International Journal of Mathematical, Engineering and Management Sciences
Vol. 9, No. 1, 147-162, 2024
https://doi.org/10.33889/IJMEMS.2024.9.1.008

Ram Arti Publishers

Statistically Significant Duration-Independent-based Noise-Robust Speaker Verification

Asmita Nirmal
Department of Electronics and Telecommunication Engineering,
Datta Meghe College of Engineering, Navi Mumbai, Maharashtra, India.
Corresponding author: asmita.nirmal@dmce.ac.in

Deepak Jayaswal
Department of Electronics and Telecommunication Engineering,
St. Francis Institute of Technology, Mumbai, Maharashtra, India.
E-mail: djjayaswal@sfit.ac.in

Pramod H. Kachare
Department of Electronics and Telecommunication Engineering,
Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India.
E-mail: pramod.kachare@rait.ac.in

(Received on May 1, 2023; Revised on July 11, 2023 & October 29, 2023; Accepted on November 4, 2023)

Abstract

A speaker verification system models individual speakers using different speech features to improve their robustness. However, redundant features degrade the system's performance. This paper presents Statistically Significant Duration-Independent Mel frequency Cepstral Coefficients (SSDI-MFCC) features with the Extreme Gradient Boost classifier for improving the noise-robustness of speaker models. Eight statistical descriptors are used to generate signal duration-independent features, and a statistically significant feature subset is obtained using a t-test. A redeveloped Librispeech database by adding noises from the AURORA database to simulate real-worldtest conditions for speaker verification is used for evaluation. The SSDI-MFCC is compared with Principal Component Analysis (PCA) and Genetic Algorithm (GA). The comparative results showed average equal error rate improvements by 4.93 % and 3.48 % with the SSDI-MFCC than GA-MFCC and PCA-MFCC in clean and noisy conditions, respectively. A significant reduction inverification time is observed using SSDI-MFCC than the complete feature set.

Keywords- Extreme gradient boost, Feature selection, Mel-frequency cepstral coefficients, Speaker verification.

1. Introduction

Speaker Verification (SV) is the method of authenticating the claimed identity of a person using speaker-specific information confined in the speech signal. Speech signals carry a substantial amount of information. Mel Frequency Cepstral Coefficient (MFCC) (Furui et al., 1981), Linear Prediction Cepstral Coefficient (LPCC) (Yujin et al., 2010), Perceptual Linear Prediction (PLP) (Alam et al., 2013), and vocal source representations like residual phase (Murty and Yegnanarayana, 2006) are some of the features mentioned in the literature according to the research on SV.

The most popular features for capturing speaker-specific data are MFCC features. However, MFCC features also carry redundant information that needs to be removed using feature selection (FS) methods (Jain and Zongker, 1997). Feature selection aims to identify and remove redundant and irrelevant features. It helps in achieving high system accuracy and low time complexity.

Many speech-related domains have already used feature selection, and the results have been positive

(Prasad et al., 2007; Ellis and Bilmes, 2000; Chakraborty and Saha, 2010). As shown in Table 1 variety of FS approaches are studied in the literature for SV tasks, including dynamic programming, mutual information, and information gain (Pandit and Kittler, 1998; Cohen and Zigel, 2002; Saranya et al., 2017) and different metaheuristic algorithms like the Genetic Algorithm (GA) (Raymer et al., 2000; Day and Nandi, 2006), Particle Swarm Optimization (Kennedy and Eberhart, 1995; Nemati and Basiri, 2010), Ant Colony Optimization (ACO) (Dorigo et al., 2006; Nemati et al., 2008; Arora and Vig, 2020), Crow Search Algorithm (CSA) (Askarzadeh, 2016).

Research published by Pandit and Kittler (1998), Cohen and Zigel (2002), and Saranya et al. (2017) encouraged investigation and utilizing dynamic feature selection techniques to improve the robustness of SV systems. Based on the characteristics of the input data, it adaptively selects features. Dynamic feature selection techniques dynamically modify the feature subset during the verification process instead of employing a fixed set of features.

Research by Zigel and Cohen (2004), Eriksson et al. (2005), and Ganchev et al. (2006) focuses on an information-theoretic view to find features that are highly informative for speaker discrimination. The goal of analyzing the amount of information every feature provides about the speaker's identity is to uncover highly informative features for speaker discrimination. It involves measuring the discriminatory value of characteristics using ideas from information theory, such as entropy and mutual information. It involves utilizing concepts from information theory, such as entropy and mutual information, to measure the discriminatory power of features.

Table 1. Literature review of different feature selection techniques in SV.

Sr.	Feature Selection	Pros	Cons	Reference
1.	Mutual Information	Captures relevant information and can handle continuous and discrete features.	Computationally expensive for large feature sets. Sensitive to noise.	Eriksson et al. (2005)
2.	Information gain	Robustness to Variability in speech signals.	Choosing the Recognition Related Criterion requires careful consideration and domain knowledge.	Zigel and Cohen (2004), Eriksson et al. (2005), Ganchev et al. (2006)
3.	GA, PSO, ACO, CSA algorithms	Handles large feature spaces. Finds globally optimal or near-optimal solution.	Requires algorithm parameters tuning. Computationally expensive.	Raymer et al. (2000), Day and Nandi (2006), Nemati and Basiri (2010), Nemati et al. (2008), Arora and Vig (2020)
4.	Dynamic feature switching	Adaptable to handle different recording conditions, speaker characteristics, enhanced computational efficiency, and optimized memory utilization.	Requires tuning of algorithm parameters. Performance heavily depends on the generation of diverse and discriminative feature subsets.	Pandit and Kittler (1998), Cohen and Zigel (2002), Saranya et al. (2017)
5.	PCA	Represents the data with a reduced set of features while still maintaining a large portion of the original variations in the data	Principal components are combinations of the original features, making it difficult to relate them to the underlying data attributes directly.	Zergat et al. (2012)

This paper mainly contributes to the following objectives:

- Developing a signal duration-independent speaker representation by applying statistical transformations on conventional MFCC.
- Selecting a significant feature subset using the statistical t-test and training a scalable speaker model using the XG-Boost classifier.

- Redesigning a noisy Librispeech database with various environmental noises from the AURORA database to simulate real-world noisy test conditions in SV.
- Analyzing the proposed feature subsets, conventional MFCC, and state-of-the-art PCA and GA-based selection techniques under different Signal-to-Noise Ratios (SNRs) in noisy simulated conditions.

The comparative results generally indicated that the SSDI-MFCC approach outperformed the GA-MFCC and PCA-MFCC methods in clean and noisy conditions. The SSDI-MFCC also significantly reduced verification time compared to using the complete set of features.

The remaining sections of this paper are organized as follows. Section 2 describes the theoretical background of feature extraction, transformation, selection, and classification. The steps of the proposed SV system implementation are explained in Section 3. The details of the database formation are provided in Section 4. Different experimental settings used during feature extraction, selection, and classification are described in Section 5. The results are discussed in Section 6, and the work is concluded in Section 7.

2. Theoretical Background

This section discusses a brief background on MFCC feature extraction, statistical feature transformation, and significant feature subset selection using a t-test, followed by speaker modelling using the XGBoost classifier (Chen and Guestrin, 2016; Xu et al., 2018; Parui, 2019).

