feature extraction method in the field of ASR. The benefit of using MFCC is that it gives higher accuracy for a less noisy audio dataset. This work mainly focuses on improving speaker recognition accuracy using less noisy data. Hence, MFCC could be a good choice. Following are the steps involved in the extraction of MFCC feature vectors [26, 51–54].

- Framing: For the detailed analysis of speech signals, their parameters are divided into frames as speech signals are continuous in nature. The signals are divided into 20–40 ms frames.
- Windowing: Since speech signals are nonstationary in nature, their parameters change approximately every 10 ms. Hence, hamming window is applied to the frames at 10 ms for MFCC; these are logarithmic in nature.
- FFT: Used for converting a time-domain speech signal into a frequency-domain signal.
- Mel-Filter bank transformation: It is on the same logarithmic scale as the human auditory system, which is also logarithmic in nature. The Mel scale is calculated using Eq. 1:

$$\text{Mel}(f) = 2595 \log 10 \{1 + f/1000\}. \quad (1)$$

where f actual frequency of speech

- LOG: A logarithmic mel scale is applied to the FFT frame, which is linear up to 1 kHz and logarithmic at higher frequencies.

The relationship between the frequency of speech and the mel scale can be established as (Eq. 2):

$$\text{Frequency (mel scaled)} = \left[2695 \log (1 + \text{f(Hz)})/700\right] \quad (2)$$

- DCT: The last step is to calculate the discrete cosine transform, which de-correlates the speech features and arranges them in descending order of information. Hence, the first 13 coefficients are used as MFCC features for building the model [26, 51–54].

Linear Predictive Coding (LPC)

LPC is commonly used because it is fast, simple, and has the capability to extract and store time-varying formant information. This technique of coding uses data encryption to secure the data until it reaches its destination. Also, in our previous research paper [19, 26], it can be seen that fusion of cepstral features, including LPC features, improves SR results, and hence LPC is considered for this work as a feature extraction technique in order to improve the recognition rate.

LPC calculates the current sample using a linear combination of the past samples. Inverse filtering is performed in which the formant is removed from the speech signals, and the remaining signal that is left after inverse filtering is called the residue [55]. The LPC features are calculated using the VQ-LBG algorithm. To reduce the bit rate, VQ is applied to LPC features in the linear spectral frequency (LSF) domain.

It is important to understand the autoregressive (AR) model of speech in order to understand LPCs. An audio signal can be modeled as a pth-order AR process, where each sample is given by Eq. (3):

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n) \quad (3)$$

Each sample at the nth instant depends on 'p' previous samples, added with a Gaussian noise $u(n)$. LPC coefficients are given by a.

Yule–Walker equations are used to estimate the coefficients. The autocorrelation at lag l is given by Eq. (4). R(l) is autocorrelation function:

$$R(l) = a_0 + \sum_{n=1}^{N} (x(n)x(n-1)) \quad (4)$$

The final form of Yule–Walker equations is given by Eqs. (5) and (6):

$$\sum_{k=1}^{p} a_k R(l-k) = R(l) \quad (5)$$

$$\begin{bmatrix} \text{R(0)} & \cdots & \text{R(p}-1) \\ \vdots & \ddots & \vdots \\ \text{R(p}-1) & \cdots & \text{R(0)} \end{bmatrix} \begin{bmatrix} \text{a}_1 \\ \vdots \\ \text{a}_\text{k} \end{bmatrix} = -\begin{bmatrix} R(1) \\ \vdots \\ R(K) \end{bmatrix} \quad (6)$$

Equation 7 gives the final solution to get LPC coefficients:

$$a = -R^{-1} r \quad (7)$$

We have used only the first 13 LPC coefficients in order to reduce system complexity. Details of the proposed LPC features, the VQ-LBG algorithm, and its calculation steps can also be found in Refs. [55–57].

Perceptual Linear Prediction (PLP)

PLP features have better noise reduction, reverberation suppression, and echo cancellation, which leads to an improvement in performance. In addition, in our previous research papers [19, 26], fusion of cepstral features, including the PLP feature, improved SR results, and hence PLP is considered for this work in order to enhance the speaker recognition rate. Following are the steps involved in the extraction of PLP features:

- PLP is very similar to MFCC. It uses equal loudness pre-emphasis and the cube-root compression technique. PLP rejects irrelevant speech information and thus increases the voice recognition rate. The PLP is the same as the LPC, except that its spectral characteristics have been transformed to match the characteristics of the human hearing system.
- After applying the hamming window and FFT function to audio samples and converting them into the frequency domain, they are transformed into a power spectrum. This spectrum is warped into a bark scale using the approximation (Eq. 8):

$$\Omega(\omega) = 6 \cdot \ln\left(\frac{\omega}{1200 \cdot \pi} + \sqrt{\left(\frac{\omega}{1200 \cdot \pi}\right)^2 + 1}\right) \tag{8}$$

  where ω is the angular frequency in rad/s and Ω represents the bark frequency.
- The bark scaled spectra is combined with the power spectra of the critical band filters. The frequency resolution of the ear is measured as a constant on the bark scale. The final samples of the critical band power spectrum with the approximation of the critical band curve $\Psi(\Omega)$ can be written as follows (Eq. 9):

$$\theta(\Omega_t) = \sum_{\Omega} \left(p_{(\Omega-\Omega_t)\cdot\Psi(\Omega)}\right) \tag{9}$$

- Equal loudness pre-emphasis is done in order to compensate for the non-equal perception of loudness at different frequencies using the following Eq. 10:

$$E(\Omega(\omega)) = E(\omega) \cdot \theta(\Omega(\omega)) \tag{10}$$

Here, $E(\Omega)$ is used as an approximation to the non-equal sensitivity of human hearing. After that, perceived loudness $\Gamma(\Omega)$ is calculated by taking the cube root of the intensity, which is known as the power law of hearing (Eq. 11):

$$\Gamma(\Omega) = \sqrt[3]{E(\Omega)} \tag{11}$$

- In the final step of PLP, $\Gamma(\Omega)$ is found by the spectrum using autocorrelation method of all-pole spectral modeling. The inverse DFT (IDFT) is applied to $\Gamma(\Omega)$ to yield the autocorrelation function. The autoregressive coefficients could be further converted into some other set of parameters of interest, such as cepstral coefficients.

The PLP feature extraction steps are clearly explained in Refs. [15, 58, 59]. Thirteen PLP features are calculated by taking the mean of all the PLP features for each voice to reduce the system complexity for MATLAB software [58, 59]. To make the PLP feature dimension equal to the dimension of the other features, the mean value of each frame is calculated, which results in dimensions of 13×1 feature vectors per audio file.

Figure 1 explains the feature extraction steps for MFCC, LPC, and PLP features.

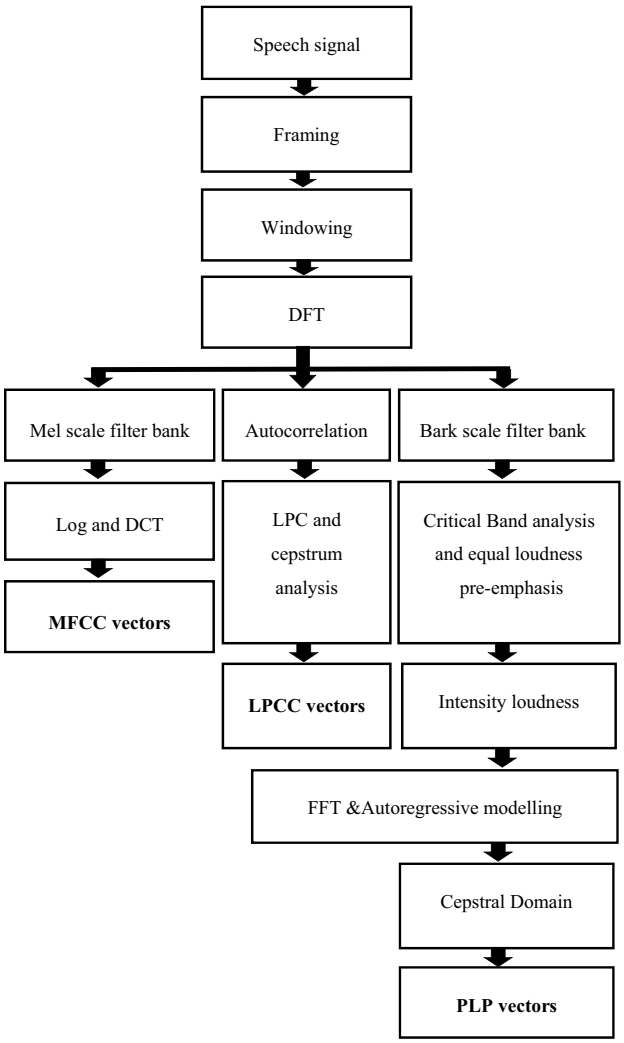

Fig. 1 MFCC, LPC and PLP feature extraction steps

Spectral Centroid (SC)

SC defines the center of gravity of the magnitude spectrum of the short time Fourier transform and gives a single value that represents the frequency domain characteristic of a speech signal. A larger value of SC corresponds to a signal with more energy [60]. A spectral centroid gives a noise-robust estimation of how the dominant frequency of a signal deviates over time. As in our previous research [60], fusion of the spectral centroid with the MFCC enhances the speaker recognition performance; hence, it is also used in the proposed work. It is computed as follows using Eq. (12): $x_i(n)$, where $n = 0, 1,\ldots N–1$, is the sample of the ith frame, and $x_i(k)$, where $k = 0,1\ldots N–1$, are the discrete Fourier transform (DFT) coefficients of the sequence. Then, the centroid $C(i)$ is calculated as follows:

$$C(i) = \frac{\sum_{k=0}^{N-1} k|x_i(k)|}{\sum_{k=0}^{N-1} |x_i(k)|} \tag{12}$$

Spectral Entropy (SE)

Entropy spectral estimation is a spectral density estimation technique that is computed in the following manner. The use of spectral entropy features as additional features showed improvements in the recognition accuracy in Ref. [61].

