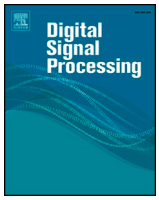
Digit. Signal Process. 156 (2025) 104811

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp



Comparative analysis of speaker identification performance using deep learning, machine learning, and novel subspace classifiers with multiple feature extraction techniques

Serkan Keser [a,*], Esra Gezer [b]

[a] Kirsehir Ahi Evran University, Department of Electrical and Electronics Engineering, Kirsehir 40100, Turkey
[b] Bursa Uludag University, Department of Electrical and Electronics Engineering, Bursa, Turkey

ARTICLE INFO

Keywords:
Speaker identification
Six different features
Hybrid classifiers
SF-CVA
Noisy speech signals

ABSTRACT

Speaker identification is vital in various application domains, such as automation, security, and enhancing user experience. In the literature, convolutional neural network (CNN) or recurrent neural network (RNN) classifiers are generally used due to the one-dimensional time series of speech signals. However, new approaches using subspace classifiers are also crucial in speaker identification. In this study, in addition to the newly developed subspace classifiers for speaker identification, traditional classification algorithms, and various hybrid algorithms are analyzed in terms of performance. Stacked Features-Common Vector Approach (SF-CVA) and Hybrid CVA-Fisher Linear Discriminant Analysis (HCF) subspace classifiers are used for speaker identification for the first time in the literature. In addition, CVA is evaluated for the first time for speaker identification using hybrid deep learning algorithms. The study includes Recurrent Neural Network-Long Short-Term Memory (RNN-LSTM), i-vector + Probabilistic Linear Discriminant Analysis (i-vector+PLDA), Time Delayed Neural Network (TDNN), AutoEncoder+Softmax (AE+Softmax), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Common Vector Approach (CVA), SF-CVA, HCF, and Alexnet classifiers for speaker identification. This study uses MNIST, TIMIT and Voxceleb1 databases for clean and noisy speech signals. Six different feature structures are tested in the study. The six different feature extraction approaches consist of Mel Frequency Cepstral Coefficients (MFCC)+Pitch, Gammatone Filter Bank Cepstral Coefficients (GTCC)+Pitch, MFCC+GTCC+Pitch+seven spectral features, spectrograms,i-vectors, and Alexnet feature vectors. High accuracy rates were obtained, especially in tests using SF-CVA. RNN-LSTM, i-vector+KNN, AE+Softmax, TDNN, and i-vector+HCF classifiers also gave high test accuracy rates.

1. Introduction

Speaker recognition technology has a wide range of applications, including voice-controlled devices [1], user authentication [2], smart homes [3], forensic analysis [4], and robotics [5]. Speaker recognition involves identifying individuals based on their speech signals and can be categorized into two main types: verification and identification. Verification systems accept or reject a speaker's claimed identity, while identification systems determine the speaker's identity among a group of registered speakers [6,7]. Despite the advancements in speaker recognition, many existing methods struggle with high noise levels and complex datasets, highlighting the need for more robust and adaptive approaches. Numerous studies on speaker recognition have employed deep learning and machine learning classifiers such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Hidden Markov Models (HMM), and Gaussian Mixture Models (GMM) [8,9]. Some of these classifiers are utilized for feature extraction, while others are employed for speaker identification [10]. While these classifiers play a crucial role in speaker recognition, the choice of features used for classification is equally essential and can significantly impact the system's performance. Most studies on speaker recognition utilize features such as Mel Frequency Cepstral Coefficients (MFCC), Gammatone Frequency Cepstral Coefficients (GFCC), spectrograms, spectral characteristics, Perceptual Linear Predictive (PLP) coefficients, and Linear Predictive

* Corresponding author.
E-mail addresses: skeser@ahievran.edu.tr (S. Keser), esraagezer@gmail.com (E. Gezer).

https://doi.org/10.1016/j.dsp.2024.104811

Available online 5 October 2024

Coding (LPC) [11–14]. However, more recent methods include Joint Factor Analysis (JFA) and i-vector-based approaches [15–17]. These methods rely on low-dimensional representations of the speech input, such as MFCC. A notable drawback of MFCCs is that their recognition performance deteriorates rapidly with increasing noise levels in the speech signal [18,19]. Since MFCCs only focus on the overall spectral envelope of the short frames used, recognition performance can be further degraded. Gammatone Filter Bank Cepstral Coefficients (GTCC) address this issue by replacing the triangular filter bank in MFCC with Gammatone filters, emphasizing different frequency bands [20,21]. Compared to MFCC, GTCC has been reported to be more robust to noise [14]. Prosodic and spectro-temporal features, such as pitch, energy, duration, rhythm, and temporal aspects, also contribute to speaker recognition. Spectral features are typically obtained by transforming the time-domain signal into the frequency domain using techniques like the Fourier Transform [22]. Pitch frequency, a perceptual characteristic of the speech signal with physical properties denoted by F0, is also utilized to enhance speaker identification performance [23,24]. Many studies apply these features individually or in combination for speaker identification [25–28].

The paper is structured as follows: Section 2 explains the structures of the speaker identification classifiers used in the study. Section 3 presents the experimental studies conducted, while Section 4 discusses the analysis of the experimental results.

1.1. Motivation and contribution

This study aims to address these challenges by developing and evaluating novel approaches to improve the accuracy of speaker recognition systems. It focuses on two key aspects influencing speaker identification performance: feature extraction effectiveness and novel subspace classifiers' introduction. The primary contribution of this research is the development and application of two innovative subspace classifiers in the field of speaker recognition, marking their first use in the literature. The first newly proposed classifier is the Hybrid CVA + FLDA (HCF) algorithm, a hybrid subspace classifier. The HCF algorithm, which combines CVA and FLDA classification capabilities, is introduced in the literature for the first time through this study. The HCF hybrid classifier adapts the hybrid Discriminative Common Vector Approach (DCVA) + Fisherface method, initially developed for image recognition, to speech identification [29]. The second classifier introduced is based on CVA. CVA, a subspace classification method, has demonstrated high recognition rates in limited, isolated word and image recognition applications [30,31]. The most important feature of CVA is that it obtains a vector containing the common features of a class from all vectors belonging to that class. Based on this, common vectors were found separately for training and testing by stacking all feature vectors for a speaker's training and testing data. Then, speaker identification was performed using the common vectors found for training and testing. This classifier is called Stacked Features CVA (SF-CVA).

The study also utilizes deep learning techniques such as i-vector + PLDA, AlexNet + Softmax, AE + Softmax, RNN-LSTM, and Time Delay Neural Networks (TDNN). Many of these classifiers were employed in hybrid forms to evaluate their performance in speaker identification. Some classifiers were used for feature extraction, while others were employed to identify speakers based on these features. In the testing phase, the feature vectors of i-vector and AlexNet were classified using SVM, KNN, CVA, SF-CVA, HCF, Bidirectional LSTM (BiLSTM), and PLDA classifiers. In addition to the i-vector and AlexNet features utilized in the study, four different feature structures were also examined during the tests. Thus, a total of six different feature structures were analyzed. Six different feature extraction methods were employed: Mel Frequency Cepstral Coefficients (MFCC) + Pitch, Gammatone Cepstral Coefficients (GTCC) + Pitch, MFCC + GTCC + Pitch + seven spectral features, spectrograms, i-vectors, and AlexNet feature vectors. MFCC and GTCC represent short-time features, while spectral features are derived from frequency domain analysis, and pitch is a prosodic feature [32].

This study uses TIMIT, MNIST, and Voxceleb1 databases for clean and noisy speech signals. A total of 120 speakers, 60 male and 60 female, were selected from the TIMIT database. The MNIST database has 60 speakers, 48 males and 12 females, and Voxceleb1 has 1251 speakers, 732 males, and 519 females—VoxCeleb1 dataset using only 4 s utterances in our experiments. In the study, zero-mean Gaussian noise was added to each speech signal with a signal-to-noise ratio of 20 dB, and the performance of the classifiers for noisy signals was also analyzed. This white noise affects all signal frequency components equally and generates moderate interference, simulating a real-world noisy environment. Silence removal and pre-emphasis were also applied to the raw speech signals for pre-processing.