2.1 MFCC Extraction

The proposed SV system uses MFCC features to characterize the spectral envelope of a vocal tract with a focus on the source filter modelling approach of speech signals. At first, the speech signal is pre-emphasized to increase theamplitude of high frequencies that are ignored during speech production. The emphasized speech signal is segmented into short frames to obtain time-invariant acoustic characteristics. A 20–30 ms frame maintains good spectraland temporal resolution. Then, a Hamming window is applied to minimize spectral leakage by narrowing the speech frames at the boundaries. Also, considering the dynamic characteristics of different speech frames, 50–75 % overlapis kept in consecutive frames. A magnitude spectrum is calculated for each frame using the Fourier Transform. A filter bank with 40 triangular bandpass filters spaced uniformly on the mel scale is applied to the frame spectrum. The physical frequency (*f*) in Hz can be converted using Equation (1) to a Mel frequency (*mel*) as perceived by the human ear.

$$mel = 2595\ \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

The logarithmic function is applied to the filter-bank outputs for dynamic range compression. The Discrete Cosine Transform (DCT) is a fundamental MFCC feature extraction process component. It serves two important purposes: compressing the energy of the speech signal by concentrating it into fewer coefficients and de-correlating the coefficients to reduce redundancy. Application of DCT to the mel filtered outputs, as shown in Equation (2), allows MFCCs to effectively capture the necessary spectrum information for SV tasks, enabling robust and discriminative representations of speech characteristics. The DCT's ability to compactly represent the spectral content of the speech signal makes it a key factor in the success of MFCC-based SV systems. The maximum of the speech information is contained in lower-order DCT coefficients. For the *n*th frame, MFCC is calculated as,

$$mc(n,k) = \sum_{i=1}^{N} \log_{10} s(n,i)\ .\cos\left(\frac{\pi k(i-0.5)}{N}\right) \tag{2}$$

where, $n \in [1, N_F]$, and $k \in [1, N_M]$, $mc(n, k)$ is the *k*th cepstral coefficient, N_F is the number of frames, N_M is the number of MFCCs, $s(n, i)$ is the *i*th output of the mel-filterbank, and N is the number of filters in the mel-filterbank. Thus, the MFCC feature extraction essentially includes windowing, transformation, applying the mel-filterbank, andlogarithm operation followed by DCT.

As reported in the literature, 13 DCT coefficients are enough to represent vocaltract shape faithfully. The dynamic characteristics of MFCC are also reported to be essential for capturing speaker-specific information. The temporal derivative of MFCC with a window of two frames on either side of the current frame has reported a good correlation to speaker-specific information and is calculated as in Equation (3).

$$\Delta mc(n,k) = \sum_{t=1}^{2} 0.5 \times t(mc(n+t,k) - mc(n-t,k)) / \sum_{t=1}^{2} t^2 \tag{3}$$

A second-order temporal derivative using a similar mathematical construct and input replaced by a first-order derivative is calculated to represent high-order feature dynamics. A feature matrix F is generated by concatenating the MFCC matrix (MC) and its first derivative (ΔMC) and second derivative ($\Delta\Delta MC$), as shown in Equation (4). It represents the spectral characteristic of each utterance.

$$F = [MC \mid \Delta MC \mid \Delta\Delta MC] \tag{4}$$

2.2 Duration-Independent Feature Transformation

The extracted feature matrix varies in size according to the signal's duration. Hence, it cannot be applied to build machine learning-based speaker models. Appending or truncating the signal to a fixed dimension could solve this issue, but these modifications may change the signal characteristics. Thus, the SV system's performance will depend on signal duration. Also, empirically fixed signal duration may not result in robust speaker models. This work computes eight statistical descriptors (Ayyub and McCuen, 2016) for each utterance's feature matrix to generate duration-independent (DI) features. The descriptors comprise minimum, maximum, mean, median, variance, kurtosis, skewness, and interquartile range. A summary of statistical descriptors used in the proposed SV system is as follows:

- **Mean** is the sum of all the data points divided by the total number of data points. As shown in Equation (5), the mean of the *i*th coefficient is obtained as,

$$Men(k) = \frac{1}{N_F} \sum_{n=1}^{N_F} F(n,k) \tag{5}$$

where, $k \in [1, N_T]$, N_F is the number of frames or rows of the feature matrix, and N_T is the total number of features, i.e., three times N_M.

- ***Median*** is the middle value of a set of data containing an odd number of observations, whereas it is the averageof the two middle observations of a set of data containing an even number of observations. The *k*th column of the feature matrix should be ordered in ascending order, considering all N_F frames before calculating the median, as shown in Equation (6):

$$Med(k) = \begin{cases} F\left(\frac{N_F}{2}, k\right) & \text{if } N_F \text{ is odd} \\ \frac{1}{2}\left(F\left(\frac{N_F}{2}, k\right) + F\left(\frac{N_F}{2} + 1, k\right)\right) & \text{if } N_F \text{ is even} \end{cases} \tag{6}$$

- **Variance** measures how data varies in its mean value. It is the ratio of the sum of squared differences between observations and their means and the total number of observations. The variance of the *k*th feature is computed below in Equation (7).

$$Var(k) = \frac{1}{N_F} \sum_{n=1}^{N_F} \left(F(n,k) - Men(k)\right)^2 \tag{7}$$

- **Skewness** measures the asymmetry of the input data around the sample mean. As shown in Equation (8), it is calculated as,

$$Ske(k) = \sum_{n=1}^{N_F}\left(F(n,k) - Men(k)\right)^3 / (N_F - 1)\,.\,Var(k)^{1.5} \tag{8}$$

- **Kurtosis** describes whether the distribution is heavy-tailed or light-tailed. Therefore, a sharply peaked distribution has low kurtosis, and the distribution with a lower peak has high kurtosis. As shown in Equation (9), it is calculated as,

$$Kur(k) = \sum_{n=1}^{N_F}\left(F(n,k) - Men(k)\right)^4 / Var(k)^2 \tag{9}$$

- **Inter-Quartile Range (IQR)** measures the spread of feature values. Quartiles arrange feature values in ascending order in equal parts. It includes features from the second and third quartiles, which includes the middle half of feature values and is computed as in Equation (10).

$$IQR(k) = F\left(\left\lfloor\frac{3(N_F+1)}{4}\right\rfloor, k\right) - F\left(\left\lfloor\frac{(N_F+1)}{4}\right\rfloor, k\right) \tag{10}$$

IQR depends on the middle half of the data, so outliers do not affect it.

2.3 Feature Selection Using Statistical Significance

The easiest way of feature selection is to check the model's performance for all possible combinations of features and select the feature subset that causes the best model performance. However, it would be an inefficient way of doing FS. Therefore, each feature is tested independently using a statistical t-test to decide if a significant difference exists between the mean of the two groups. More specifically, the aim of a statistical t-test while selecting a feature is to decide whether it is following the null hypothesis or alternate hypothesis. In our proposed work, the features are from two groups of speech utterances: the target group and the imposter group. Therefore, the hypothesis will be: Null Hypothesis: There is no significant difference between the mean of features representing the target and imposter groups. Here, the alternative hypothesis signifies that the imposter and target groups represented by a feature vector are significantly different. Firstly, the *t*-value is calculated as shown in Equation (11) to select salient features. The output of the t-test is to be compared with the *p*-value. If the *t*-value is larger than the p-value, the null hypothesis will be accepted; else, reject it.

$$t(k) = \left(m_{ipt}(k) - m_{tgt}(k)\right)\left(\frac{SD_{ipt}(k)^2}{N_{ipt}} + \frac{SD_{tgt}(k)^2}{N_{tgt}}\right)^{-1/2} \qquad k \in [1, N_T] \tag{11}$$

where, (m_{ipt} , SD_{ipt} , N_{ipt}) and (m_{tgt} , SD_{tgt} , N_{tgt}) are the mean, standard deviation, and total examples of imposter (*ipt*) and target (*tgt*) features, respectively. In this work, the number of imposters and target utterances is assumed to be equal to avoid imbalance class problems. If the t-test results in a high p-value, the difference between the two groups' means is insignificant. Features with low *p*-values are retained, and features with a *p*-value greater than 0.05 are discarded.