- For the given signal $x(t)$, $s(f)$, which is the power spectral density, is computed with the Fourier transform of the autocorrelation function of the signal $x(t)$.
- Depending on the frequency of interest, the power in the spectral band is extracted. After calculating the spectral band power, the power in the given band of interest is normalized.
- The spectral entropy is calculated using Eq. 13 [48]:

$$\mathrm{SE} = \sum s(f) * \ln \frac{1}{s(f)} \tag{13}$$

Root Mean Square (RMS)

Prosodic features such as pitch, energy, RMS, and duration are less affected by channel differences and noise. Although systems based on spectral features, such as MFCC, give better SR performance than prosody-based models, their combination may provide the robustness needed by recognition systems [16, 17]. RMS is a measure of the loudness of an audio signal. It is found by calculating the square root of the sum of the mean squares of the amplitudes of the sound samples. The RMS formula is given in Eq. 14 [62], where $x_1, x_2,\ldots x_n$ are n observations, and x_{rms} is the RMS value for the n observations:

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} xi^2} \tag{14}$$

Delta Features

Delta features are used to calculate the rate of change in speech power to the change in the short time power of the noise. Using the delta function, small changes can be calculated for speech features. Delta (Δ) and delta–delta ($\Delta\Delta$) features can also be used to compute small dynamic information of the speech signals [7, 42]. The performance of the system with and without delta values is observed in the proposed work. For a feature f_k and the time constant k, Δ (Eq. 15) and $\Delta\Delta$ (Eq. 16) are calculated as

$$\Delta_k = f_k - f_{k-1} \tag{15}$$

$$\Delta\Delta_k = \Delta - \Delta_{k-1} \tag{16}$$

Classification

The LD (Fig. 2), KNN (Fig. 3), and ensemble (Fig. 4) classification techniques are used for the proposed work. All classification tasks are performed using the classification learner application in MATLAB.

For the proposed work, LD is used as a classifier. LD approaches obtain a linear combination of features to distinguish two or more classes, and the resulting combination is used as a linear classifier. All the features are used in the model for LD classification.

LD uses Bayes' Theorem to find the probabilities. If the output class is (k) and the input is (x), here is how Bayes' theorem works to estimate the probability that the data belongs to each class (Eqs. 17 and 18):

$$P(Y = x|X = x) = (PIk * fk(x)) / \mathrm{sum}(PII * fl(x)) \tag{17}$$

$$PIk = nk/n \tag{18}$$

In the above equation, PIk is the prior probability. This is the base probability of each class as observed in the training data. $f(x)$ is the estimated probability that x belongs to that class. $f(x)$ uses a Gaussian distribution function; n is the number of instances; and K is the number of classes. Plug the Gaussian into the equation above and simplify, and we

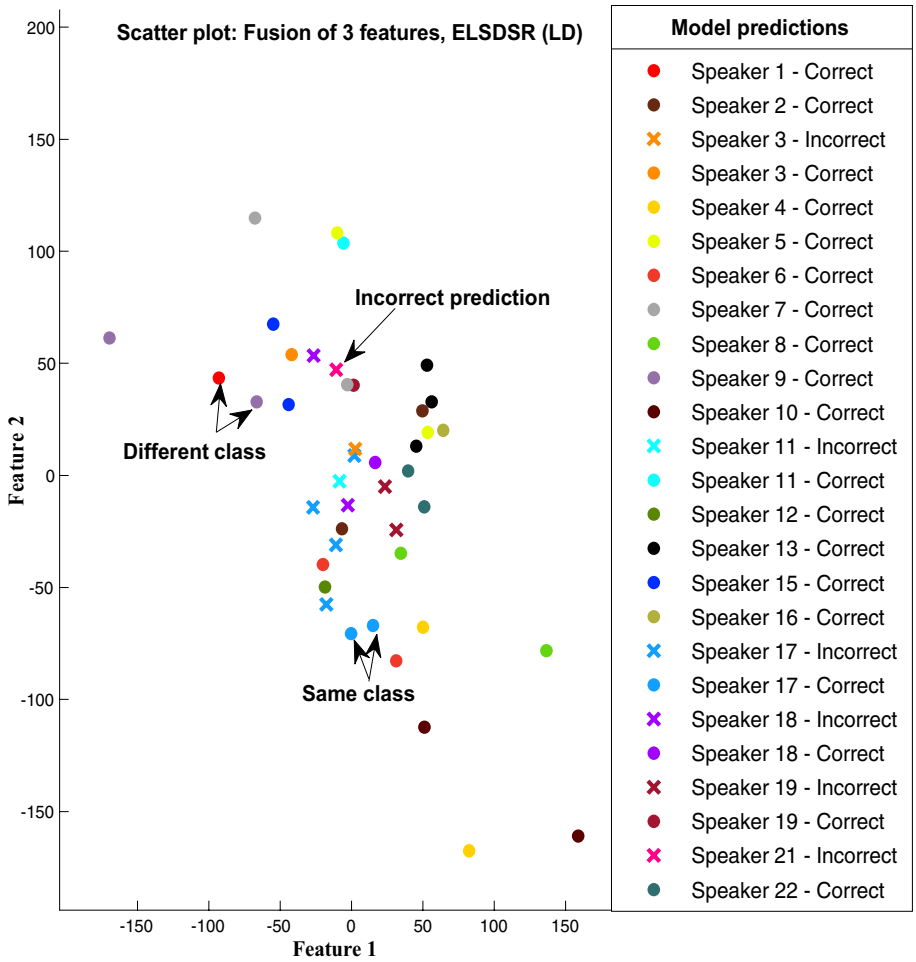

Fig. 2 Linear discriminant (LD) classification

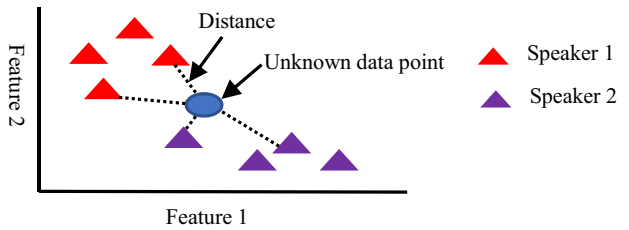

Fig. 3 KNN classification

find ourselves with the Eq. 19. This is a discriminating function, and the class computed as having the greatest value will be the output classification (y):

$$\mathrm{Dk}(x) = x * (muk/\mathrm{sig}a^2) - (muk^2/(2 * \mathrm{sigma}^2)) + \ln(\mathrm{PIk}) \quad (19)$$

Dk (x) is the discriminate function for class k given input x, the muk, sigma^2 and PIk are all calculated from your data. Detail explanations of LD classification can be found in Refs. [63, 64].

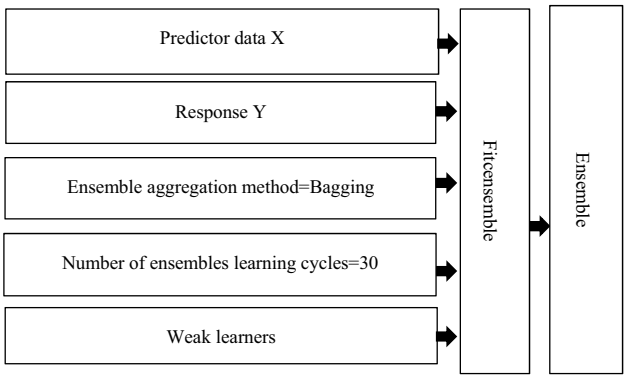

Fig. 4 Ensemble classification

Figure 2 shows the scatter plot for prediction data points for 22 speakers using the LD classifier when tested on ELSDSR speech data for the combination of MFCC, $\Delta\Delta_{\mathrm{entropy}}$, and ΔPLP features which consists of total 13(MFCC) + 1($\Delta\Delta_{\mathrm{entropy}}$) + 13(ΔPLP) = 27 feature vectors. Feature 1 (x-axis) and feature 2 (y-axis) data points are shown in Fig. 2 out

of a total of 27 feature vectors for 22 speakers (22 different classes). A different-colored dot indicates it belongs to a different speaker or class, while the same-colored dot indicates it belongs to the same speaker or class, while *x* indicates an incorrect prediction of speaker or class.

The KNN approach is used to classify a set of unknown data points based on their similarity with a neighbor. Here, *K* is the number of dataset elements that help in classification; for the proposed work, *K* is taken as 1. The KNN algorithm can be explained based on the following steps:

- Select the number *K* of the neighbors.
- Calculation of the Euclidean distance for *K* neighbors.
- Select the *K* nearest neighbors according to the calculated Euclidean distance.
- Count the number of data points in each class among these *k* neighbors.
- Assign the new data points to that class for which the number of the neighbor is more.

The KNN algorithm is explained in Ref. [65]. Figure 3 shows the structure of KNN for two different speakers.

The ensemble classification method helps improve the SR results by combining different models and reducing the risk of overfitting [66]. The random subspace ensemble method is used with a determinant learner (the number of learners is 30 and the subspace dimension is 5) in the proposed work for the ensemble classifier. The random subspace ensemble approach, also called bagging or feature bagging, is a machine learning algorithm that combines the predictions from many decision trees trained on various subsets of columns in the training dataset and decreases the correlation between estimators in an ensemble by training them on random samples of features instead of the all-feature set [67]. The fitcensemble function in MATLAB is used to train ensemble classification. Let *X* be a data matrix. Each row contains a single observation, and each column contains a single predictor variable. *Y* is the vector of responses and has an equal number of observations as the rows in *X*. Figure 4 shows the ensemble classification creation details and the information used for ensemble classification.

An ensemble of models using the random subspace method can be calculated using the following algorithm.

- Assume that the number of training points is *N* and the number of features in the training data is *D*. *L* be the number of individual models in the ensemble.
- For each individual model *L*, select n_l ($n_l < N$) to be the number of input points for l. It is mutual to have only one value of n_l for all the individual models.
- For each individual model l, generate a training set by selecting d_l features from *D* with replacement and train the model.
- Now, to apply the ensemble model to a hidden point, combine the outputs of the *L* individual models by majority voting [66].