1.2. Related work

The literature identifies several common classifiers for speaker identification, including i-vector + Probabilistic Linear Discriminant Analysis (PLDA) [33,34], Hidden Markov Models (HMM) [35], Universal Background Models - Gaussian Mixture Models (UBM-GMM) [36,37], RNN-LSTM [10], autoencoders [38], vector quantization (VQ) [39], neural networks [40,41], support vector machines (SVM) [42], and the Common Vector Approach (CVA) [43]. Additionally, convolutional neural networks (CNN) are often used in speaker identification due to their ability to handle noisy datasets effectively without additional features [44,45]. CNN can perform both feature extraction and classification; however, some researchers have used CNN solely as a feature extractor while employing other classifiers for the classification task [46]. These classifiers can be used independently or in hybrid forms [47–49]. Despite the diversity of classifiers, applying subspace classifiers to speaker identification remains relatively uncommon in the literature [50].

There are also recent studies in speaker identification and noise reduction for noisy speech signals [46,51,52]. In [52], Seke and Özkan grouped noisy speech signals into frames and found the common and difference vectors of the groups with CVA. It is stated that they are generally concentrated in difference vectors due to the uncorrelated nature of the noise. In this study, a denoise algorithm is applied to the difference vectors to reduce the noise, and denoised difference vectors are combined with the common vectors to recover the speech signal. In this study, it has been observed that the methods using common vectors are less affected by noise.

Another essential factor for speaker identification is feature extraction. A review on feature extraction [7] described, compared, and analyzed feature extraction methods and algorithms. Recently, a combination of features with multiple classifiers and deep learning methods has been proposed [53-55,87,88]. In [53] and [54], examples of multiple classifiers in the literature are described in detail. In [55], time-based features (MFCCT) are proposed. It is argued that the MFCCT attribute better represents speaker characteristics. This study observed a 93% accuracy rate in classification with Deep neural network (DNN) for 100 speakers in the Librispeech database. In [87] and [88], an Extreme Learning Machine (ELM) classifier was used for speaker identification. TIMIT was used as the database, and MFCC and power normalization cepstral coefficients (PNCC) were used as features. Experiments were also performed by adding Gaussian noises in the 0–30 dB range to the speech signals. In [87], the highest accuracy for clean signals is 95.83%; in [88], it is 97.52%.

In [89], a comprehensive framework for the problem of text-independent speaker identification with variable-length speech segments was proposed using databases such as Voxceleb1 and LibriSpeech. Accuracy rates between 99.67%−99.99% for 1000 speakers were obtained with Wav2Vec2 +Augmentation. These accuracy rates were obtained for 3–4 s of speech signals. This approach uses the Top-1 (%), Top-2 (%), and Top-5 (%) accuracy rate rules, which mean that a speaker is considered correctly classified if found in the top k classes for

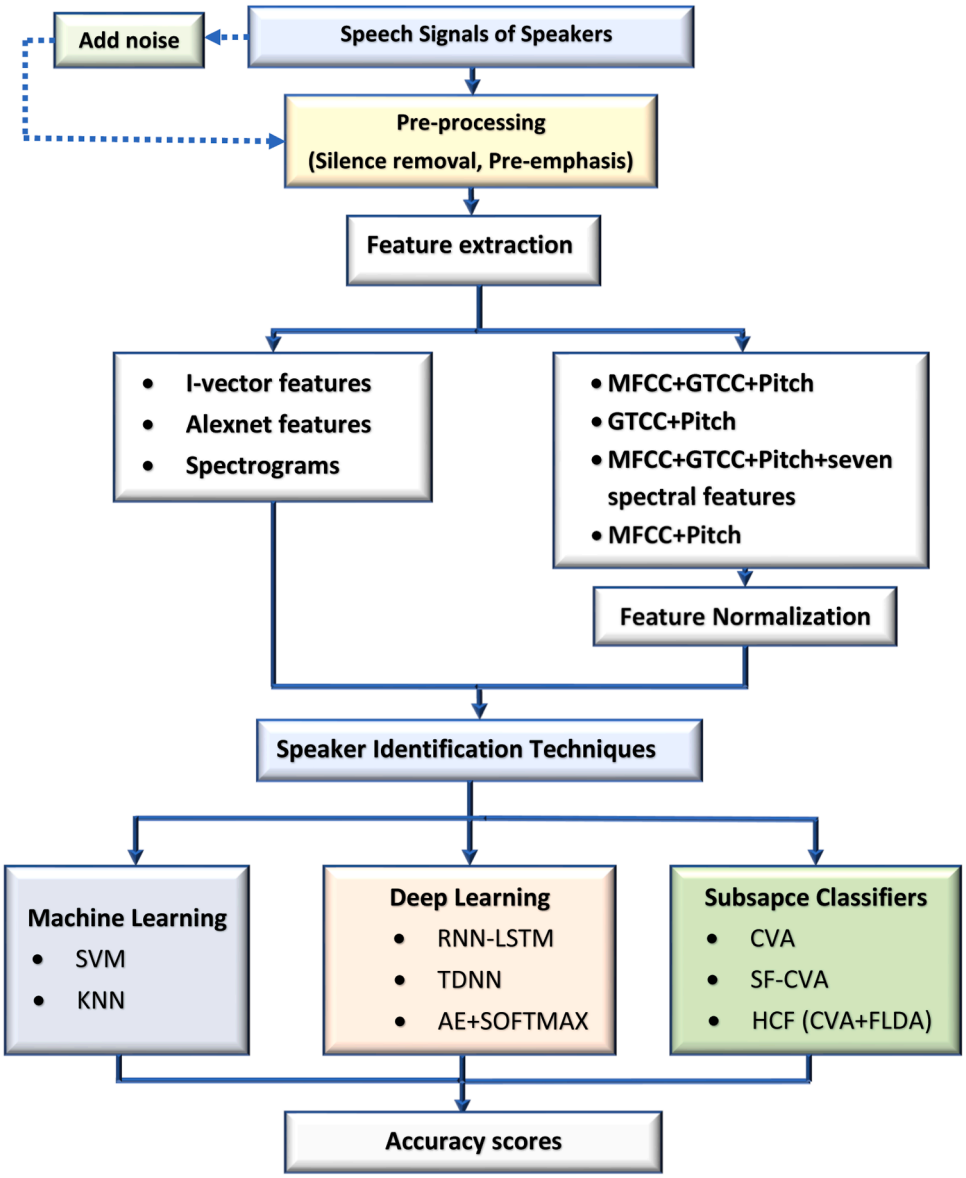


Fig. 1. The main block diagram.