2.4 Speaker Modeling using XGBoost

The main objective of the Machine Learning (ML) algorithm is to find the function that maps input features to the output class. In this work, the mapping function represents the relationship between the input feature vector extracted from a speech utterance and the type of utterance class, which can be either a target class or an imposter class. The word boosting in the XGBoost algorithm signifies that in the boosting category of the ensemble learning algorithm. Instead of training models together, XGBoost trains decision trees representing different training models one after the other. After each iteration, the algorithm mainly focuses on training examples that are wrongly classified. Each iteration obtains a new model to classify the wrongly classified examples to the correct class. For doing so, afterevery iteration, the correctly classified outputs are given a lower weight than the ones that were misclassified inthe previous iteration; with this process, after each iteration, residuals keep on decreasing, which in turn optimizes the loss function, as shown in Equation (12). The training process starts with an initial assumption of prediction and a loss function. XGBoost uses the following loss function:

$$\mathcal{L} = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{T} \Omega(f_k) \tag{12}$$

The loss function keeps on checking whether the prediction is correct or not. The term $l(\hat{y}_i, y_i)$ is residuals of *i*th training example amongst *n* training examples. Residues are calculated using the actual value y_i and predicted value $\hat{y}_i$. The second term is the complexity of the tree $\Omega(f)$, computed as shown in Equation (13).

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{13}$$

The tree $f(x)$ and score, w on its corresponding leaf, are related, as shown below in Equation (14).

$$f(x) = w_{q(x)}, \qquad q: R^d \rightarrow \{1, 2, \ldots\ldots, T\} \tag{14}$$

where, $w \in R^T$, $x \in R^d$, q is a function that allocates every input feature vector x to the corresponding *j*th leaf, T is the total number of leaves of tree f, and w_j is the output score at the *j*th leaf. The parameter λ is related to regularization thathelps to avoid tree over-fitting. It should be chosen carefully as the high λ value lowers the similarity score, resultingin in lower gain, which will cause more tree pruning. The parameter γ controls tree pruning and is set in the initialstep. If the gain is higher than γ, further tree splitting will occur; otherwise, it will not. Thus, a high γ value causes more tree pruning. The goal is to find an optimized output value for the leaf to minimize the loss function. The aim is to set the model's hyperparameters and use the feature vectors capturing speaker-specific information to perform averification task using the XGBoost algorithm for a binary classification task.

3. Proposed Methodology

This section details all the stages used in the proposed SV system. A framework shown in Figure 1 illustrates the various steps involved in the SV system. First, speaker utterances are collected from the redevelopednoisy Librispeech database. Then, after going through pre-processing stages such as framing and windowing, MFCCfeatures are extracted from all the collected speaker files. Eight different feature transformations are applied to convertvariable-size MFCC feature matrices to fixed-size DI features for each speaker utterance. Then, a statistical test-based feature selection strategy is applied to reduce the dimensionality and remove the irrelevant features. Then, the final feature sets obtained through FS are labeled as target and imposter sets and applied as input to an XGBoost classifier to form a speaker-specific model. Lastly, the performance of trained speaker models is checked for a different subset of features. Furthermore, speaker models are trained under different noisy conditions to check their robustness.

The main problem is the variable size of the MFCC feature matrix F due to a variable number of frames. The structure of the matrix is shown in Figure 2. It can be seen that the rows of the original cepstral coefficient matrix F indicate frames of a speech utterance, and columns indicate cepstral coefficients, including delta and delta-delta coefficients.

Algorithm 1 shows the steps to convert the original cepstral feature matrix F to a DI feature vector FT. Here, variables N_F and N_T represent the number of rows and columns of matrix F. As explained in Algorithm 1, total N_S statistical descriptors are derived for each matrix column to make the features invariant to the length of the speech utterance. The 'feature transform' function returns transformed coefficients represented by the vector FT. The framework of FT is shown in Figure 2. The output FT vector has a size ($1 \times N_{FT}$), where $N_{FT} = N_F \times N_T = N_F \times 3 \times N_M$. Hence, the size of the transformed vector depends on the number of MFCC coefficients and the number of statisticaldescriptors computed from each cepstral representation.

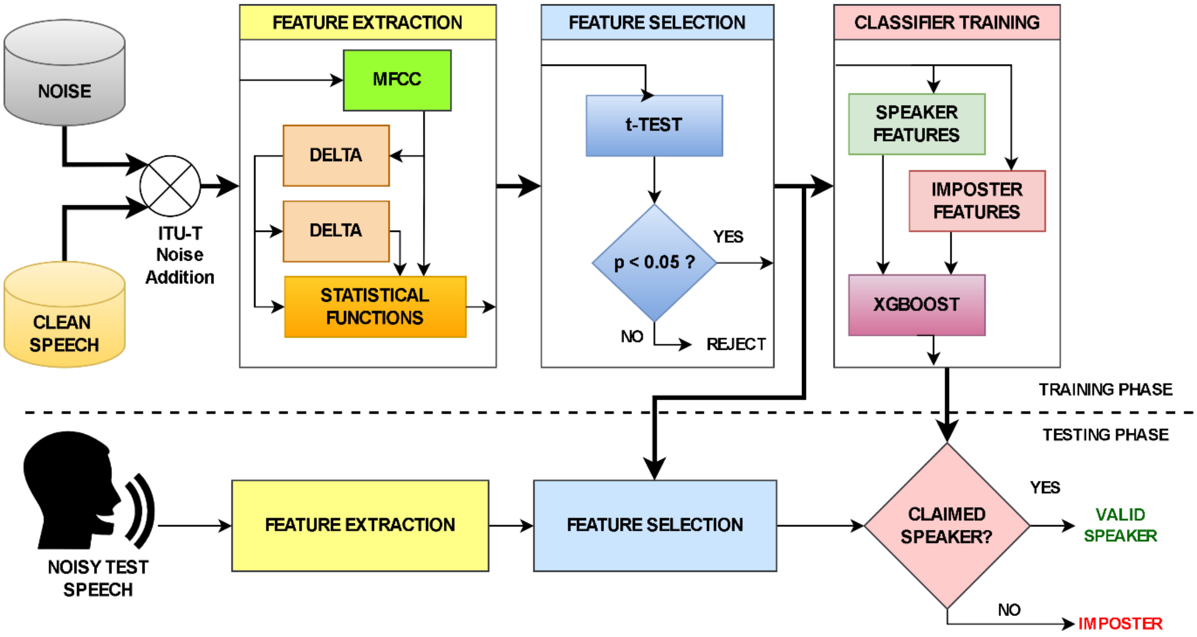

Figure 1. Proposed speaker verification system using statically significant duration-independent MFCC features and XGBoost classifier.

FT is computed for all the speaker utterances of clean and noisy databases. After calculating the DI feature matrix for all utterances, the next task is to select statistically significant (SS) features. A structure of target and imposter files is formed for selecting significant features, as shown in Figure 3. Here each element $FT^{spkr}(i, j)$ the matrix represents the j^{th} transformed feature of i^{th} speaker utterance where $spkr \in \{tgt, ipt\}$.