Feature Fusion and Model Optimization Steps

Feature Fusion Methodology

A methodology for feature aggregation using one dataset (NIST-2008) is developed, and the 2 best models with the highest SI accuracy and the lowest SV EER values are selected at each step for testing the remaining 4 datasets to reduce the complexity of testing many models, since testing all combinations is impractical and therefore requires a heuristic algorithm to find the best feature combinations. To increase the probability of obtaining an effective model that is suitable for all the databases used, the two best models are selected at each evaluation step using the NIST-2008 database, which involves training and testing a total of 315 models. In case the selection of the two best models from each step does not give satisfactory results, the three best models are selected; and this process continues with the same methodology of feature fusion until a final effective model suitable for all databases used is found, which involves the training and testing of approximately 475 models (where three models are selected), and feature fusion becomes particularly complex. Table 2 shows the total number of models

Table 2 Total number of models tested on the NIST-2008 database

Step	Feature fusion number	Total number of models training using 1, 2 and 3 best models at each feature fusion step		
		1 model	2 models [Proposed]	3 models
1	1	18	18	18
2	2	17	34	51
3	3	16	32	48
4	4	15	30	45
5	5	14	28	42
6	6	13	26	39
7	7	12	24	36
8	8	11	22	33
9	9	10	20	30
10	10	9	18	27
11	11	8	16	24
12	12	7	14	21
13	13	6	11	18
14	14	5	9	15
15	15	4	7	12
16	16	3	3	9
17	17	2	2	6
18	18	1	1	1
		Total = 171	Total = 315	Total = 475

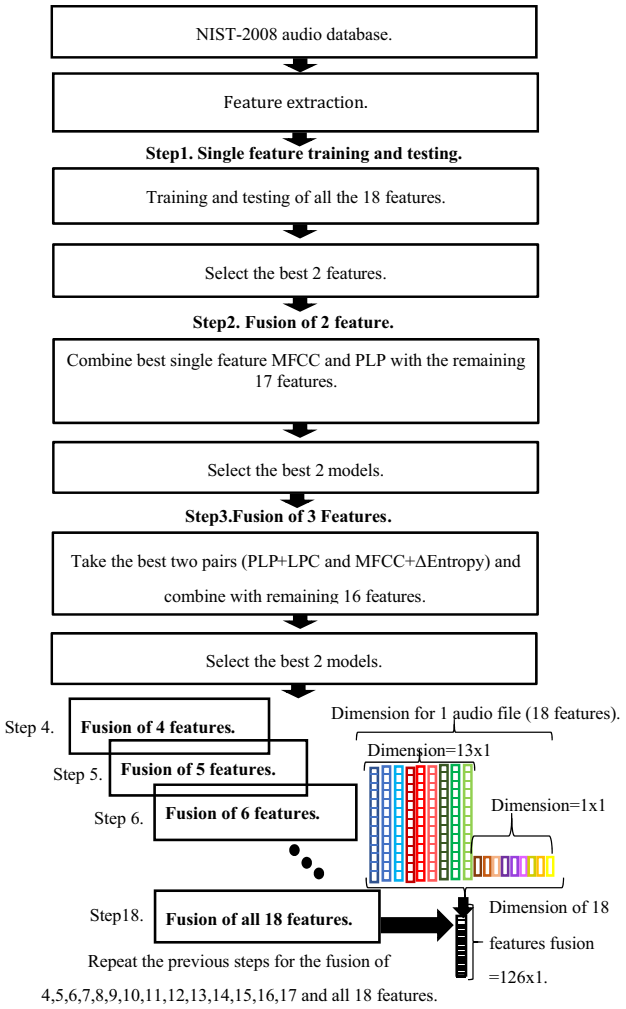

Fig. 5 Methodology for feature fusion using NIST-2008

tested when the best 1, 2 and 3 models are selected in each step of feature fusion. Figures 5 and 6 explain the feature fusion methodology and workflow, respectively. Figure 7 shows the computational steps for the SI and SV systems.

Model Optimization

The main goal of selecting the best 2 models at each step of feature fusion using NIST-2008 data and model optimization is to achieve a common effective model that is suitable for all sizes of speech datasets. Table 2 shows the total number of models tested when the best 1, 2 and 3 models are selected in each step of feature fusion. Some steps used to select the 2 best models apply a smaller number of models due to the repetition of features.

Optimization Algorithm

The following steps show how different features are fused and how the best models are selected. Figure 8 shows how the models are optimized to obtain an effective model suitable for various sized voice datasets. Table 3 shows the dimensions of each feature.

1. MFCC, LPC, PLP, centroid, RMS, and entropy features and their delta and delta–delta feature vectors are extracted for all 5 voice datasets using MATLAB software. Table 3 shows the total number of feature vectors extracted for one audio file. The input matrix files are created using single features and combinations of features in MATLAB. The labeling of each speaker is performed with the features values so that a MATLAB classification learner application can be used for testing (Figs. 5, 6).
2. The first step of feature fusion involves training and testing all 18 features individually and then selecting the best 2 features with the highest SI accuracy and lowest average EER among all 18 features. To select the best model, the average accuracy and average EER values of the three classifiers are considered. PLP and MFCC are the first- and second-best models, respectively, because they have the highest average accuracy values of 80% and 61.1% and the lowest EER values of 5.2% and 15.4%, respectively, compared to other features. Equation 20 shows the calculation of the average accuracy and EER values using all three classifier results:

$$\text{Average result} = \frac{\text{LD} + \text{KNN} + \text{ensemble (accuracy or EER)}}{3} \tag{20}$$

3. In the second step, 2 features are fused by combining the best features, MFCC and PLP, separately with the remaining 17 features, and again, the best two models are selected from this step. The two best models from this step are the MFCC and Δentropy fusion model and the PLP and LPC fusion model.
4. In the third step, 3 features are fused by combining the remaining 16 features separately with the two best models selected from step 2. Fusion of 4 to 18 features is performed in the same way, and the two best models are selected at each step. A total of 315 models are tested on the NIST-2008 data, and 35 feature models are selected for testing on the remaining 4 databases for fast computation.