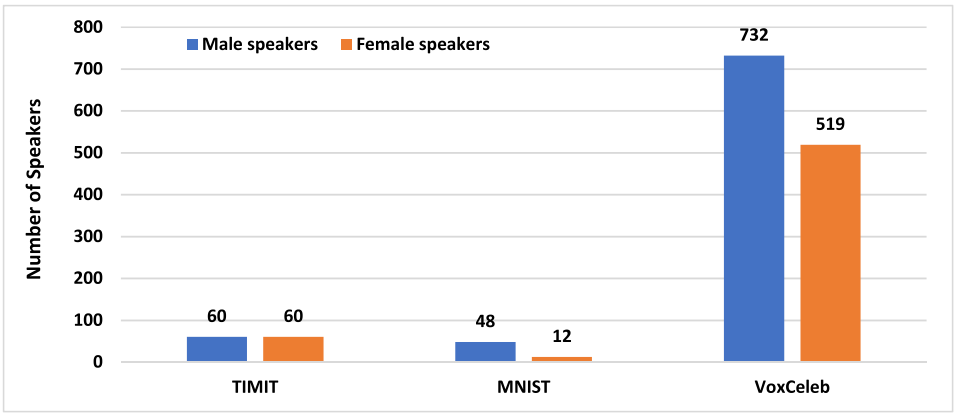
the test. Using the ResNet-18 method [90], the length of the input sequence in the Voxceleb1 dataset is fixed at 3 s. This study's Top-1 (%) and Top-5 (%) accuracy rates using MFCC features were 90.8% and 96.5%, respectively. In [91], the authors use the MFCC method to represent the feature vectors of the Voxceleb set. Various parts of the training datasets were tested with Bidirectional LSTM neural networks. Bidirectional LSTM neural networks provided up to 76.9% accuracy for individual audio segments. Apart from classifiers and features, the pre-processing process is another important factor for speaker identification. Raw speech signals often contain background noise, which leads to misclassification. Therefore, silence removal and pre-emphasis are applied to raw speech signals before extracting features [32,56].

2. Materials and methods

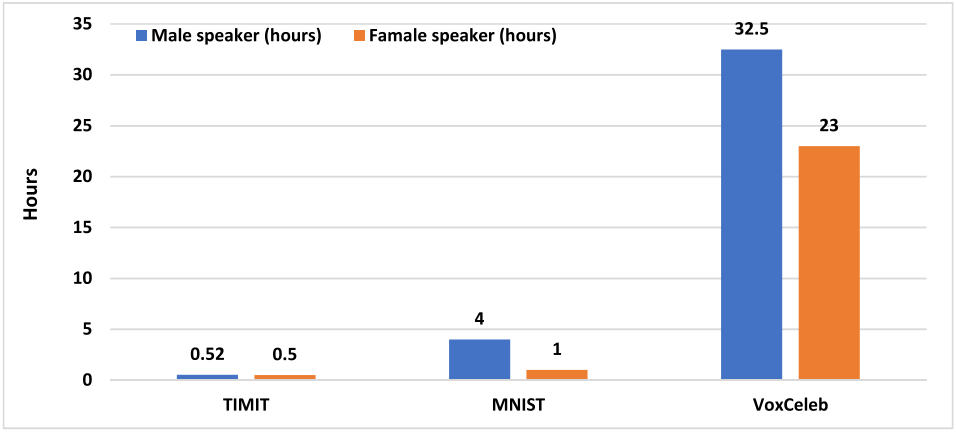
The speaker identification main block diagram used in this paper is summarized in Fig. 1. As seen from this figure, the speech signals are pre-processed to remove the silent parts of the signal and reduce the noise level through a pre-emphasis filter. After pre-processing, features are extracted from the speech signals. Normalization is applied to all feature sets, including the combinations of MFCC and GTCC. Accuracy rates are obtained for each feature type using machine learning, deep learning, and subspace classifiers. These accuracy rates are recalculated after adding noise to the speech signals with a signal-to-noise ratio (SNR) of 20 dB.

2.1. Materials

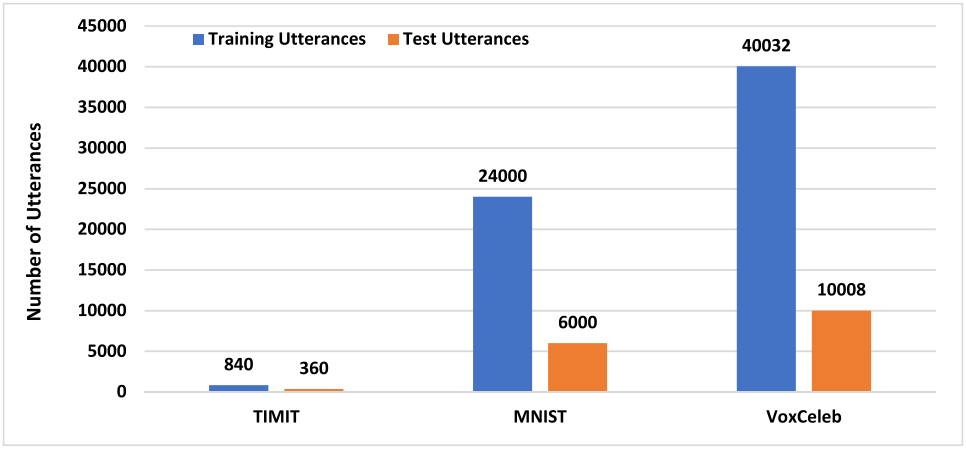
This study used the TIMIT, MNIST, and Voxceleb1 databases. The TIMIT database is a speech recognition dataset comprising 630 speakers from across the United States, with 70% male and 30% female [57]. Each speaker contributed ten utterances, with speech signals ranging from 2 to 4 s, recorded at a sampling frequency of 16 kHz. From the total 630 speakers, we selected 120 (60 male and 60 female), resulting in 1200 utterances. Of these, 840 utterances were used for training and 360 for testing. In other words, seven speech signals from each speaker were used for training and three for testing. The MNIST database, used for text-dependent speaker recognition, contains 30,000 English digit utterances (0–9) from 60 speakers, each digit repeated 50 times [58]. The signals were recorded at a sampling frequency of 48 kHz and 16-bit resolution. All 60 speakers (48 male and 12 female) were used in this study. Each speaker contributed 400 speech signals for training and 100



(a)



(b)



(c)

Fig. 2. The numbers of male and female speakers used for the three databases (a), the speaking times of males and females (b), and the total number of utterances used for training and testing (c).

for testing. For the MNIST database, male speakers have a total speaking time of about 4 h, and female speakers have about 58 min. For all speech signals in TIMIT, females have a speaking time of 32.6 min and males 30.4 min. For the Voxceleb1 database, male speakers have 32.5 h of speaking time, and female speakers have 23 h of speaking time. Fig. 2 shows the number of male and female speakers used for the three databases, the speaking time of males and females, and the total number of utterances used for training and testing.

2.2. Methods

The study employed a variety of classifiers, including hybrid classifiers. The study employed many classifiers, including SVM, KNN, CVA, SF-CVA, HCF, AE+softmax, TDNN, RNN-LSTM, i-vector+PLDA, i-vector+KNN, i-vector+SVM, i-vector+CVA, i-vector+BiLSTM, i-vector+HCF, Alexnet+Softmax, Alexnet+KNN, Alexnet+SVM, Alexnet+SF-CVA, Alexnet+HCF, and Alexnet+CVA. Table 1 shows the list of abbreviations used throughout this paper.

2.2.1. The AlexNet classifier

The pre-trained AlexNet CNN model was also used in the study [59]. While CNN yields good face recognition results for large databases, AlexNet can perform better than classical CNNs for smaller databases [60]. Therefore, the pre-trained AlexNet CNN model was also utilized in

Table 1
The list of abbreviations.

Abbreviations	Expansions
AE	Autoencoder
BiLSTM	Bidirectional Long Short-Term Memory
Conv1D	1D Convolutional
CVA	Common Vector Approach
FC+BN+ReLU	Fully Connected + Batch Normalization + Rectified Linear Unit
FLDA	Fisher Linear Discriminant Analysis
GTCC	Gammatone Cepstral Coefficients
HCF	Hybrid Common Vector Approach + Fisher Linear Discriminant Analysis
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MFCC	Mel Frequency Cepstral Coefficients
MPSA	Minimum Proportional Score Algorithm
Nf	Normalized features
PLDA	Probabilistic Linear Discriminant Analysis
RNN	Recurrent Neural Network
RUA	Recognition Update Algorithm
SF-CVA	Stacked Features- Common Vector Approach
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TPS	Time Per Speaker
UBM	Universal Background Model
WCCN	Within-Class Covariance Normalization

the study. All spectrogram images were resized to 227 × 227 for use in AlexNet. In the MATLAB environment, AlexNet consists of 25 layers, with the last three layers employed to classify the features extracted from the previous layers. In transfer learning, these last three layers are removed, and a new fully connected layer is added according to the number of classes in the database. These modified layers are then adapted to the new classification task. Thus, a new deep transfer learning model was constructed for classification. In Fig. 3, Mel spectrograms obtained in different dimensions were converted to 227 × 227 with data augmentation. Then, convolutions were carried out utilizing the conventional Alexnet architecture. For speaker identification, 4096-dimensional features obtained from the Fc7 layer were classified using AlexNet's Softmax. The study also classified the feature vectors obtained from AlexNet using KNN, SVM, CVA, SF-CVA, and HCF classifiers. AlexNet model employed a learning rate of 0.001, stochastic gradient descent with momentum (SGDM) optimization, and 60 epochs.