Algorithm 1. Steps for duration-independent feature transformation.

```
Input:
$F$ = Original cepstral coefficient feature matrix,
$N_F$ = Number of rows of $F$,
$N_T$ = Number of columns of $F$,
$N_S$ = Number of statistical features

Output:
$FT$ = Duration-independent feature vector

Procedure:
feature_ transform ($F, N_F, N_T, N_S$)
    for $k$ = 1 to $N_T$ do
        $FT(k, 1)$ = minimum($F(:, k)$)
        $FT(2N_T+k, 1)$ = maximum($F(:, k)$)
        $FT(3N_T+k, 1)$ = mean($F(:, k)$)
        $FT(4N_T+k, 1)$ = median($F(:, k)$)
        $FT(5N_T+k, 1)$ = variance($F(:, k)$)
        $FT(6N_T+k, 1)$ = skewness($F(:, k)$)
        $FT(7N_T+k, 1)$ = kurtosis($F(:, k)$)
        $FT(N_{stats}*N_T+k, 1)$ = iqr($F(:, k)$)
    endfor
```

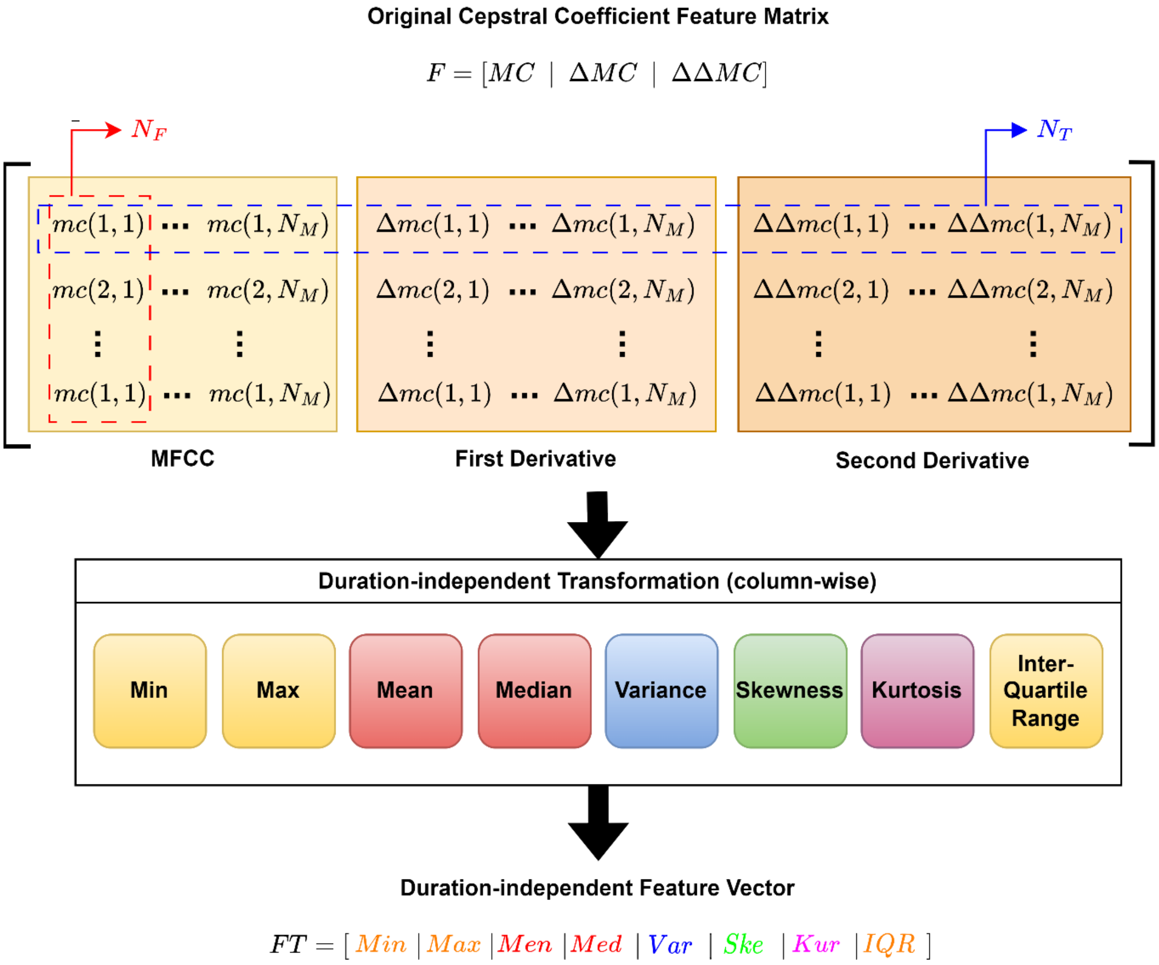
Original Cepstral Coefficient Feature Matrix
$F = [MC \mid \Delta MC \mid \Delta\Delta MC]$
N_F
N_T
$mc(1,1)$ ⋯ $mc(1,N_M)$
$mc(2,1)$ ⋯ $mc(2,N_M)$
$mc(1,1)$ ⋯ $mc(1,N_M)$
$\Delta mc(1,1)$ ⋯ $\Delta mc(1,N_M)$
$\Delta mc(2,1)$ ⋯ $\Delta mc(2,N_M)$
$\Delta mc(1,1)$ ⋯ $\Delta mc(1,N_M)$
$\Delta\Delta mc(1,1)$ ⋯ $\Delta\Delta mc(1,N_M)$
$\Delta\Delta mc(2,1)$ ⋯ $\Delta\Delta mc(2,N_M)$
$\Delta\Delta mc(1,1)$ ⋯ $\Delta\Delta mc(1,N_M)$
MFCC
First Derivative
Second Derivative
Duration-independent Transformation (column-wise)
Min
Max
Mean
Median
Variance
Skewness
Kurtosis
Inter-Quartile Range
Duration-independent Feature Vector
$FT = [\,Min \mid Max \mid Men \mid Med \mid Var \mid Ske \mid Kur \mid IQR\,]$

Figure 2. Transformation of variable-sized MFCC matrix to fixed-size DI feature vector.

Each row of the matrix shown in Figure 3 belongs to a transformed feature vector of target or imposter utterances obtained using the procedure shown in Figure 2. The set of utterances labelled as $FT^{tgt}(i,j)$ and $FT^{ipt}(i,j)$ indicates the j^{th} DI feature of i^{th} target and imposter utterances. Both sets should include an equal number of utterances to avoid class imbalance. The first column from both DI feature matrices is selected at the start. The statistical t-test is performed, as discussed in Section 2.3. If the *p*-value of the current feature is less than 0.05, then retain the column else, remove the feature from the subset. Repeat the process for all remaining features in *FT*. Ultimately, the subset comprising the retained features is returned as Statistically Significant DI (SSDI) features. It must be noted that the current work applied the SSDI procedure on the MFCC matrix and is hence called SSDI-MFCC, but it can be easily adapted for any short-time feature representation of the speech utterance.