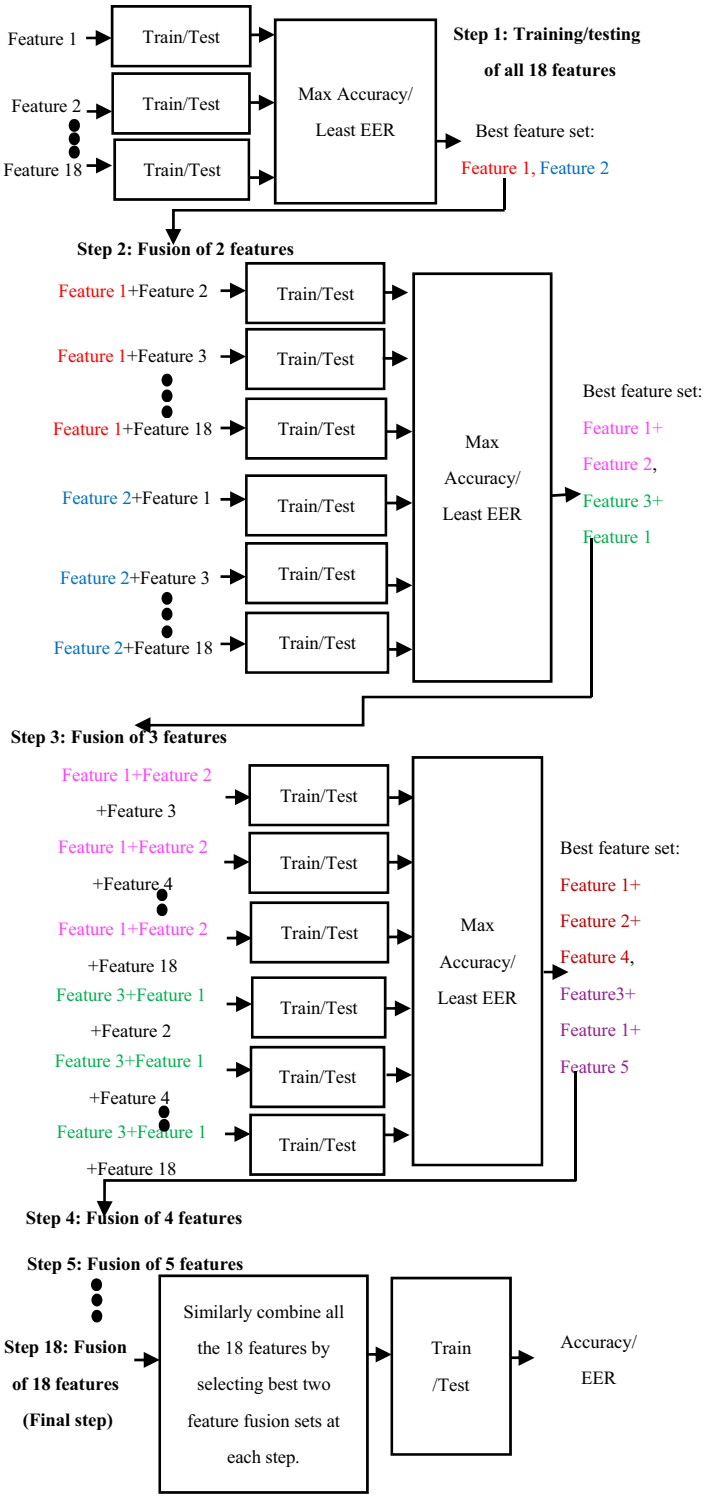

Fig. 6 Flow diagram for selecting the best model using NIST-2008

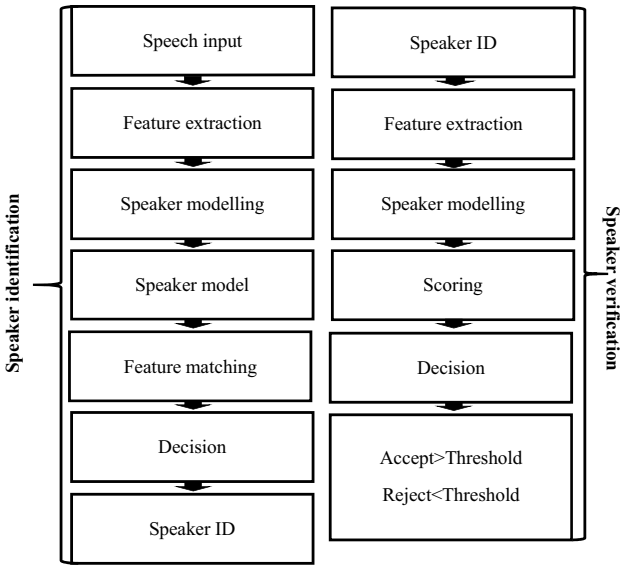

Fig. 7 SI/SV computation steps

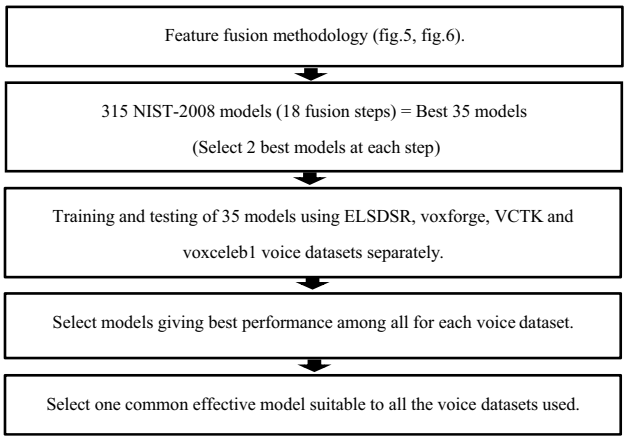

Fig. 8 Model optimization

Table 3 Feature dimensions

Feature	Number of feature vector for 1 audio file (row × column)
MFCC, ΔMFCC, ΔΔMFCC	13 × 1,13 × 1,13 × 1
LPC, ΔLPC, ΔΔLPC	13 × 1,13 × 1,13 × 1
PLP, ΔPLP, ΔΔPLP	13 × 1,13 × 1,13 × 1
Centroid, ΔCentroid, ΔΔCentroid	1 × 1,1 × 1,1 × 1
RMS, ΔRMS, ΔΔRMS	1 × 1,1 × 1,1 × 1
Entropy, ΔEntropy, ΔΔEntropy	1 × 1,1 × 1,1 × 1

5. The best models that obtain the highest accuracy and lowest EER values on each speech dataset are selected, and one common effective model that obtains the best result on all 5 datasets is found.

Evaluation

Database Preparation

The following five voice databases are used in the proposed work, and their explanations are as follows. All the following datasets are text-independent. Only the speaker's voice is important for the recognition purpose, content of the speech is not important, and the speaker can speak freely in a text-independent speaker recognition system.

ELSDSR is a small corpus dataset that was recorded at the Technical University of Denmark (DTU) by faculty, Ph.D. students, and master's students. ELSDSR consists of voice messages from 22 speakers, 12 males and 10 females. For training, 154 voices were recorded with 7 sentences each. For the testing set, 44 utterances were provided, and 2 sentences were spoken by each speaker. The time duration for the training data is 78 s for males and 88.3 s for females. The test data duration is 16.1 s for males and 19.6 s for females [68].

1. Voxforge is an open speech dataset (medium size) consisting of many speaker voices. For the proposed work, 100 English speakers are randomly selected. Each speaker spoke 10 sentences recorded at a sampling rate of 8 kHz. A total of 1000 voice files are used for 100 speakers. Out of the 1000 voices, 800 voices are used for training, and 200 voices are used for testing [40].
2. The CSTR VCTK (medium size) corpus consists of speech data from 109 native English speakers with different accents. Each speaker recorded approximately 400 English sentences. For the proposed work, 5 sentences from each speaker are selected. A total of 545 voices are used. A total of 436 voices are used for training, and the remaining 109 voices are used for testing [40].
3. A total of 942 h of multilingual telephone speech and English interview speech are included in the NIST-SRE-2008 database (medium size) [46, 69]. The sampling frequency was converted from the original 8–16 kHz, and 120 English-only microphone channels were selected for better comparison with other databases. Audacity software [70] is used to separate a single speaker from multiple speakers and segment the speaker voice into 10 equal parts. Each speaker consists of 10

Table 4 Database details

Information	ELSDSR (small dataset)	Voxforge (medium dataset)	VCTK (medium dataset)	NIST-2008 (medium dataset)	Voxceleb1 for SI (large dataset)	Voxceleb1 for SV (large dataset)
Total number of speakers	22	100	109	120	1251	1251
Each speaker utterance	9	10	5	10	Undefined	Undefined
Total utterance for training	154	800	436	720	145,265	148,642
Total utterance for testing	44	200	109	480	8251	4874
Total number of audio recordings	198	1000	545	1200	153,516	153,516
Source	Open	Open	Open	Linguistic Data Consortium	Open	Open
Language	English	English	English	English	English	English
Environment	Clean	Clean	Clean	Clean	Multimedia	Multimedia

audio files with a fixed length of 8 s each. Six audio files are used for training, and the remaining 4 audio files are used for testing. A total of 1200 voices are used, out of which 720 voices are used for training and 480 voices are used for testing.

4. Voxceleb1 (large size) contains more than 100,000 voice samples. Videos included in the database are recorded in challenging multispeaker environments, including at red carpet events and in outdoor stadiums. All the datasets are degraded with real-world noise, such as laughter, overlapping speech and room acoustics. For this paper, all 1251 pieces of speaker data are used for a total of 153,516 speaker voices. To obtain a fair comparison with Refs. [48] and [49], 148,642 utterances are used for training and 4874 utterances are used for testing in the SV task; in addition, 145,265 utterances are used for training, and 8251 utterances are used for testing in the SI task. Table 4 provides the details of all the voice datasets used.

Evaluation Using the 5 Databases

Performance Evaluation for Speaker Identification

The SI accuracy (%) is calculated using a classification learner application in MATLAB for all proposed models using all 3 classifiers. The proposed work determines the overall SI accuracy of the system for comparison with systems in other work. The accuracy (Eq. 21) indicates how many voice samples of all speakers are correctly identified from the total number of voice samples. The feature fusion models that provide better SR results are shown in the results of tables.

$$\text{Accuracy} = \frac{\text{Number of voices correctly identified}}{\text{Total number of audio files}} \quad (21)$$

Performance Evaluation for Speaker Verification

ROC curves of different models are plotted using the false-positive rate (FPR) and true-positive rate (TPR) for each speaker at an interval of 0.005. The EER is calculated from the intersection of the ROC curve and the diagonal axis

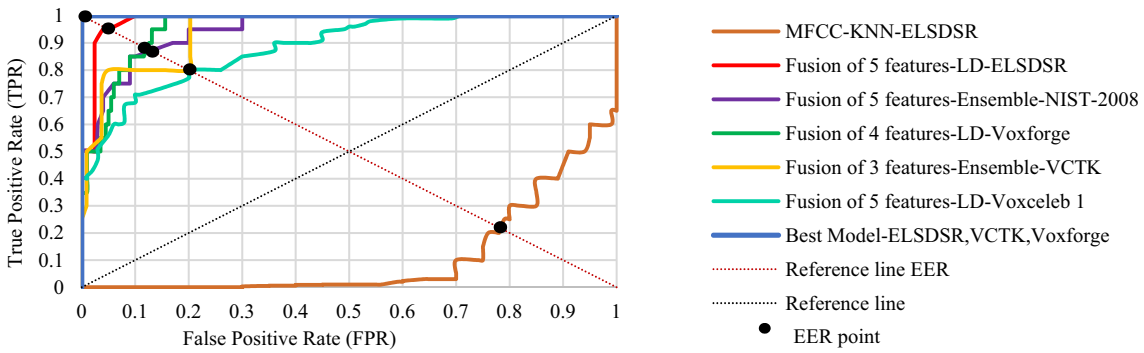

Fig. 9 Schematic diagram of ROC curves and the EER calculation

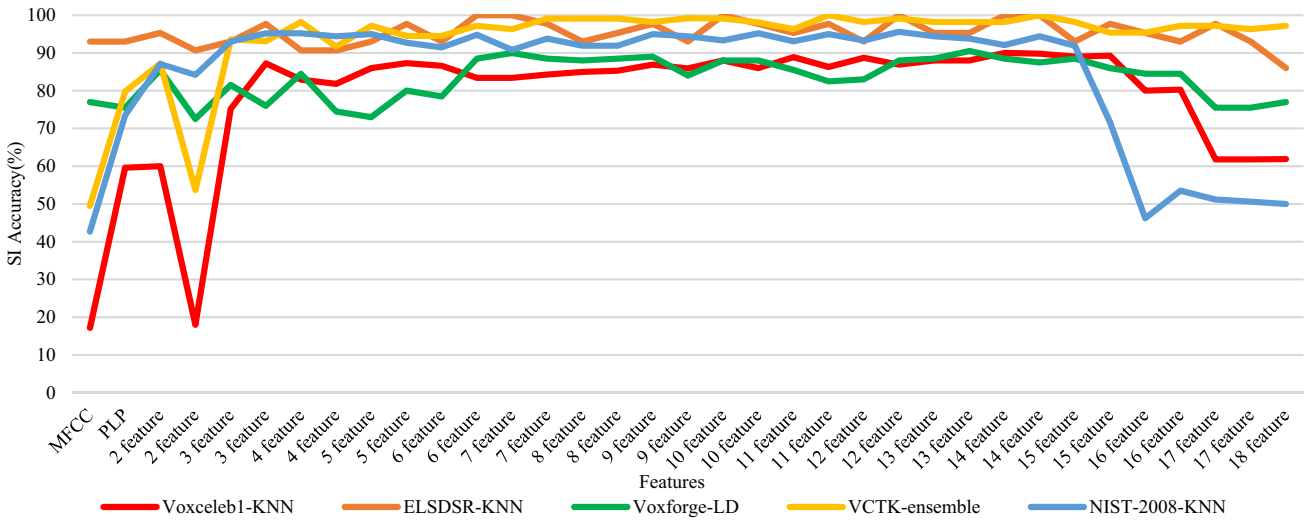

Fig. 10 SI performance on the ELSDSR, Voxforge, VCTK, NIST-2008 and voxceleb1 audio datasets