2.2.2. The TDNN classifier

Time delay neural network (TDNN) is one of the most popular DNNs in speaker recognition [61,62]. TDNN has demonstrated state-of-the-art performance on large datasets [63]. TDNN is a dynamic artificial neural network created by placing memory cells in the input layer of a multilayer feedforward artificial neural network [64]. Finite impulse response (FIR) filtering is utilized in the memory cells between the input and hidden layers. In the TDNN structure, the total number of connections of neurons is reduced, thereby shortening the learning time.

Additionally, TDNN helps achieve successful recognition despite potential losses of time-dependent data. Due to this filtering, the network processes the input data sequentially, considering time delays. The most used algorithm in TDNN is the temporal (time-dependent) backpropagation algorithm. For TDNN, Fig. 4 shows the TDNN speaker identification block diagram implemented in the study.

Fig. 4 depicts the TDNN speaker identification block diagram, where the initial step involves extracting feature vectors from the speech signal. The TDNN receives these features as input. The speaker's identification is accomplished by employing the TDNN structure depicted in Fig. 5. In Fig. 5, M represents the total number of frames in a speech signal, d is the dimension of features, and N represents the number of classes (speakers) in the training set. The time delay in TDNN enables the model to understand past data. At each time step, the delayed features are gathered and transmitted to the subsequent layer, the Conv1D (1D Convolutional) layer. This layer analyses the input time series and emphasizes specific features. The outputs are then normalized using Batch Normalization. This process makes the network more stable and trainable. The rectified linear unit (ReLU) activation function is applied to the Batch Normalization outputs. ReLU facilitates nonlinear learning in the network by assigning a value of zero to negative values. The delayed outputs are transmitted to the next Conv1D layer, repeating this iterative process. This approach allows the acquisition of features that are enhanced using previous information. The generated features are fed into a Fully Connected (Dense) layer and then transmitted to the output layer, which is employed for classification. The Softmax algorithm selects the class (speaker) with the highest score value. The study used a 0.001 learning rate, 256 filters, and eight epochs.

2.2.3. The RNN-LSTM classifier

Speaker recognition is a task that involves processing sequential speech data. When utilized for speaker recognition, RNN-LSTM helps analyze changes in speech signals over time and extracts speaker-specific features [65]. Long Short-Term Memory (LSTM) is a type of RNN specifically designed to handle long-term dependencies more effectively. LSTMs have a structure developed to overcome the memory issues of traditional RNNs. LSTM cells provide access to past time steps of input data and can capture long-term dependencies more effectively.

Consequently, RNN types like RNN-LSTM can be highly effective in

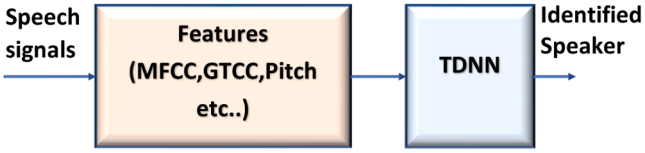


Fig. 4. TDNN speaker identification block diagram.

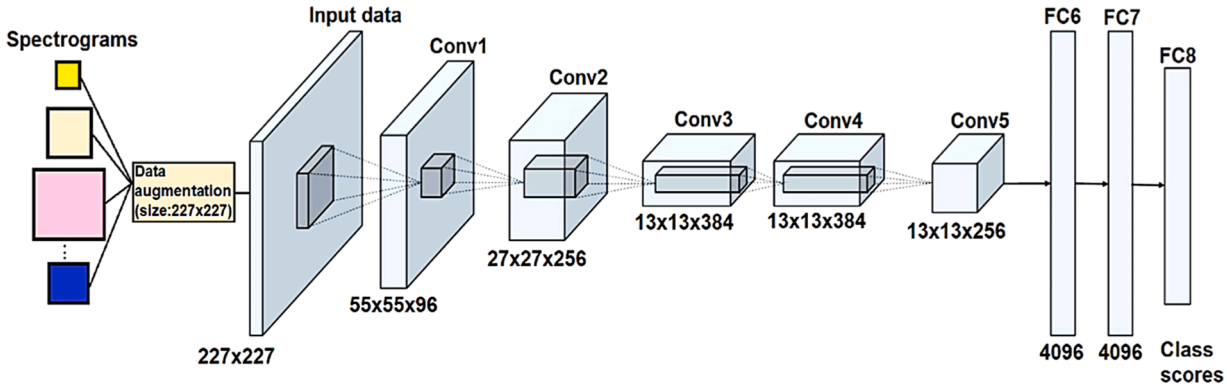


Fig. 3. Alexnet structure used for spectrograms.

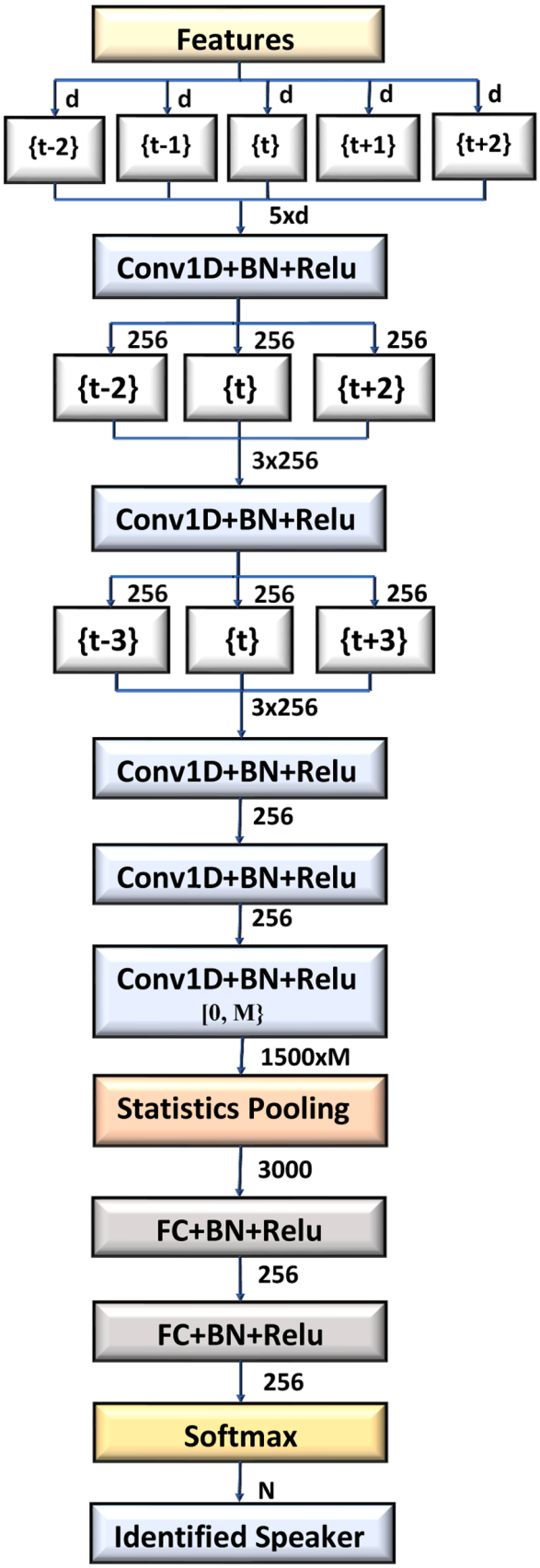


Fig. 5. Structure of the TDNN network applied in the study.