4. Database Formation

Choosing a suitable dataset is one of the critical tasks while developing an SV system. The speaker modelsare commonly formed under clean conditions. Merely using clean speech utterances will not give any idea aboutthe robustness capability of our system in noisy conditions. Therefore, a noisy version of the

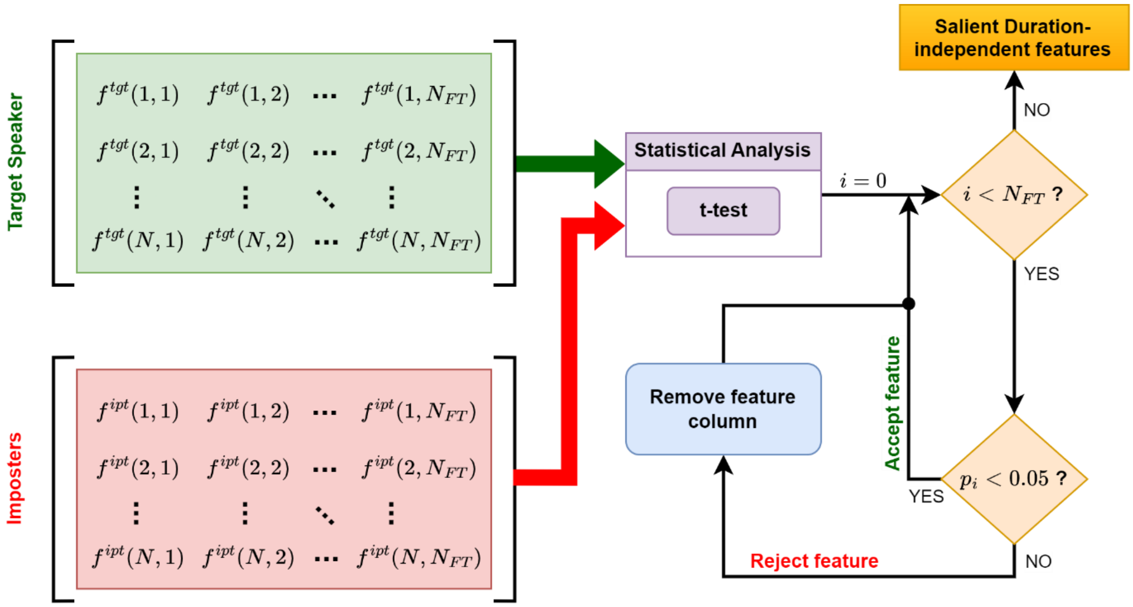
Target Speaker
$f^{tgt}(1,1)$ $f^{tgt}(1,2)$ ⋯ $f^{tgt}(1,N_{FT})$
$f^{tgt}(2,1)$ $f^{tgt}(2,2)$ ⋯ $f^{tgt}(2,N_{FT})$
$f^{tgt}(N,1)$ $f^{tgt}(N,2)$ ⋯ $f^{tgt}(N,N_{FT})$
Imposters
$f^{ipt}(1,1)$ $f^{ipt}(1,2)$ ⋯ $f^{ipt}(1,N_{FT})$
$f^{ipt}(2,1)$ $f^{ipt}(2,2)$ ⋯ $f^{ipt}(2,N_{FT})$
$f^{ipt}(N,1)$ $f^{ipt}(N,2)$ ⋯ $f^{ipt}(N,N_{FT})$
Statistical Analysis
t-test
$i = 0$
$i < N_{FT}$?
NO
Salient Duration-independent features
YES
$p_i < 0.05$?
YES
Accept feature
NO
Reject feature
Remove feature column

Figure 3. Selection of statistically significant DI feature using statistical t-test.

original clean speech recordings from the original Librispeech database (Panayotov et al., 2015) are developed.

This work uses the Librispeech database comprising 125 female and 126 male English sentence recordings, each of around 25 minutes and sampled at 16 kHz. A subset of 100 hours duration with balanced female and male speakers is selected for all experiments. However, there can be noisy conditions during the actual deployment of the SV system in real-life applications. Various noises from the AURORA database (Hirsch and Pearce, 2000) are added to original clean utterances to make the system robust and deployable in a real-life environment.

All the possible noise conditions during real-life deployment cannot be simulated. Hence, eight different noiseconditions from the AURORA database are used as references. The available noise conditions are train, babble, car, restaurant, street, subway, airport, and exhibition hall. Each noise recording is 10 seconds and sampled at 8 kHz. The noise samples are up-sampled by a factor of two, and the International Telecommunication Union Telecommunication algorithm (ITU, 2011) is used to add noise to the clean speech utterances. The SNR is the ratio of the power of active speechlevel in a clean signal and the power of the noise signal. The active speech level estimation uses an envelope value atevery sampling instant and compares it with a set of threshold values. A noisy speech signal of the desired SNR level is computed as,

$$y(n) = x(n) + \alpha \, . \, d(n) \tag{15}$$

where, $x(n)$ and $d(n)$ are the original clean speech signal and noise speech signal, respectively. The scale factor α in Equation (15) controls the amount of the noise signal to be added to the clean speech to obtain a noisy speech signal of the desired SNR level. The power of the clean signal Equation (16) and that of the noise signal Equation (17) is computed as follows (Oppenheim, 1999):

$$P_x = \sum_{n=1}^{N_x} x(n)^2 / N_x \tag{16}$$

$$P_d = \sum_{n=1}^{N_x} d(n)^2 / N_x \tag{17}$$

The scale factor α is computed as shown below in Equation (18),

$$\alpha = \sqrt{\frac{1}{SNR_y} \cdot \frac{P_x}{P_d}} \tag{18}$$

where SNR_y is the desired SNR level of noisy signal $y(n)$.

5. Experimental Settings

As the proposed methodology explains, various steps are carried out during our system implementation. It includes pre-processing, feature extraction and transformation, and feature selection, followed by the formation of classifiers. During our experimentations, performance measures such as accuracy and Area Under Curve (AUC), precision, recall, F1-score, and an Equal Error Rate (EER) and Detection Error Trade-off (DET) curve (Martin et al., 1997) are used to evaluate our SV system.

5.1 Feature Selection Settings

In the pre-processing step, the pre-emphasized speech utterance is segmented into short frames of 20 ms with10 ms overlap. A magnitude spectrum of Hamming windowed frames is calculated using the Fast Fourier Transform (FFT). Finally, the DCT of the log of magnitude spectrum is computed, and the first 13 coefficients are used as MFCC features. A 39-dimensional feature vector for each speech frame is obtained by concatenating 13 MFCCs, 13 deltacoefficients, and 13 delta-delta coefficients. Next, eight different statistical descriptors: minimum (Min), maximum (Max), median (Med), mean (Mea), variance (Var), kurtosis (Kur), skewness (Ske), and interquartile range (IQR) are applied to transform the feature matrix F of variable size into a fixed-size DI feature vector of length 312 (39 ×8), as in Table 1 and Figure 2.

The relative importance of 312 features using the FS method described in Section 2.3 is investigated. Two subsets of features are investigated. The first subset selects a feature if it is found significant at all SNR levels (AND strategy). The second subset selects a feature if it is found significant at any SNR level (OR strategy). The primary motivation behind these two strategies is that the AND strategy is more restrictive regarding the numberof selected features, while the OR strategy is all-inclusive with a more significant number of features. For low-resource application implementation, the AND strategy is the best, with a considerable reduction in performance, while if memory is nota concern, the OR strategy can be implemented. A more fortunate situation will be when performance degradation using the AND strategy is not high. Hence, an SV system with few features and high performance can be implemented.