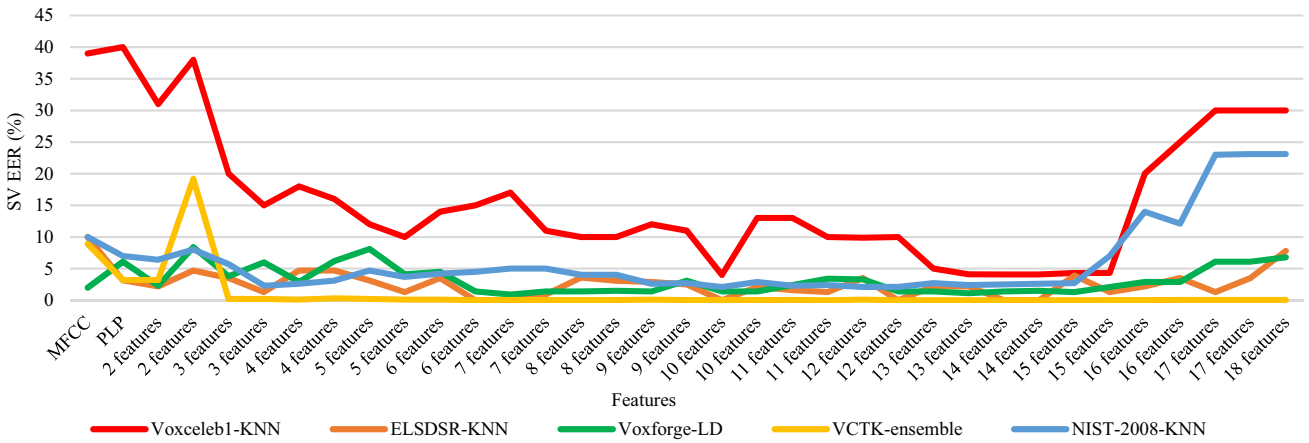

Fig. 11 SV performance on the ELSDSR, Voxforge, VCTK, NIST-2008 and voxceleb1 audio datasets

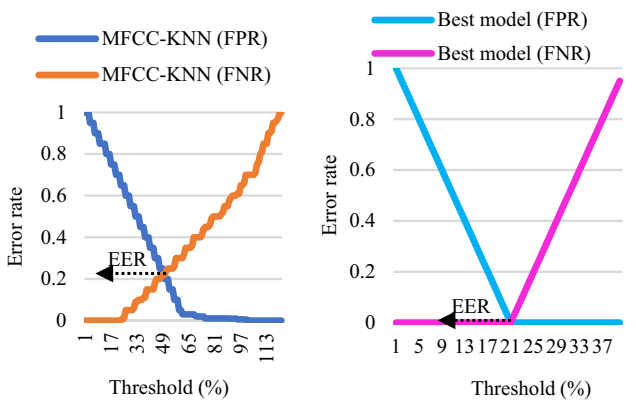

Fig.12 FPR, FNR and EER on different threshold value, for different features fusion models, ELSDSR database

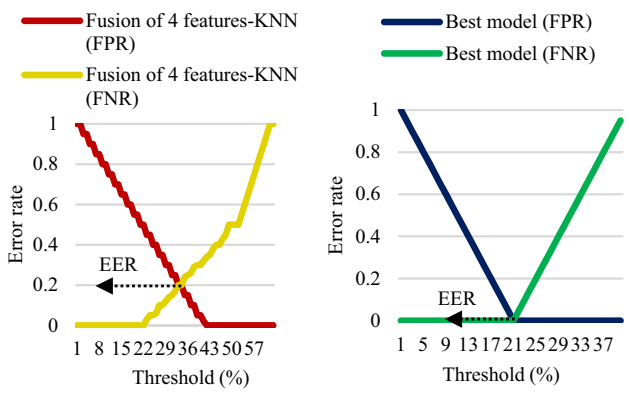

Fig.13 FPR, FNR and EER on different threshold value, for different features fusion models, voxforge database

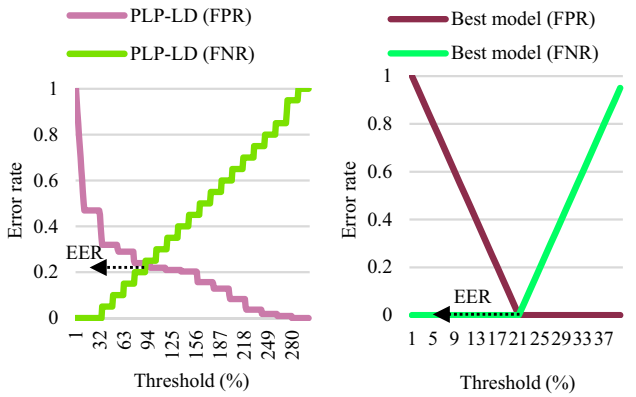

Fig.14 FPR, FNR and EER on different threshold value, for different features fusion models, VCTK database

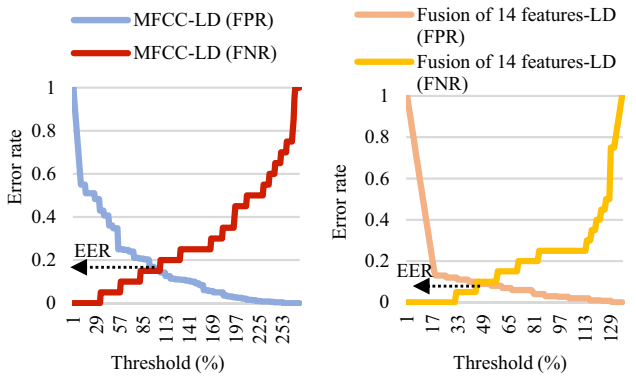

Fig.15 FPR, FNR and EER on different threshold value, for different features fusion models, NIST-2008 database

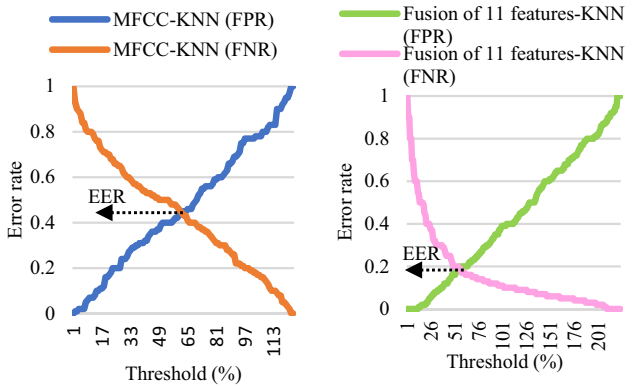

Fig.16 FPR, FNR and EER on different threshold value, for different features fusion models, voxceleb1 database

from (0, 1) to (1, 0) and is the value of the false positive rate. For each speaker, a selected speaker is considered a true speaker, and the rest of the speakers are collectively considered impostors. Figure 9 shows the schematic diagram for the ROC curve and how the EER value is calculated. The final EER values are shown in the result tables. A perfect ROC curve includes a straight line from the starting point (0.0, 0.0) to the upper left corner (0.0, 1.0) and a straight line from the upper left corner to the upper right corner (1.0, 1.0) [71–74], as shown in Fig. 8 (blue line-best models), which means that the classifiers give 0 false positives and 0 false negatives and are absolutely accurate. Similar ROC curves are generated by the model that gives the best results. ROC curves near the left corner indicate better results [72, 73]. Figures 10 and 11 show how SI accuracy and SV EER change with different feature fusion models. Only a few models are considered in Figs. 10 and 11 to show the variation in SR performance because including all the models makes the graph unclear.

Two of the most popularly used measures in biometrics are the false positive rate (FPR) and the false negative rate (FNR). The FPR is the ratio between the number of false acceptances and the total number of imposter attempts. Hence, it measures the likelihood that the biometric model will falsely accept access from an imposter (see Figs. 12, 13, 14, 15, 16). The FNR calculates the likelihood that the biometric model will incorrectly reject a genuine speaker, and it represents the ratio between the number of false recognitions and the total number of speaker attempts [72]. Figures 12, 13, 14, 15, and 16 show the graphs for the false positive rate (FPR) and false negative rate (FNR) at different thresholds and the equal error rate (EER) (%) (EER = FPR = FNR) for a few features of the fusion model for all the speech databases used.

Figure 12 shows the error rate graph at different threshold values with MFCC features using a KNN classifier, and the best model (Table 5) gives the lowest EER of 0 and 100% accuracy for ELSDSR speech data. Figure 13 shows the error rate graph at different threshold values with the fusion of 4 features (PLP + LPC + ΔPLP + MFCC) using KNN classifier, and the best model (Table 6) gives the lowest EER of 0 and 100% accuracy (fusion of 7 features) for voxforge speech data. Figure 14 shows the error rate graph at different threshold values with the PLP feature using the LD classifier, and the best model (Table 7) gives the lowest 0 EER and 100% accuracy for the VCTK speech data (7). Figure 15 shows the error rate graph at different threshold values with the MFCC feature using the LD classifier and the best model having the fusion of 14 (Table 8) for NIST-2008 data. Figure 16 shows the error rate graph at different threshold values with the MFCC feature using the KNN classifier and the fusion of the 11 feature with the KNN (Table 9) for voxceleb1 data. From Figs. 12, 13, 14, 15, and 16, we can observe that the error rate decreases when more feature combinations are used. An error rate graph of only

Table 5 Best feature fusion models on the ELSDSR audio datasets (proposed best models vs. other best models)