tasks that require sequential data processing, such as speaker recognition. The diagram in Fig. 6 shows the data flow in an LSTM layer with input x and output y with T time step. In the diagram, $\mathbf{h}_t$ is the output and $\mathbf{c}_t$ is the cell state at time step t. If the layer outputs the entire array, then it outputs $\mathbf{y}_1,\ldots,\mathbf{y}_T$, which is equivalent to $\mathbf{h}_1,\ldots,\mathbf{h}_T$. If the layer outputs only the last time step, the layer outputs $\mathbf{y}_T$, which is equivalent to $\mathbf{h}_T$. The number of channels at the output equals the number of hidden units of the LSTM layer [66]. The first LSTM process uses the initial state of the RNN and the first step of the sequence to calculate the initial output and updated cell state. At time step t, the process uses the current state of the RNN $(\mathbf{c}_{t-1},\mathbf{h}_{t-1})$. Next, the output and updated cell state are calculated for the next time step of the sequence [67]. The state of the layer consists of the hidden state and the cell state. The hidden state at time step t contains the output of the LSTM layer for that time step. The cell state contains information learned from previous time steps. The layer adds or subtracts information to the cell state at each time step. The layer controls these updates using gates [68].

The LSMT unit consists of input gate (I), forget gate (f), cell candidate (g), and output gate (o). Fig. 7 indicates a diagram of an LSTM unit.

The learnable weights of an LSTM layer are the input weights W, the recurrent weights R, and the bias b. The matrices W, R, and b are concatenations of the input weights, the recurrent weights, and the bias of each component, respectively [68]. The layer concatenates the matrices according to Eq. (1):

$$W=\begin{bmatrix}W_i\\W_f\\W_g\\W_o\end{bmatrix},\ R=\begin{bmatrix}R_i\\R_f\\R_g\\R_o\end{bmatrix},\ b=\begin{bmatrix}b_i\\b_f\\b_g\\b_o\end{bmatrix}, \tag{1}$$

where $i.f,g$, and o denote the input gate, forget gate, cell candidate, and output gate, respectively. The cell state at time step t is given by

$$\mathbf{c}_t=f_t\odot\mathbf{c}_{t-1}+i_t\odot g_t, \tag{2}$$

where $\odot$ denotes the Hadamard product. The hidden state at time step t is given by

$$\mathbf{d}_t=o_t\odot\eta_c(\mathbf{c}_t), \tag{3}$$

where η_c is the state activation function, and the hyperbolic tangent function is used to calculate it. The states of the blocks of the LSTM unit at time t can be written using Eq. (4).

$$\begin{aligned}i_t&=\eta_g(W_i\mathbf{x}_t+\mathrm{R}_i\mathbf{d}_{t-1}+b_i)\\f_t&=\eta_g\left(W_f\mathbf{x}_t+\mathrm{R}_f\mathbf{d}_{t-1}+b_f\right)\\g_t&=\eta_c\left(W_g\mathbf{x}_t+\mathrm{R}_g\mathbf{d}_{t-1}+b_g\right)\\o_t&=\eta_g(W_o\mathbf{x}_t+\mathrm{R}_o\mathbf{d}_{t-1}+b_o)\end{aligned} \tag{4}$$

The RNN-LSTM network architecture in this study consists of an input layer, an LSTM layer with 120 hidden units, a fully connected layer, a softmax classification layer, and the classification output. Fig. 8 illustrates the RNN-LSTM structure used for 40-dimensional feature vectors, where N represents the number of classes.

2.2.4. *The autoencoder+ softmax classifier*

An autoencoder is a neural network trained to get its input at the output. The training phase of an autoencoder does not require labeled data and is, therefore, unsupervised [69]. The training process is based on the optimization of a cost function. The cost function measures the error between the input and the recreation of the input at the output. The autoencoder reduces the input data size in the hidden layer by removing unimportant features. The features in the hidden layer are smaller than the original input data size. In the study, the 40, 86, and 107-dimensional input feature vectors were reduced to 38, 80, and 90

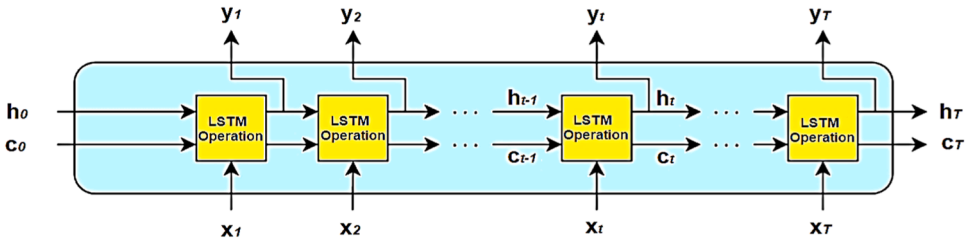


Fig. 6. The diagram of the data flow in an LSTM layer.

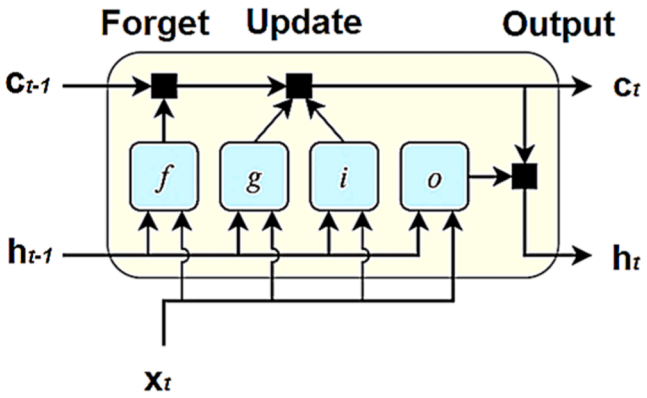


Fig. 7. The diagram of the data flow in an LSTM layer.

dimensions, respectively, in the hidden layer of the autoencoder. For classification, the features obtained in the hidden layer were classified using Softmax. The L2 wt Regularization value was set to 0.004. Fig. 9 shows the autoencoder structure used in the test phase, with 90-dimensional features in the hidden layer.

2.2.5. *The i-vector classifier*

An "i-vector" system has become the state-of-the-art speaker recognition technique [70]. These i-vector systems reduce large input data to a small-dimensional feature vector while retaining the most relevant information [71]. The i-vector extractor transforms the sequence of feature frames into a single low-dimensional vector representing the entire utterance. A Universal Background Model (UBM) is required to collect statistics from the speech utterances. The UBM contains N Gaussian components and includes mean vectors μ(n), covariance matrices Σ(n), and weights ω(n). Fig. 10 shows the block diagram of the i-vector system used for testing. The study used UBM with 64 Gaussian components to obtain the i-vector. The i-vectors used had dimensions of 40, 50, and 60.