Table 2 describes the SSDI-MFCC features selected using both strategies. An entry 'M' indicates MFCC, 'V' indicates delta coefficients, and 'A' indicates delta-delta coefficients. The subset using the AND strategy $SSDI_{AND}$ selects only 19 features, emphasized with bold and italics: the minimum of the 1st, 3rd, 4th, 5th, 6th, 7th, 9th, 10th, and 11th MFCC, the median of 1st, 4th,5th, 6th, 7th, 8th, and 10th MFCC and standard deviation of 9th MFCC. The subset using OR strategy $SSDI_{OR}$ selects 154 features as des described in Table 2. It contains all the features with p-values less than 0.05 for at least one of the SNR levels. A feature with a high p-valuefor one SNR level might have a low p-value for some other SNR level carrying a different level of speaker information. Hence, it must be noted that $SSDI_{OR}$ is a proper subset of $SSDI_{AND}$.

The performance of proposed SSDI features is compared with GA and PCA-based FS approaches on the Librispeech database. GA uses a population comprising 30 chromosomes, each of size 312, and each chromosome is encoded as a binary bit string. The best chromosome to reproduce offspring out of 30 chromosomes is selected using a fitness function. The fitness of a chromosome is checked using the state-of-the-art 5-nearest neighbour classifier.A uniform crossover is used where every bit is selected from one or the other parent with equal probability. A few bits in the candidate solution are flipped using mutation to maintain diversity among the population and avoid early convergence. This process is done until there is no progress in the fitness value after going through 100 iterations. GA-MFCC reduces 312 features to a feature subset of 152 features.

Table 2. SSDI-MFCC features for speaker verification using AND -OR strategies.

	Min	**Max**	**Men**	**Med**	**Var**	**Kur**	**Ske**	**IQR**	**# Coefs.**
mc-0	M,V,A	M,V,A	M	M	M,V,A		M	M,A	7M+3V+4A
mc-1	***M***,V,A	M,V,A	***M***	***M***	M,V,A	M	M	M,V,A	8M+4V+4A
mc-2	M,V,A	M,V,A	M	M	M,V,A	M	M	M,V,A	8M+4V+4A
mc-3	***M***,V,A	M	***M***	M	M,V,A		M	M,V,A	7M+3V+3A
mc-4	M,A	M	***M***	***M***	M,V,A	M	M	M,V,A	8M+2V+3A
mc-5	M	M	***M***	***M***	M,V,A			M,V,A	6M+2V+2A
mc-6	M	M	***M***	***M***	M,V,A	M,		M	7M+1V+1A
mc-7	M,V	M,A	***M***	***M***	M,V,A		M		6M+2V+2A
mc-8	M	M,V	M	***M***	M,V		M	M	7M+2V+0A
mc-9	M,A	M,V,A	***M***	M	M,V,A		M	M,V,A	7M+3V+4A
mc-10	M,A	M	***M***	***M***	M,V,A			M,V	6M+2V+2A
mc-11	M	M	***M***	M	M,V,A			M,A	6M+1V+2A
mc-12	M	M,A	M	M	M,V,A			M,V,A	6M+2V+3A
#Desc.	*25*	*24*	*13*	*13*	*38*	*4*	*8*	*29*	*154*

Furthermore, the PCA-based feature selection is used for comparative analysis. The aim of using PCA as a tool for feature selection is to de-correlate features. In a nutshell, PCA finds the direction of maximum variance in high-dimensional data and projects it onto a subspace with fewer dimensions than the original one. The relation between Eigen space's dimension and the Eigen information percentage is used to select the optimum dimension. PCA-MFCC reduces the dimension from 312 to 195 (maintains 95% variance).

5.2 Classifier Settings

The performance of the proposed SSDI features is evaluated using the XGBoost classifier. While forming a speaker model, feature files are grouped into target and imposter classes. The imposters for a speaker are randomly chosen from the speaker's utterances other than the target speaker's utterances. In addition, care is taken that the number of target imposter utterances is the same to avoid class imbalance issues. The data is divided into 70-30 % as training and testing. Four hyper-parameters are optimized. The learning rate is set to 0.01, controls the step size to allow the feature weights to follow the boosting process, and improves generalization. The sub-sample is set to 0.7. It randomly selects 70 % of training data for each new tree and avoids over-fitting. The number of boosted trees is set to 1200 and the maximum depth of trees is set at 3.

6. Results and Discussions

While investigating the use of different feature subsets as input to a speaker model, getting its output is insufficient to evaluate the verification system performance. The model's accuracy, precision, recall, and DET are evaluation metrics.

The performance of the XGBoost classifier is first compared using a complete feature set FT_{MFCC}, AND strategy $SSDI_{AND}$, and OR strategy $SSDI_{OR}$, as shown in Figure 4. Different feature subsets at different SNR levels are compared to different performance measures. As expected, it can be seen in Figure 4 that EER increases as noise conditions worsen. Thus, the system with a low EER value will have a low False.

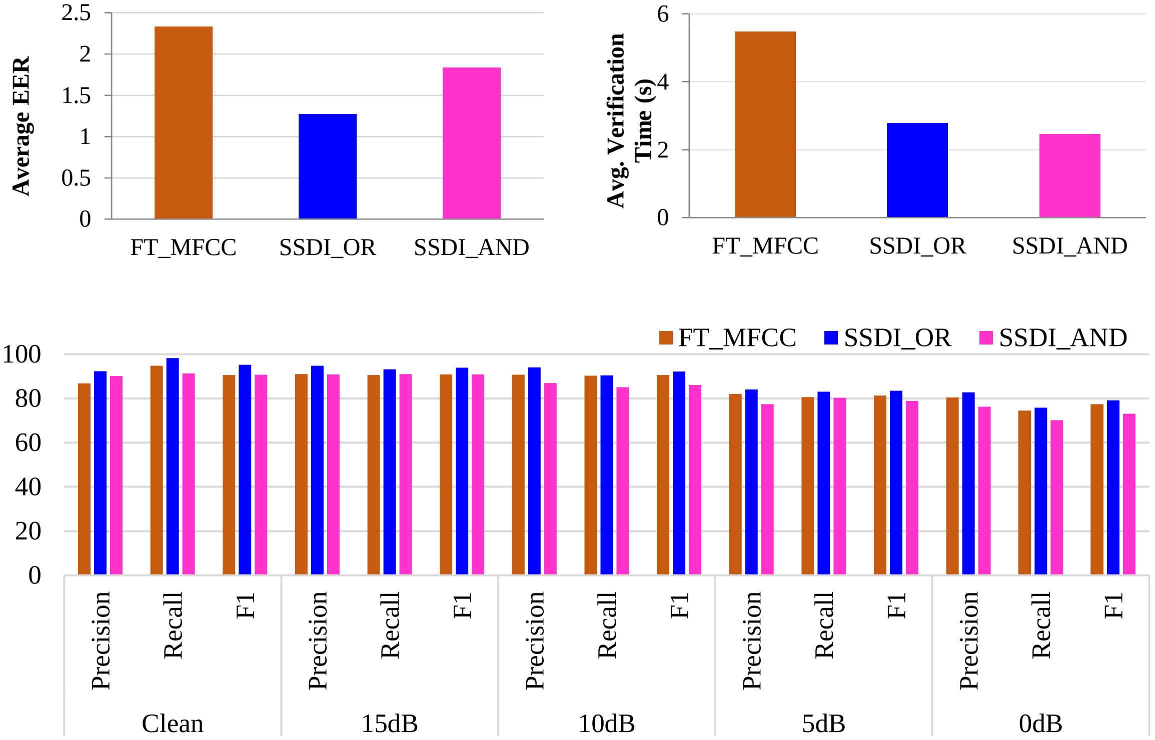

Figure 4. Effects of different subsets on SV system performance EER, verification time, and precision, recall, and F1.