Method	Features used (model)	Classifier, modeling	Number of speakers	SI accuracy (%)	SV EER (%)
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC (6 features)	KNN	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔMFCC (7 features)	KNN	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔMFCC + ΔRMS + ΔLPC (9 features)	LD	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔMFCC + ΔRMS + ΔMFCC + ΔLPC (10 features)	KNN	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔRMS + ΔLPC + entropy Δentropy (12 features)	KNN	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔMFCC + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	KNN, Ensemble	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	LD, KNN	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS + ΔΔMFCC (15 features)	Ensemble	22 (198 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + centroid + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS + ΔΔentropy (15 features)	LD	22 (198 audios)	100	0
Score-level fusion [34]	Zero crossing rate (ZCR), short time energy (STE), entropy, spectral centroid, entropy, beam forming	Random forest (RF) + support vector machine (SVM)	22 (198 audios)	95.2	–
Feature-level fusion [35]	Deep belief network (DBN) Layers, MFCC	GMM-UBM	22 (198 audios)	95	–
Novel pipelined [39]	Gabor filter GF), convolutional neural network (CNN)	SVM, RF, Deep neural network (DNN)	22 (198 audios)	94.8	–

Table 6 Best feature fusion models on the voxforge audio datasets (proposed best models vs. other best models)

Method	Features used (model)	Classifier, modeling	Number of speakers	SI accuracy (%)	SV EER (%)
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔMFCC (7 features)	KNN	100 Speakers, (1000 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	KNN	100 Speakers, (1000 audios)	99.5	0
Feature-level fusion [40]	Two-fold information set (TFIS), MFCC, delta MFCC, delta–delta MFCC	SVM, KNN, Improved Hanman Classifier (IHC)	100 Speakers, (1000 audios)	100	2
Feature-level fusion [42]	MFCC, delta MFCC, delta–delta MFCC	Probabilistic neural network (PNN)	5 Speakers, (750 audios)	94	–
Model fusion [43]	MFCC	GMM, hidden Markov model (HMM), generalized fuzzy model (GFM)	100 Speakers, (1000 audios)	93	–
Feature fusion [44]	MFCC	Convolutional neural network (CNN)	100	–	1.18

a few combinations is used to show the effectiveness of the proposed method and to show that SR performance improves using more feature fusion.

Comparison of Results and the Best Models

Tables 5, 6, 7, 8, and 9 show the best model results, which provide the top SR performance (with the highest SI accuracy and lowest SV EER values) compared to other fusion models, and the results generated by other models. For a fair comparison, we included only the results of the other models that use the same database and number of audio clips as used by the proposed model. In addition, Tables 5, 6, 7, 8, and 9 show a comparison of the proposed best model results (red font), the common effective model results on all datasets (bold red font) and other models' best results on the ELSDSR, voxforge, NIST-2008, VCTK and voxceleb1 databases, respectively. The following points explain the number of best models (those that obtain the highest SI accuracy and lowest SV EER values) obtained by the proposed work and one effective model suitable for all the datasets:

1. For the ELSDSR database, the best average SV EER of 0% and SI accuracy of 100% are achieved by the proposed work using a fusion of 6, 7, 9, 10, 12, 14, and 15 features; however, for ELSDSR data, less research has been performed for SV; hence, previous results are difficult to compare. The highest SI accuracies of 95.2% [34], 95% [35], and 94.8% [39] are obtained with score-level fusion and GMM-UBM modeling, respectively. As ELSDSR is a small voice dataset, the proposed models with the fusion of 6, 7, 9, 10, 12, 14 and 15 features can be considered suitable for small audio databases (Table 5).
2. For the voxforge speech database, the proposed models with a fusion of 7 and 14 features obtain the lowest EER value of 0% and the highest SI accuracy of 100% and 99.5% with the KNN classifier, while previous SR models on voxforge speech data [40] achieved an EER value of 0% using the feature-level fusion method when the same number of voices were used for training and testing as used by the proposed model. The best SI accuracies of 94% and 93% are achieved by Refs. [42, 43] using feature fusion and model fusion, respectively (Table 6). While it can be seen from Ref. [44] result that 1.18% EER is achieved using CNN-based approach for voxforge data.
3. For the NIST-2008 database, the proposed models achieve the best EER of 0.2% and the best SI accuracy of 96.9% using a fusion of 11, 12, and 14 features with an LD classifier, and the score fusion method with GMM-UBM modeling [45] achieves the best accuracy of 95.83%. The model in Ref. [46] achieved the best SI accuracy of 96.6% with an *i*-vector approach, and with GMM-UBM modeling, a best accuracy of 95.83% is achieved when tested on the same NIST-2008 data. References [45] and [46] included only SI results; therefore, only the SI results of proposed work can be

Table 7 Best feature fusion models on the VCTK audio datasets (proposed best models vs. other best models)

Method	Features used (model)	Classifier, modeling	Number of speakers	SI accuracy (%)	SV EER (%)
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + ΔΔPLP (5 features)	LD	109 Speakers (545 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + Δ ΔRMS + ΔΔPLP (8 features)	KNN	109 Speakers (545 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + Δ ΔRMS + ΔΔPLP + ΔLPC (9 features)	LD, KNN	109 Speakers (545 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + Δ ΔRMS + ΔΔPLP + entropy + ΔLPC (10 features)	LD	109 Speakers (545 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔ LPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + Δentropy + ΔLPC + entropy (11 features)	LD, Ensemble	109 Speakers (545 audios)	100	0
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + Δ ΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	LD, KNN, Ensemble	109 Speakers (545 audios)	100	0
Feature-level fusion [40]	TFIS, MFCC, delta MFCC, delta–delta MFCC	SVM, KNN, Improved Hanman classifier (IHC)	109 Speakers, (545 audios)	98.9	5
Prototypical network loss [41]	MFCC	DNN Prototype network loss (PNL)	90 (450 audios)	95.63	4.08

compared with the results in Refs. [45, 46] (Table 8). In Ref. [47], DNN and bottle neck-based techniques is used, and NIST-2008 data are used only for testing. For English NIST-2008 data, best EER of 5.86% is achieved using bottleneck *i*-vector technique and 7.26% EER is archived using DNN-based approach. Only result for English database is used for comparison.

4. For the proposed models using 5, 8, 9, 10, 11, and 14 features, the lowest EER value of 0% and highest SI accuracy of 100% on VCTK data were achieved, while other approaches achieved a lowest EER value of 5% [40] and highest SI accuracy of 98.9% [40] with feature-level fusion, score-level fusion, and *i*-vector/GMM-UBM on the VCTK voice dataset (Table 7). For Ref. [41], best accuracy of 95.63% and EER of 4.08% is achieved using DNN method.
5. Voxforge, NIST-2008 and VCTK are medium-size voice databases; therefore, the fusion of 14 features, which is the best common model among the three, can be considered appropriate for medium-size audio datasets.
6. For the voxceleb1 dataset, the least SV EER values of 4.07% and 4.31% and 90% and SI accuracy values of 89.3% are achieved using the fusion of 14 features and 15 features, respectively, with the KNN classifier, while in Ref. [48], the best EER of 3.85% is achieved using the *x*-vector and time delay neural network (TDNN) approach. Nagrani et al. [49] achieved the best SV EER of 7.8% using a CNN architecture. A total of 1251 speakers were used in Refs. [48, 49], as in the proposed work. In Ref. [48], the total number of speaker voice samples is slightly less than that used in the proposed work; hence, the fusion of 14 features can be considered better than the results achieved by Refs. [48, 49] (Table 9).
7. The voxceleb1 dataset has the largest number of speakers among all the datasets used; therefore, the 14 and 15 feature fusion models should be considered suitable for large audio datasets.
8. From the results in Tables 5, 6, 7, 8, 9, it is observed that feature fusion with delta and delta–delta values generates better SR results than using single features.

Table 8 Best feature fusion models on the NIST-2008 audio datasets (proposed best models vs. other best models)

Method	Features used (model)	Classifier, modeling	Number of speakers/ NIST-2008 database	SI accuracy (%)	SV EER (%)
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + Δentropy + ΔLPC + entropy (11 features)	LD	120 (1200 audios)	96.9	0.2
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔRMS + ΔLPC + entropy + Δentropy (12 features)	LD	120 (1200 audios)	96.9	0.7
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	LD	120 (1200 audios)	96.9	0.3
Score-level fusion [45]	MFCC, power normalized cepstral coefficient (PNCC)	Log likelihood ratio (LLR), GMM-UBM	120 (1200 audios)	95.83	
Score-level fusion [46]	PNCC, MFCC	I vector	120 (1200 audios)	96.67	–
Score-level fusion [46]	PNCC, MFCC	GMM-UBM	120 (1200 audios)	95.83	–
DNN based [47]	PLP	Deep neural network (DNN)	Training: 300 h telephonic switch board data Testing: female short2-short3 telephonic data	–	7.26
Bottleneck features [47]	PLP	I-vector	Training: 300 h telephonic switch board data Testing: female short2-short3 telephonic data	–	5.86

The fusion of PLP, LPC, ΔPLP, ΔΔLPC, RMS, MFCC, ΔΔRMS, ΔΔPLP, ΔΔentropy, ΔLPC, entropy, Δentropy, ΔMFCC, and ΔRMS (14 features) highlighted in bold red font in the results in Tables 5, 6, 7, 8, 9 is the only model that obtains effective results for SI as well as SV on all 5 voice datasets.

9. Furthermore, when the performance of the three classifiers is compared, it is observed that the KNN classifier performs better on the ELSDSR, voxforge, and voxceleb1 databases (small, medium and large audio datasets), while the LD classifier gives better results on VCTK and NIST-2008 (medium audio datasets). Different classifiers generate different SR results for each voice dataset due to variation in the size of the training/testing datasets. This is why the final effective model with the fusion of 14 features is generated by different classifiers for each voice dataset.

Table 10 shows the summary of results obtained by the common best model of fusion of PLP, LPC, ΔPLP, ΔΔLPC, RMS, MFCC, ΔΔRMS, ΔΔPLP, ΔΔentropy, ΔLPC, entropy, Δentropy, ΔMFCC, and ΔRMS (14 features), for all databases, including training and testing data division and timing for model computation using a KNN classifier. From Table 10, it can be observed that training and testing time increase as the size of the database increases. Voxceleb1 is taking maximum training and testing timings of 2203.8 s and 43.3 s, respectively, for speaker identification, while for speaker verification, a different number of training and testing data divisions are used, hence training and testing times for SV are 2502.1 s and 52.2 s, respectively. In addition, the results generated by the proposed models are better than the other results; hence, the selection of the 2 best models at each step can be considered an effective way to produce the best SR results.