First, the feature vectors are applied to the i-vector extractor. The GMM supervector is computed, and then the i-vector is extracted. For post-processing, LDA, within-class covariance normalization (WCCN), and whitening are applied to the i-vectors. Then, PLDA generates as many score values as the number of classes, and the class (speaker) with the highest score value is selected. For post-processing, LDA is first applied to the speakers' i-vector. LDA tries to minimize the intra-class variance and maximize the variance between speakers. We first find the intra-class and inter-class scatter matrices for LDA using Eq. (5) and Eq. (6).

$$S_b = \sum_{s=1}^{S} (\overline{a_s} - \overline{a})(\overline{a_s} - \overline{a})' \tag{5}$$

$$S_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (a_i^s - \overline{a_s})(a_i^s - \overline{a_s})' \tag{6}$$

Where $\overline{a_s}$ is the average of the i-vectors for each speaker, $\overline{a}$ is the average i-vector for all speakers and n_s is the number of utterances for each speaker. The best eigenvectors are found using Eq. (7).

$$S_b v = \lambda S_w v \tag{7}$$

The eigenvectors with the highest eigenvalues are then used to find the LDA projection matrix and projected i-vectors. WCCN attempts to scale the i-vector space inversely to the intraclass covariance so that aspects of high intra-speaker variability are not emphasized in i-vector comparisons [72]. The intraclass covariance matrix (WCNN) was found using Eq. (8).

$$W = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (a_i^s - \overline{a_s})(a_i^s - \overline{a_s})' \tag{8}$$

Where s represents the number of speakers, B is computed using Cholesky decomposition. The WCNN projection matrix is found by multiplying the LDA projection matrix. The WCNN projection matrix is applied to the training dataset, and trained i-vectors are obtained. A whitening matrix is then found using the trained i-vectors. This whitening matrix is applied to the trained i-vectors to perform whitening. After this process, score values are found for the i-vectors using PLDA, for the test signal, features and i-vectors are found, respectively. The i-vectors are then post-processed and finally classified using PLDA. The system performance is evaluated in terms of accuracy (%). In addition to i-vector+PLDA, i-vector+BiLSTM, i-vector+KNN, i-vector+SVM, i-vector+CVA, i-vector+SF-CVA, and i-vector+HCF classifiers were used in the study.

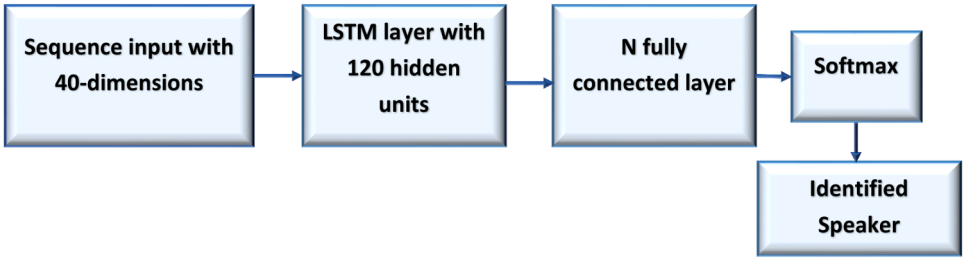


Fig. 8. The structure of the RNN-LSTM network for 40-dimensional feature vectors.

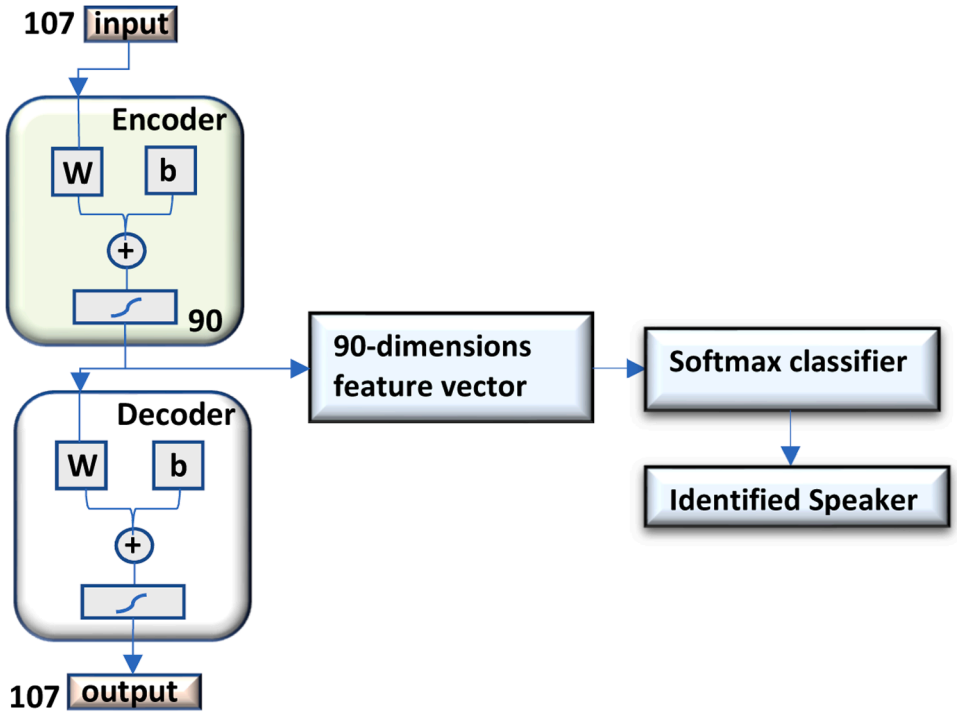


Fig. 9. An autoencoder structure with 90-dimensional features in the test phase.

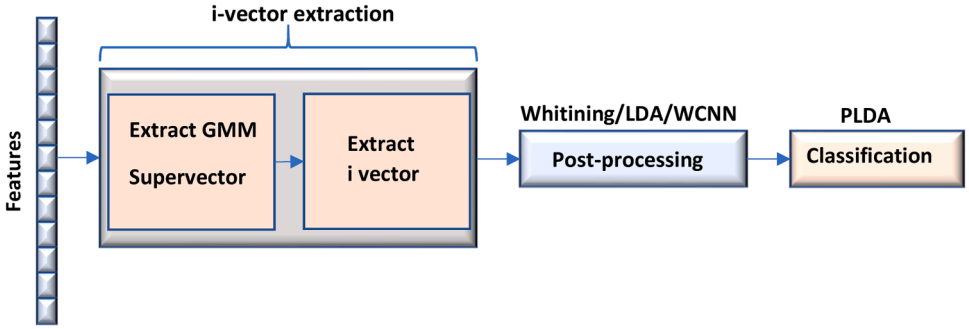


Fig. 10. Block diagram of i-vector system.

2.2.6. *The KNN classifier*

In the k-nearest neighbors algorithm, a vector is classified by multiple votes of its neighbors [73]. Here, the vector is assigned to the most common class among its k nearest neighbors (k is a positive integer, typically small). In the study, k-values of 5 and 10 were used. Additionally, the Euclidean distance criterion was utilized.

2.2.7. *The CVA and SF-CVA classifiers*

The common vector approach (CVA) is a subspace classification method used in speech and image recognition. In this method, a vector containing the invariant features of each class is obtained, referred to as the "common vector" [74]. If there are c classes in the training set, each with k examples, there will be a total of $m = k \times c$ examples in the training set. Here, m represents the number of vectors belonging to a speaker class, and n is the size of each vector. CVA can be applied in both cases of sufficient data ($m \geq n$) and insufficient data ($m < n$) [74]. If we represent the rth signal sample of class I with x_j^i in the n-dimensional space, the within-class scatter matrix S_w is given as follows:

$$S_w = \sum_{i=1}^{c} \sum_{r=1}^{k} \left(\left(x_r^i - \mu_i\right) \left(x_r^i - \mu_i\right)^T \right) \tag{9}$$

Here, μ_i represents the average vector of the ith class. It is divided into two subspaces perpendicular to each other: the difference subspace $\mathbf{B}$ and the indifference subspace $\mathbf{B}^{\perp}$. The indifference subspace $\mathbf{B}^{\perp}$ is spanned by the eigenvectors corresponding to the zero eigenvalues of the matrix $\boldsymbol{S_w}$. If P and $\overline{\boldsymbol{P}}$ matrices are taken as the projection matrices of $\mathbf{B}$ and $\mathbf{B}^{\perp}$ subspaces, respectively; the projections of the samples in the training set into $\mathbf{B}^{\perp}$ subspace will be as follows [30,31]. The common vector of the ith class ($\boldsymbol{x}_{com}^i$) is indicated in Eq. (10).

$$\boldsymbol{x}_{com}^i = \boldsymbol{x}_r^i - \boldsymbol{P}\boldsymbol{x}_r^i = \overline{\boldsymbol{P}}\boldsymbol{x}_r^i, \; r = 1, \; 2, \ldots, \mathrm{k}, \; I = 1, 2, \ldots, \mathrm{c} \tag{10}$$

CVA was used in two ways in the study. First, classical CVA was applied to the speaker's feature vectors. Secondly, in the testing phase, the feature vectors of each speaker were stacked and projected to the indifference subspace using the P projection matrix, and a common vector was found and compared with the common vectors which were obtained by stacking the feature vectors of each speaker and projecting them onto the indifference subspace during the training phase. A similar process was applied to the test feature vectors, followed by CVA classification. This approach is referred to as Stacked Features-CVA (SF-CVA).