Acceptance Rate (FAR) and False Rejection Rate (FRR). Besides system verification time, precision, recall, and F1-score also degrade as SNR degrades from clean to 0 dB. However, the EER values obtained from $SSDI_{OR}$ feature subsets are improved by 1.057 % and 0.56 % compared to FT_{MFCC} and $SSDI_{AND}$ feature subsets, respectively. It means $SSDI_{AND}$ does not guarantee that the features that do not capture information at one SNR level also do not capture the information at other SNR levels. Therefore, it may discard relevant features unnecessarily. The $SSDI_{OR}$ provides the best performance, followed by $SSDI_{AND}$ and FT_{MFCC} feature subsets.

This paper also aims to check whether the features selected under two settings would perform satisfactorily with minimum verification time. It can be observed in Figure 4 that the feature subset $SSDI_{AND}$ not only provides average EER improvement over the entire set FT_{MFCC} but also reduces the verification time by 49.14 %, which is 11.83 % higher than $SSDI_{OR}$.

Moreover, to confirm the consistency of the results, other performance measures are also computed during experimentation, such as precision, recall, and F1-score. Precision to check the frequency with which our model correctly predicts the target speaker and imposter. Recall to specify out of all target speakers how many speakers are correctly classified by the speaker model and the F1-score is computed as a weighted average of precision and recall. Figure 4 depicts precision, recall, and F1-score for SNR levels from 15–0 dB. The values are obtained by averaging the performance measures of individual speakers across different noise types. It is seen that although performance degrades in noisy conditions for all competing, $SSDI_{OR}$ performs better than other methods.

Currently, the SV system is not implemented using low-resource hardware. Hence, $SSDI_{OR}$ is selected as the final subset SSDI-MFCC for comparison with state-of-the-art systems. The EER values in Table 3 demonstrate that the proposed SSDI-MFCC method outperformed GA-MFCC and PCA-MFCC when dealing with clean and noisy conditions. The SSDI-MFCC with 154 features has the best EER of 3.41 % in clean speech conditions, degrading to 16.83 % in 0 dB SNR. In GA-MFCC, the dimension is reduced to 150, but EER is degraded to 7.62 % in a clean environment. However, the proposed SSDI-MFCC resulted in a lower EER using 154 features, slightly more than GA-MFCC. The GA-MFCC performed second best at high SNR compared to the proposed SSDI-MFCC at low SNR. The PCA-MFCC with 195 features performs worst among the three FS methods. At 10 dB SNR, the feature subsets obtained from PCA-MFCC resulted in better performance than GA-MFCC, possibly due to some inherent noise compensation in linear transformation. However, using linear transformation in PCA is more costly as it usually increases the dimensionality of the selected feature subset.

Table 3. Comparative EER (%) analysis for different FS methods in real-world noises at different SNR levels.

Method	Clean	15 dB	10 dB	5 dB	0 dB
SSDI-MFCC	3.41	4.62	9.06	11.94	16.83
GA-MFCC	7.62	8.15	9.56	12.62	16.81
PCA-MFCC	9.06	10.31	9.53	22.34	23.41

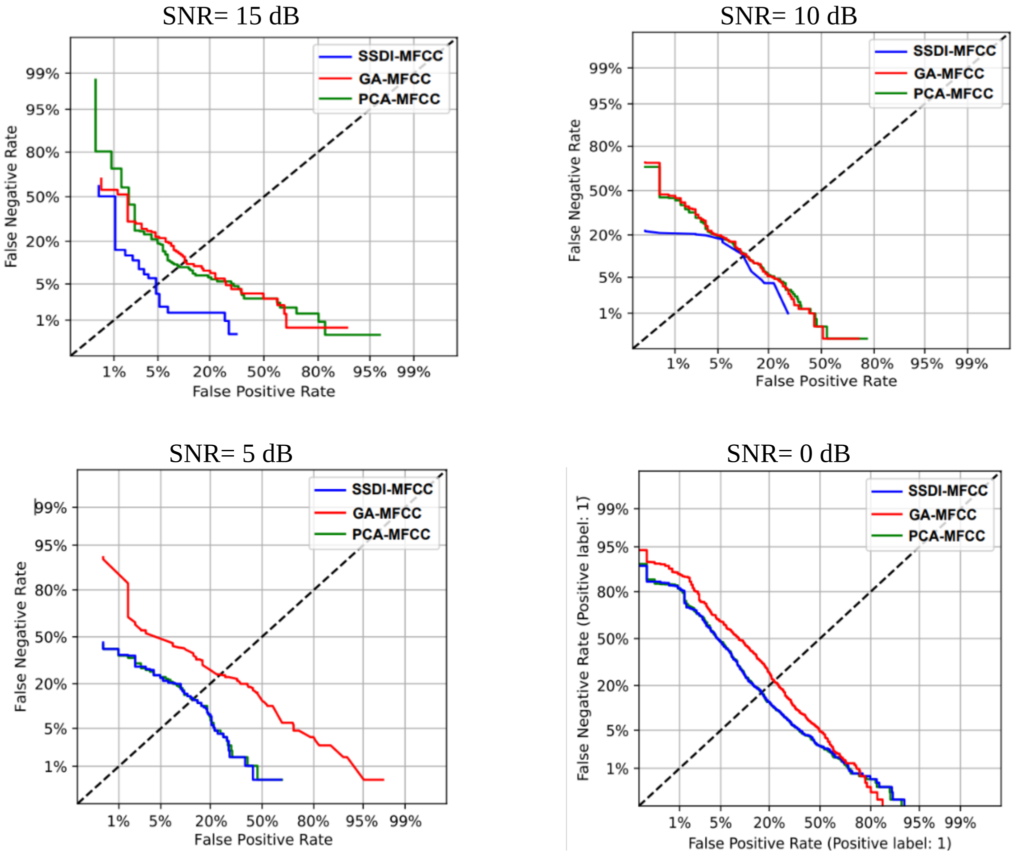

Figure 5. Comparative DET curve analysis for SV system using SSDI-MFCC, GA-MFCC, and PCA-MFCC at different SNR levels.

The SV system may make errors in detecting some of the target or imposter speaker test utterances. Using FAR and FRR pairs of different target speakers to plot the DET curve would be inappropriate for the system. The SV system may make errors in detecting some of the target or imposter speaker test utterances. Using FAR and FRR pairs of different target speakers to plot the DET curve would be inappropriate as the system includes many targets. Therefore, DET curves for individual target speakers are averaged across different speakers and noise types instead of combining the error rates, as shown in Figure 5. As anticipated, the noticeable trend is that as the signal-to-noise ratio (SNR) decreases, the DET curve steadily ascends diagonally. Notably, this progression is particularly evident in the DET curves corresponding to 5 dB and 0 dB SNR levels. These curves shift upward, underscoring a notable escalation in the FPR and FNR. The underlying cause for this shift is the heightened prominence of noise relative to the diminishing signal quality, a consequence of the reduced SNR compared to the 15 dB scenario.

In instances characterized by either 0 dB or 5 dB noise conditions, the performance of the GA-MFCC feature extraction method is notably suboptimal. The observed decline in its effectiveness can be attributed to the elevated noise levels, potentially causing the search agents within the genetic algorithm to become trapped in local minima. This indicates the challenges posed by increased noise in converging toward optimal solutions. An alternative approach that holds promise for achieving robust performance in the presence of noise involves using SSDI-MFCC features coupled with the XGBoost classifier. This combination appears to counteract the adverse effects of noise, enhancing noise tolerance and classification accuracy. The fusion of SSDI-MFCC features, adept at preserving essential information amidst noise, with the adaptable learning capabilities of XGBoost presents a viable strategy for cultivating noise-robust performance.