Table 9 Best feature fusion models on the voxceleb1 audio datasets (proposed best models vs. other best models)

Method	Features used (model)	Classifier, modeling	Number of speakers	SI accuracy (%)	SV EER (%)
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + ΔΔentropy + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS (14 features)	KNN	1251 Speakers, (153,516 audios)	90	4.07
Feature-level fusion [Proposed]	PLP + LPC + ΔPLP + ΔΔLPC + RMS + MFCC + ΔΔRMS + ΔΔPLP + centroid + ΔLPC + entropy + Δentropy + ΔMFCC + ΔRMS + ΔΔentropy (15 features)	KNN	1251 Speakers, (153,516 audios)	89.3	4.31
Score-level fusion [48]	MFCC, deep neural network (DNN)	x vector, attentive static pooling	1246 Speakers, (145,058 audios)	–	3.85
Score-level fusion [48]	MFCC, DNN	I vector,	1246 Speakers, (145,058 audios)	–	5.39
Automated pipelined [49]	Short time magnitude spectrogram	CNN + Embedding	1251 (153,516 audios)	–	7.8
Automated pipelined [49]	Short time magnitude spectrogram	Convolutional neural network (CNN)	1251 (153,516 audios)	80.5	–

Table 10 Summary of SR result using Fusion of PLP, LPC, PLP, ΔΔLPC, RMS, MFCC, ΔΔRMS, ΔΔPLP, ΔΔentropy, ΔLPC, entropy, Δentropy, ΔMFCC, and ΔRMS (14 features), KNN classifier (common best model) for all databases

Database	Number of speakers	Number of audios per speaker	Total number audio file	Training set	Testing set	Training time (second)	Testing time (second)	SI accuracy (%)	SV EER (%)
ELSDSR	22	9	198	154	44	1.87	0.54	100	0
Voxforge	100	10	1000	800	200	1.89	0.89	99.5	0
VCTK	109	5	545	436	109	1.86	0.83	100	0
NIST-2008	120	10	1200	720	480	1.02	0.9	96.9	0.3
Voxceleb1 (SI)	1251	Undefined	153,516	145,265	8251	2203.8	43.3	90	–
Voxceleb1 (SV)	1251	Undefined	153,516	148,642	4874	2502.1	52.2	–	4.07

Factors Influencing the SR Performance

1. The best model usually contains RMS features, so it can be predicted that adding spectral features with prosodic features, such as RMS, will improve SR performance.
2. The fusion of many features does not necessarily produce better SR results, and sometimes small feature fusion models produce better results than models with more features.
3. SR performance and training and testing timing can be affected when large datasets are used.

Conclusion and Future Work

In this paper, a new and unique feature fusion methodology is implemented to find an effective model suitable for all clean speech databases using a total of 18 features with LD, KNN and ensemble classifiers when the input is ELSDSR, VCTK, voxforge, NIST-2008 and voxceleb1 speech data. The experimental results show that the SI accuracy of the system increases to 100% and the EER value is reduced to 0% when multiple fusions of features are tested on ELSDSR, voxforge, and VCTK data. For the NIST-2008 dataset, the proposed model achieves the best SI accuracy of 96.9% with the fusion of 11, 12 and 14 features and the best EER of 0.2% with the fusion of 11 features using the LD classifier. For voxceleb1, the fusion of 14 and 15 features gave the best SI accuracy of 90% and 89.3% and SV EER values of 4.07% and 4.31%, respectively. From the experimental results, it is observed that the fusion of PLP, LPC, PLP, ΔΔLPC, RMS, MFCC, ΔΔRMS, ΔΔPLP, ΔΔentropy, ΔLPC, entropy, Δentropy, ΔMFCC, and ΔRMS (14 features) gives the best SI and SV results on all five speech datasets, from which it can be concluded that the proposed model with the fusion of 14 features is suitable for various sizes of speech datasets.

The future challenge is how to achieve faster and better ASR models. Dimension reduction techniques, such as principal component analysis (PCA) and independent component analysis (ICA), can be used to solve this problem. In addition, feature selection optimization techniques can be used in the future to reduce the computation time, and the results can be compared with the ones proposed.

Acknowledgements The proposed work is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Author Contributions Neha Chauhan performed the experimental part and calculation of the results with the help of the other authors. All the authors contributed the literature analysis, manuscript preparation, editing, proofreading and approved the final manuscript.

Funding The proposed work is based on results obtained from a project commissioned and funded by the New Energy and Industrial Technology Development Organization (NEDO).

Data Availability ELSDSR [37] is an open source database and can be downloaded from http://cogsys.compute.dtu.dk/soundshare/elsdsr.zip. Voxforge [19] is an open source database and can be downloaded from http://www.voxforge.org/home/download. CSTR VCTK [19] is an open source dataset and can be downloaded from https://datashare.ed.ac.uk/handle/10283/3443. NIST-SRE 2008 [38] can be purchased and downloaded from https://doi.org/10.35111/fyxw-v682. Voxceleb1 is an open source database and can be downloaded from https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html.

Code Availability Feature extraction code is available at https://github.com/gabenespoli/extractMIR, https://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/. Models Classification code is available at https://www.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html.

Declarations

Conflict of Interest The authors declare that they have no competing interests.

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. El-Samie FEA. Information security for automatic speaker identification. Springerbriefs in electrical and computer engineering. Berlin: Springer; 2011.
2. Barbu T. A supervised text-independent speaker recognition approach. Int J Electron Commun Eng. 2007;1:2726–30.
3. de Lara JRC. A method of automatic speaker recognition using cepstral features and vectorial quantization. In: Sanfeliu A, Cortés ML (eds) Progress in pattern recognition, image analysis and applications. CIARP 2005. Lecture notes in computer science. Berlin, Heidelberg: Springer; pp. 146–53. 2005.
4. Minh ND. An automatic speaker recognition system. Lausanne, Switzerland: Audio Visual Communications Laboratory Swiss Federal Institute of Technology. 1996.
5. Lei HH. Structured approaches to data selection for speaker recognition. In: Technical Report No. UCB/EECS. Berkeley: University of California. 2010
6. Chaudhary R. Short-term spectral feature extraction and their fusion in text independent speaker recognition: a review. BIJIT BVICAM's Int J Inf Technol. 2013;5:630–9.
7. Furui S. Comparison of speaker recognition methods using statistical features and dynamic features. IEEE Trans Acoust Speech Signal Process. 1981;29:342–50. https://doi.org/10.1109/TASSP.1981.1163605.
8. Kermorvant C, Morris A. A comparison of two strategies for ASR in additive noise: missing data and spectral subtraction. In: Proc. 6th European Conference on Speech Communication and Technology (Eurospeech 1999), 2841–2844. https://doi.org/10.21437/Eurospeech.1999-628.
9. Varga AP, Moore RK. Hidden Markov model decomposition of speech and noise. In: International conference on acoustics, speech, and signal processing. Albuquerque, NM, USA: IEEE, vol. 2, pp. 845–8. 1990.
10. Mittal U, Phamdo N. Signal/noise KLT based approach for enhancing speech degraded by colored noise. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing. (Cat. No.00CH37100). Istanbul, Turkey: IEEE; pp. 1847–50. 2000.
11. Hu Y, Loizou PC. Subjective comparison and evaluation of speech enhancement algorithms. Speech Commun. 2007;49:588–601. https://doi.org/10.1016/j.specom.2006.12.006.
12. Vaseghi SV, Milner BP. Noise compensation methods for hidden Markov model speech recognition in adverse environments. IEEE Trans Speech Audio Process. 1997;5:11–21. https://doi.org/10.1109/89.554264.
13. Boll S. Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans Acoust Speech Signal Process. 1979;27:113–20. https://doi.org/10.1109/tassp.1979.1163209.
14. Hermansky H, Morgan N. RASTA processing of speech. IEEE Trans Speech Audio Process. 1994;2:578–89. https://doi.org/10.1109/89.326616.
15. Hermansky H, Morgan N, Bayya A, Kohn P. Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTAPLP). In: Proceedings of 2nd European conference on speech communication and technology (Eurospeech 1991). Genovo, Italy; pp. 1367–70. 1991.
16. Thyme-Gobbel AE, Hutchins SE. On using prosodic cues in automatic language identification. In: Proceeding of fourth international conference on spoken language processing. Philadelphia, PA, USA: IEEE; pp. 1768–71. 1996.
17. Mary L, Yegnanarayana B. Extraction and representation of prosodic features for language and speaker recognition. Speech Commun. 2008;50:782–96. https://doi.org/10.1016/j.specom.2008.04.010.
18. Kumari TRJ, Jayanna HS. Limited data speaker verification: fusion of features. Int J Electr Comput Eng. 2017;7:3344–57. https://doi.org/10.11591/ijece.v7i6
19. Chauhan N, Isshiki T, Li D. Speaker recognition using fusion of features with feedforward artificial neural network and support vector machine. In: International conference on intelligent engineering and management (ICIEM). London, UK: IEEE; pp. 170–6. 2020.
20. Adami AG, Mihaescu R, Reynolds DA, Godfrey JJ. Modeling prosodic dynamics for speaker recognition. In: Proceedings of 2003 IEEE international conference on acoustics, speech, and signal processing. Hong Kong, China: IEEE; pp. IV–788. 2003.
21. Hossan MA, Memon S, Gregory MA. A novel approach for MFCC feature extraction. In: 4th international conference on signal processing and communication systems. Gold Coast, QLD, Australia: IEEE; pp. 1–5. 2011.
22. Peacocke RD, Graf DH. An introduction to speech and speaker recognition. Computer. 1990;23:26–33. https://doi.org/10.1109/2.56868.
23. Kumar K, Kim C, Stern RM. Delta-spectral cepstral coefficients for robust speech recognition. In: IEEE international conference on acoustics, speech and signal processing. Prague, Czech Republic: IEEE; pp. 4784–7. 2011.
24. Sönmez MK, Shriberg E, Heck LP, Weintraub M. Modeling dynamic prosodic variation for speaker verification. In: The 5th international conference on spoken language processing. Sydney, Australia: Sydney Convention Centre; pp. 3189–9192. 1998.
25. Carey MJ, Parris ES, Lloyd-Thomas H, Bennett S. Robust prosodic features for speaker identification. In: Proceeding of fourth international conference on spoken language processing. Philadelphia, PA, USA: IEEE; pp. 1800–3. 1996.
26. Chauhan N, Isshiki T, Li D. Speaker recognition using LPC, MFCC, ZCR features with ANN and SVM classifier for large input database. In: IEEE 4th international conference on computer and communication systems (ICCCS). Singapore: IEEE; pp. 130–3. 2019.
27. Lip CC, Ramli DA. Comparative study on feature, score and decision level fusion schemes for robust multibiometric systems. In: Sambath S, Zhu E, editors. Frontiers in computer education. Berlin, Heidelberg: Springer; 2012. p. 941–8.
28. Alam MJ, Kenny P, Stafylakis T. Combining amplitude and phase-based features for speaker verification with short duration utterances. In: Proceedings of the 16th annual conference of the international speech communication association. Interspeech. Dresden, Germany, pp. 249–53. 2015.
29. Li Z, He L, Zhang W, Liu J. Multi-feature combination for speaker recognition. In: 7th international symposium on Chinese spoken language processing. Tainan, Taiwan: IEEE; pp. 318–21. 2010.