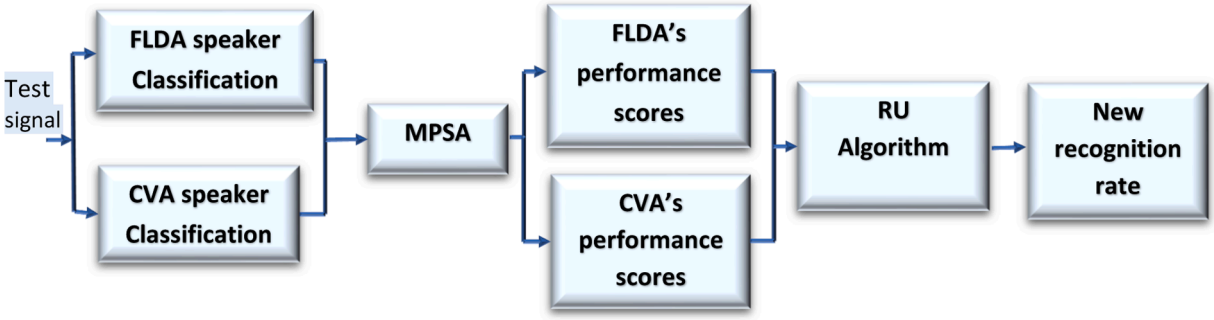


Fig. 11. The block diagram for the testing phase of HCF.

2.2.8. The HCF classifier

A hybrid classifier was implemented using CVA and FLDA, following a methodology similar to the HDF algorithm [29] based on recognition performance in image recognition. This classifier is referred to as Hybrid CVA+FLDA (HCF). Similar to HDF, the Minimum Proportional Score Algorithm (MPSA) and Recognition Update Algorithm (RUA) are employed in the HCF classifier. Classification with the MPSA approach is based on the performance scores obtained instead of Euclidean distance. The classifier with the highest performance is determined as the base classifier. If this base classifier misclassifies any test signals according to the Euclidean distance, these signals can be assigned to the correct class using the performance scores from both the base classifier and the other classifier. With the proposed mathematical method, the performance score matrix $\left(\boldsymbol{S}_{uv}^{k}\right)$ of the classifiers for the samples belonging to all classes is as follows [29],

$$\boldsymbol{S}_{uv}^{k} = \left(\frac{d_{x_{\min}}^{k}}{d_{x_{uv}}^{k}}\right)^{2} + \left(\frac{d_{x_{\min}}^{k}}{d_{x_{uv}}^{k}}\right)^{2} \cdots + \left(\frac{d_{x_{\min}}^{k}}{d_{x_{uv}}^{k}}\right)^{2},\ u = 1, 2, \ldots, n \text{ and } v = 1, 2, \ldots, m, \tag{11}$$

where $d_{x_{\min}}^{k}$ represents the minimum Euclidean distance among classes for a given test signal x, where u denotes the class index, n indicates the total number of classes, k symbolizes the index of a specific classifier, v refers to the test signal index, and m represents the number of samples. The classifiers' effectiveness is evaluated using the MPSA algorithm. These classifiers are scored using a matrix of overall performance values (S_{uv}^{2},S_{uv}^{1}) across all test signals. Subsequently, for each test signal v within class u, the disparity in scores ($dif_{u,v} = S_{uv}^{2} - S_{uv}^{1}$) is calculated. If the difference is positive, the *classifier* ^{1} has better performance, and this positive difference is added to its performance score $\left(pf^{1}\right)$. Conversely, if the difference is negative, the *classifier* ^{2} has better performance, and the performance value (pf^{2}) of the *classifier* ^{2} is assigned the absolute value of this difference. Following the conclusion of this process, the summed performance scores for pf^{1} and pf^{2} are allocated to ts^{1} and ts^{2} respectively. The comparison of total scores, where ts^{1} exceeds ts^{2}, suggests that *classifier* ^{1} outperforms *classifier* ^{2} across all test signals, resulting in its selection as the base classifier. If not, *classifier* ^{2} is chosen [29].

Algorithm 1. Minimum Proportional Score Algorithm.

Input: S_{uv}^{1},S_{uv}^{2}, s1=0, s2=0.
Output: pf^{1}, pf^{2}, ts^{1}, ts^{2}, b_cl //b_cl: base classifier.
for $u = 1:n$ **do**
  for $v = 1:m$ **do**
    $dif_{u,v} = S_{uv}^{2}$-S_{uv}^{1}
    if $dif_{u,v} > 0$
      $s1 = s1 + 1$
      $pf^{1}(s1) = dif_{i,j}$
    else
      $s2 = s2 + 1$

(continued on next column)

(continued)

      $pf^{2}(s2) = \left|dif_{i,j}\right|$
  end for
end for
$ts^{1} = sum\left(pf^{1}\right)$
$ts^{2} = sum\left(pf^{2}\right)$
if $ts^{1} > ts^{2}$
  $b\_cl = classifier^{1}$
else
  $b\_cl = classifier^{2}$
end for
end algorithm

Using the MPSA algorithm, the base classifier is chosen, after which the RUA method is applied. The previously mentioned *euc_class*^{1} and *euc_class*^{2} matrices are transformed into *class*^{1} and *class*^{2} matrices. The RUA method uses a recognition matrix called *euc_class*, based on Euclidean values from the base classifier. Any misclassifications in this matrix are corrected using the performance score matrices (S_{uv}^{2}, S_{uv}^{1}) from the base and another classifier. For a test signal, if both classifiers correctly classify the signal, the updated recognition matrix (*up_rec*) is marked as "*cc*" (correct classification). If not, it is marked as "*fc*" (false classification). The *up_rec* matrix has dimensions $n \times m$, representing n classes and m test signals [29]. The details of the RUA algorithm are given below.

Algorithm 2. Recognition Update Algorithm.

Input: S_{uv}^{2},S_{uv}^{1},$class^{1}$, $class^{2}$, b_cl
Output: up_rec
for $u = 1:n$ **do**
  for $v = 1:m$ **do**
    if $(class^{1}(u,v) = cc)$ & $(class^{2}(u,v) = cc)$
      $up\_rec\{u,v\} = cc$;
    elseif $(class^{1}(u,v) = fc)$ & $(class^{2}(u,v) = fc)$
      $up\_rec\{u,v\} = fc$;
    elseif $(class^{1}(u,v) = cc)$ & $(class^{2}(u,v) = fc)$& $S_{uv}^{1}<S_{uv}^{2}$ & $b\_cl = classifier^{2}$
      $up\_rec\{u,v\} = cc$;
    elseif $(class^{1}(u,v) = fc)$ & $(class^{2}(u,v) = cc)$& $S_{uv}^{1}>S_{uv}^{2}$ & $b\_cl = classifier^{1}$
      $up\_rec\{u,v\} = cc$;
    elseif $(class^{1}(u,v) = cc)$ & $(class^{2}(u,v) = fc)$& $S_{uv}^{2}<S_{uv}^{1}$& $b\_cl = classifier^{2}$
      $up\_rec\{u,v\} = fc$;
    elseif $(class^{1}(u,v) = fc)$ & $(class^{2}(u,v) = cc)$& $S_{uv}^{2}>S_{uv}^{1}$& $b\_cl = classifier^{1}$
      $up\_rec\{u,v\} = fc$;
    elseif $(class^{1}(u,v) = fc)$ & $(class^{2}(u,v) = cc)$ & $b\_cl = classifier^{2}$
      $up\_rec\{u,v\} = cc$;
    elseif $(class^{1}(u,v) = cc)$ & $(class^{2}(u,v) = fc)$ & $b\_cl = classifier^{1}$
      $up\_rec\{u,v\} = cc$;
  end for
end for
end for
end algorithm

Fig. 11 shows the block diagram for the testing phase of HCF, which

Table 2
The total dimensions of the study's MFCC and GTCC features.