7. Conclusion

This paper develops a statistically significant duration-independent MFCC feature subset to improve the speaker verification system in real-world noisy conditions. MFCC and its first and second derivatives were transformed using eight statistical descriptors to generate duration-independent features. Two subsets of statistically significant features were selected based on a *t*-test using AND (significant in all cases) and OR (significant in at least one case) strategies. A classifier based on the XGBoost algorithm was also introduced for speaker modelling. The performance of the proposed system was evaluated using a redeveloped Librispeech database by adding different noises at different SNRs. A subset of 154 features based on OR strategy reduced the average EER to 1.27 % from 2.33 % for the complete set of 312 features and verification time is reduced by 49.14 %. Although the AND strategy provided smaller set of significant features with higher performance than the complete set, performance improvement was not greater than OR strategy. The best performance of the proposed feature subset is consistent for all noise conditions. Comparative analysis showed higher performance of the proposed selection than the state-of-the-art PCA and GA-based feature selection methods. The duration independence and statistical selection were applied to state-of-the-art MFCC-based features, and its effect on other short-time features needs to be investigated.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that helped improve the quality of this work.

References

Alam, M.J., Kinnunen, T., Kenny, P., Ouellet, P., & O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech Communication*, *55*(2), 237-251. https://doi.org/10.1016/j.specom.2012.08.007.

Arora, S.V., & Vig, R. (2020). An efficient text-independent speaker verification for short utterance data from mobile devices. *Multimedia Tools and Applications*, *79*, 3049-3074. https://doi.org/10.1007/s11042-019-08196-7.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, *169*, 1-12. https://doi.org/10.1016/j.compstruc.2016.03.001.

Ayyub, B.M., & McCuen, R.H. (2016). *Probability, statistics, and reliability for engineers and scientists*. CRC Press, London, New York.

Chakroborty, S., & Saha, G. (2010). Feature selection using singular value decomposition and QR factorization with column pivoting for text-independent speaker identification. *Speech Communication*, *52*(9), 693-709. https://doi.org/10.1016/j.specom.2010.04.002.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *KDD '16: Proceeding of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. San-Francisco, California, USA.

Cohen, A., & Zigel, Y. (2002). On feature selection for speaker verification. In *Proceeding COST 275 Workshop on The Advent of Biometrics on the Internet* (pp. 89-92). European Cooperation in Science and Technology. Hertfordshire, UK.

Day, P., & Nandi, A.K. (2007). Robust text-independent speaker verification using genetic programming. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(1), 285-295. https://doi.org/10.1109/tasl.2006.876765.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28-39. https://doi.org/10.1109/mci.2006.32969.

Ellis, D.P.W., & Bilmes, J.A. (2000). Using mutual information to design feature combinations. In *6th International Conference on Spoken Language Processing* (Vol. 3, pp. 79-82). International Convention Center, Beijing, China.

Eriksson, T., Kim, S., Kang, H.G., & Lee, C. (2005). An information-theoretic perspective on feature selection in speaker recognition. *IEEE Signal Processing Letters, 12*(7), 500-503. https://doi.org/10.1109/lsp.2005.849495.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *29*(2), 254-272. https://doi.org/10.1109/tassp.1981.1163530.

Ganchev, T., Zervas, P., Fakotakis, N., & Kokkinakis, G. (2006). Benchmarking feature selection techniques on the speaker verification task. In *5th International Symposium on Communication Systems, Networks and Digital Signal Processing* (pp. 314-318). IEEE. Patras, Greece.

Hirsch, H.G., & Pearce, D. (2000). The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *6th International Conference on Spoken Language Processing* (pp. 181-188). ISCA. Beijing, China.

ITU-T, P.56 (2011). Objective measurement of active speech level. *Recommendation ITU-T P.56.*

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(2), 153-158. https://doi.org/10.1109/34.574797.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE. Perth, WA, Australia. https://doi.org/10.1109/icnn.1995.488968.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Eurospeech,* (pp. 1895-1898). ISCA. Rhodes, Greece.

Murty, K.S.R., & Yegnanarayana, B. (2006). Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Processing Letters*, *13*(1), 52-55. https://doi.org/10.1109/lsp.2005.860538.

Nemati, S., & Basiri, M.E. (2010). Particle swarm optimization for feature selection in speaker verification. In *Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC* (pp. 371-380). Springer. Istanbul, Turkey. https://doi.org/10.1007/978-3-642-12239-2_39.

Nemati, S., Boostani, R., & Jazi, M.D. (2008). *Erratum: A Novel Text-Independent Speaker Verification System Using Ant Colony Optimization Algorithm*. *Image and Signal Processing*. 5099. Springer Berlin, Heidelberg. ISBN: 978-3-540-69904-0(p), ISBN: 978-3-570-69905-7(e). https://doi.org/10.1007/978-3-540-69905-7_71.

Oppenheim, A.V. (1999). *Discrete-time signal processing*. Pearson Education India. ISBN: 81-317-0492-0.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audiobooks. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 5206-5210). IEEE. South Brisbane, QLD, Australia. https://doi.org/10.1109/icassp.2015.7178964.

Pandit, M., & Kittler, J. (1998). Feature selection for a DTW-based speaker verification system. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)* (pp. 769-772). IEEE. Seattle, WA, USA.

Parui, S., Bajiya, A.K.R., Samanta, D., & Chakravorty, N. (2019). Emotion recognition from EEG signal using XGBoost algorithm. In *16th India Council International Conference* (pp. 1-4). IEEE. Rajkot, India.

Prasad, K.S., Sheela, K.A., & Sridevi, M. (2007). Optimization of TESPAR features using robust F-ratio for speaker recognition. In *IEEE International Conference on Signal Processing, Communications and Networking* (pp. 20-25). Chennai, India. https://doi.org/10.1109/icscn.2007.350673.

Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., & Jain, A.K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *4*(2), 164-171.

Saranya, M.S., Padmanabhan, R., & Murthy, H.A. (2017). Feature-switching: Dynamic feature selection for an i-vector based speaker verification system. *Speech Communication*, *93*, 53-62.

Xu, L., Liu, J., & Gu, Y. (2018). A recommendation system based on extreme gradient boosting classifier. In *10th International Conference on Modelling, Identification and Control* (pp. 1-5). IEEE. Guiyang, China. https://doi.org/10.1109/icmic.2018.8529885.

Yujin, Y., Peihua, Z., & Qun, Z. (2010). Research of speaker recognition based on combination of LPCC and MFCC. In *IEEE International Conference on Intelligent Computing and Intelligent Systems* (pp. 765-767). Xiamen, China.

Zergat, K.Y., Amrouche, A., Asbai, N., & Debyeche, M. (2012). Robust PCA-GMM-SVM system for speaker verification task. In *8th International Conference on Signal Image Technology and Internet Based Systems* (pp. 214-217). IEEE. Sorrento, Italy. https://doi.org/10.1109/sitis.2012.40.

Zigel, Y., & Cohen, A. (2004). Text-dependent speaker verification using feature selection with recognition related criterion. In *Proceedings of the Speaker and Language Recognition Workshop* (pp. 329-336). Toledo, Spain.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.