30. Hosseinzadeh D, Krishnan S. Combining vocal source and MFCC features for enhanced speaker recognition performance using GMMs. In: IEEE 9th workshop on multimedia signal processing. Chania, Greece: IEEE; pp. 365–8. 2007.
31. Nakagawa S, Wang L, Ohtsuka S. Speaker identification and verification by combining MFCC and phase information. IEEE Trans Audio Speech Lang Process. 2012;20:1085–95. https://doi.org/10.1109/tasl.2011.2172422.
32. Venturini A, Zao L, Coelho R. On speech features fusion, α-integration Gaussian modeling and multi-style training for noise robust speaker classification. IEEE/ACM Trans Audio Speech Lang Process. 2014;22:1951–64. https://doi.org/10.1109/taslp.2014.2355821.
33. Elmir Y, Elberrichi Z, Adjoudj R. Score level fusion based multimodal biometric identification (fingerprint and voice). In: 6th international conference on sciences of electronics, technologies of information and telecommunications (SETIT). Sousse, Tunisia: IEEE; pp. 146–50. 2012.
34. Ali RH, Salam MA, Abed BF. Speaker identification and localization using fusion of features and score level fusion. J Theor Appl Inf Technol. 2018;96:7113–23.
35. Banerjee A, Dubey A, Menon A, Nanda S, Nandi GC. Speaker recognition using deep belief networks. 2019. arXiv:1805.08865.
36. Gupta M, Bharti SS, Agarwal S. Gender-based speaker recognition from speech signals using GMM model. Mod Phys Lett B. 2019;33:1950438. https://doi.org/10.1142/s0217984919504384.
37. Assaad FS, Serpen G. Transformation based score fusion algorithm for multi-modal biometric user authentication through ensemble classification. Procedia Comput Sci. 2015;61:410–5. https://doi.org/10.1016/j.procs.2015.09.175.
38. Dehak N, Kenny PJ, Dehak R, Dumouchel P, Ouellet P. Front-end factor analysis for speaker verification. IEEE Trans Audio Speech Lang Process. 2011;19:788–98. https://doi.org/10.1109/tasl.2010.2064307.
39. Dhakal P, Damacharla P, Javaid A, Devabhaktuni V. A near real-time automatic speaker recognition architecture for voice-based user interface. Mach Learn Knowl Extr. 2019;1:504–20. https://doi.org/10.3390/make1010031.
40. Medikonda J, Bhardwaj S, Madasu H. An information set-based robust text-independent speaker authentication. Soft Comput. 2019;24:5271–87. https://doi.org/10.1007/s00500-019-04277-9.
41. Wang J, Wang K-C, Law M, Rudzicz F, Brudno M. Centroid-based deep metric learning for speaker recognition. 2019;3652–3656. https://doi.org/10.1109/ICASSP.2019.8683393.
42. Ahmad KS, Thosar AS, Nirmal JH, Pande VS. A unique approach in text independent speaker recognition using MFCC feature sets and probabilistic neural network. In: Eighth international conference on advances in pattern recognition. Kolkata, India: IEEE; pp. 1–6. 2015.
43. Bhardwaj S, Srivastava S, Hanmandlu M, Gupta JRP. GFM-based methods for speaker identification. IEEE Trans Cybern. 2013;43:1047–58. https://doi.org/10.1109/TSMCB.2012.2223461.
44. Hannah M, Mathew M-D, Sebastein M. Towards directly modeling raw speech signal for speaker verification using CNNS. 2018. https://doi.org/10.1109/ICASSP.2018.8462165
45. Al-Kaltakchi MTDS, Woo WL, Dlay S, Chambers JA. Evaluation of a speaker identification system with and without fusion using three databases in the presence of noise and handset effects. EURASIP J Adv Signal Process. 2017;2017:1–17. https://doi.org/10.1186/s13634-017-0515-7.
46. Al-Kaltakchi MTS, Woo WL, Dlay SS, Chambers JA. Comparison of I-vector and GMM-UBM approaches to speaker identification with TIMIT and NIST 2008 databases in challenging environments. In: 25th European signal processing conference (EUSIPCO). Kos, Greece: IEEE; pp. 533–7. 2017.
47. Tian Y, Cai M, He L, Liu J. Investigation of bottleneck features and multilingual deep neural networks for speaker verification. 2015. https://doi.org/10.21437/Interspeech.2015-300.
48. Okabe K, Koshinaka T, Shinoda K. Attentive statistics pooling for deep speaker embedding. arXiv:1803.10963. 2018.
49. Nagrani A, Chung JS, Zisserman A. Voxceleb: a large-scale speaker identification dataset. 2017. arXiv:1706.08612.
50. Ross A. Fusion, feature-level. In: Li SZ, Jain A, editors. Encyclopedia of biometrics. Boston: Springer; 2009. p. 597–602.
51. Lartillot O, Toiviainen P. MIR in Matlab (II): a toolbox for musical feature extraction from audio. In: Proceedings of the 10th international conference on digital audio effects. Bordeaux, France, pp. 127–30. 2017
52. Davis S, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process. 1980;28:357–66. https://doi.org/10.1109/TASSP.1980.1163420.
53. Selvaraj M, Bhuvana R, Padmaja S. Human speech emotion recognition. Int J Eng Technol. 2016;8:311–23.
54. Saste ST, Jagdale SM. Emotion recognition from speech using MFCC and DWT for security system. In: International conference of electronics, communication and aerospace technology (ICECA). Coimbatore, India: IEEE; pp. 701–4. 2017
55. Budati AK, Valiveti H. Identify the user presence by GLRT and NP detection criteria in cognitive radio spectrum sensing. Int J Commun Syst. 2022;35:e4142. https://doi.org/10.1002/dac.4142.
56. Slifka J, Anderson TR. Speaker modification with LPC pole analysis. In: International conference on acoustics, speech, and signal processing. Detroit, MI, USA: IEEE. pp. 644–7. 1995
57. Wang L, Chen Z, Yin F. A novel hierarchical decomposition vector quantization method for high-order LPC parameters. IEEE/ACM Trans Audio Speech Lang Process. 2015;23:212–21. https://doi.org/10.1109/TASLP.2014.2380352.
58. Das A, Guha S, Singh PK, Ahmadian A, Senu N, Sarkar R. A hybrid meta-heuristic feature selection method for identification of indian spoken languages from audio signals. IEEE Access. 2020;8:181432–49. https://doi.org/10.1109/ACCESS.2020.3028241.
59. Daniel PW. PLP, RASTA, MFCC and inversion in Matlab. 2005.@misc{Ellis05-rastamat; http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/.
60. Hermansky H. Perceptual linear predictive (PLP) analysis of speech. J Acoust Soc Am. 1990;87:1738–52. https://doi.org/10.1121/1.399423.
61. Chauhan N, Chandra M. Speaker recognition and verification using artificial neural network. In: Conference on wireless communications, signal processing and networking (WiSPNET). Chennai, India: IEEE; pp. 1147–9. 2017
62. Toh A, Togneri R, Nordholm S. Spectral entropy as speech features for speech recognition. In: Proceedings of PEECS. 2005
63. Root-mean-square value. A Dictionary of Physics (6 ed.). Oxford University Press. 2009 (ISBN 9780199233991).
64. Subasi A. Machine learning techniques. In: Subasi A, editor. Practical machine learning for data analysis using python. London: Academic Press; 2020. p. 91–202.
65. https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/
66. Yao Z, Ruzzo WL. A Regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. BMC Bioinform. 2006;7:S11. https://doi.org/10.1186/1471-2105-7-S1-S11.

67. Dietterich TG. Ensemble learning. In: Arbib MA, editor. The handbook of brain theory and neural networks. Cambridge: MIT Press; 2012. p. 110–25.
68. Tin KH. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell. 1998;20:832–44. https://doi.org/10.1109/34.709601.
69. Feng L. Speaker recognition, informatics and mathematical modelling. Denmark: Technical University of Denmark; 2004.
70. NIST Multimodal Information Group. NIST speaker recognition evaluation test set LDC2011S08. Web download. Philadelphia: Linguistic Data Consortium. 2008
71. Release notes 2.4.2. Audacity Wiki. 2020; https://manual.audacityteam.org/man/new_features_in_this_release.html New features in Audacity 2.4.2.
72. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One. 2015;10:e0118432. https://doi.org/10.1371/journal.pone.0118432.
73. Tharwat A. Classification assessment methods: a detailed tutorial. Appl Comput Inform. 2020;17:168–92. https://doi.org/10.1016/j.aci.2018.08.003.
74. Furui S. Speech and speaker recognition evaluation. Dordrecht: Springer; 2007.
75. Sugrim S, Liu C, McLean M, Lindqvist J. Robust performance metrics for authentication systems. In: 26th Annual network and distributed system security symposium. San Diego, USA. pp. 1–15. https://doi.org/10.14722/ndss.2019.23351