MFCC+GTCC+Pitch+7 spectral features	MFCC+GTCC+Pitch+7 spectral features	MFCC+Pitch features	GTCC+Pitch features
20 MFCC	13 MFCC	13 MFCC	13 GTCC
20 Δ MFCC	13 Δ MFCC	13 Δ MFCC	13 ΔGTCC
20 ΔΔ MFCC	13 ΔΔ MFCC	13 ΔΔ MFCC	13 ΔΔ GTCC
13 GTCC	13 GTCC	Pitch	Pitch
13 ΔGTCC	13 ΔGTCC		
13 ΔΔ GTCC	13 ΔΔ GTCC		
Pitch	Pitch		
7 spectral features	7 spectral features		
107-dimensional feature	**86-dimensional feature**	**40-dimensional feature**	**40-dimensional feature**

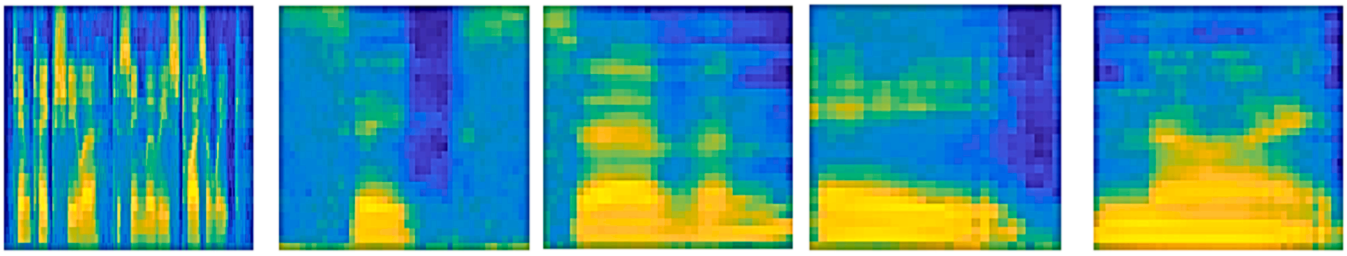


Fig. 12. Some Mel-spectrograms of a speaker's different utterances.

is a Hybrid classifier [29].

2.2.9. The SVM classifier

As a robust classifier, the SVM aims to find a separating hyperplane between two classes [75]. SVM classifiers can use kernel functions to solve nonlinear classification problems. Kernel functions map input vectors into a higher-dimensional space, allowing data that cannot be linearly classified in the input space to become linearly classifiable in the expanded space. In this study, Gaussian and polynomial kernels were used.

3. Experimental studies

The experimental studies were conducted for the MNIST, TIMIT, and Voxceleb1 databases with 60, 120, and 1251 speakers, respectively. The entire study was performed using 5-fold cross-validation. The signals of the speakers in the Voxceleb1 database were fixed to 4 s, and 40 audio files were used for each speaker. In addition to accuracy, the performance of the classifiers was evaluated based on test computation time and complexity.

3.1. Feature extraction

Before extracting the features, pre-processing is applied to each speech signal, and the feature vectors are obtained. The pre-emphasis filter used in the study is $H(z) = 1 - 0.97z^{-1}$. After pre-processing, features are extracted from the speech signals. Normalization is applied to all features, including the MFCC and GTCC combinations. This process is illustrated in Eq. (12).

$$N_f = (f - M)/S \tag{12}$$

Where M is the mean of the features, S is the standard deviation of the features, and N_f is the normalized features. Accuracy rates were obtained using machine learning, deep learning, and subspace classifiers, separately, using noisy and clean speech signals for each feature type. For MFCC and GTCC features, the speech signals were divided into 30 ms frames with a 15 ms overlap. Four different feature extraction methods were applied to each speech signal. The first method includes MFCC and pitch frequency features; the second method includes GTCC and pitch frequency features; the third method includes GTCC, MFCC, pitch frequency, and spectral features (entropy, kurtosis, skewness, slope, spectral centroid, zero cross-ratio, short-time energy), and the fourth method includes spectrograms. Table 2 shows the total dimensions of the MFCC and GTCC features used in the study.

Mel Spectrogram shows the intensity of various frequency components in the speech signal, usually using a color map representing intensity. The spectrogram's horizontal axis represents time, and the vertical axis represents frequency (on the Mel scale). Sample Mel spectrograms of some speech signals used in the study are shown in Fig. 12 for the MNIST database. The dimensions of these spectrograms are 227 × 227.

In the study, spectrograms were resized to 227 × 227 to be applied to AlexNet. In addition to these features, i-vector and AlexNet feature vectors were used with PLDA, BiLSTM, KNN, SVM, CVA, SF-CVA, and HCF classifiers for speaker identification. In total, six different feature extraction methods were examined.

3.2. Performance metrics

In the study, the identification accuracy of all speakers is calculated. The average of these identification accuracies represents the overall identification accuracy of the proposed speaker identification system. The accuracy rates are found using Eq. (13).

$$\text{Ac} = \frac{\textit{Correctly identified recordings}}{\textit{Total testing recordings}} \times 100 \tag{13}$$

The computation time of the classifiers for the test data was evaluated in seconds. This process is calculated by averaging the computation time per speaker of the classifiers for the two databases according to Eq. (14).

$$TPS_k = \frac{\textit{Total computation time for test}}{\textit{The number of speakers per database}}, \quad k = 1, 2, \ldots, C \tag{14}$$

where k is the classifier index, C is the number of classifiers used in the study, and TPS is the time per speaker. The TPS value is found separately for the MNIST and TIMIT databases, and then the two TPS values are averaged.

Table 3
Accuracy results that found using MFCC+Pitch and GTCC+Pitch.

	MNIST		TIMIT		Voxceleb1	
Classifiers	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)
AE+Softmax	93.64	91.52	88.40	87.56	51.23	51.98
RNN-LSTM	99.25	98.45	94.71	92.10	54.74	55.59
TDNN	98.28	96.89	92.22	89.71	55.81	56.89
CVA	73.69	69.29	83.12	80.34	51.73	50.17
SF-CVA	99.67	99.33	98.50	97.50	65.78	63.53
HCF	78.47	73.39	87.35	83.90	55.68	53.67
i-vector+PLDA	91.20	92.56	84.20	79.58	57.28	56.42

Table 4
Accuracy results that found using MFCC+Pitch and GTCC+Pitch for noisy speech signals.

	MNIST		TIMIT		Voxceleb1	
Classifiers	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)	MFCC+ pitch (40 -dimensional)	GTCC+pitch (40 -dimensional)
AE+Softmax	91.27	91.10	84.60	83.34	48.42	49.18
RNN-LSTM	93.58	93.40	81.30	82.15	50.29	51.59
TDNN	90.78	90.32	82.94	84.75	51.18	52.28
CVA	72.33	63.74	76.98	73.94	48.76	47.55
SF-CVA	99.00	99.66	95.16	94.80	60.85	59.46
HCF	75.84	70.16	80.48	77.24	51.12	50.44
i-vector+PLDA	87.16	90.18	78.23	75.74	51.33	51.46

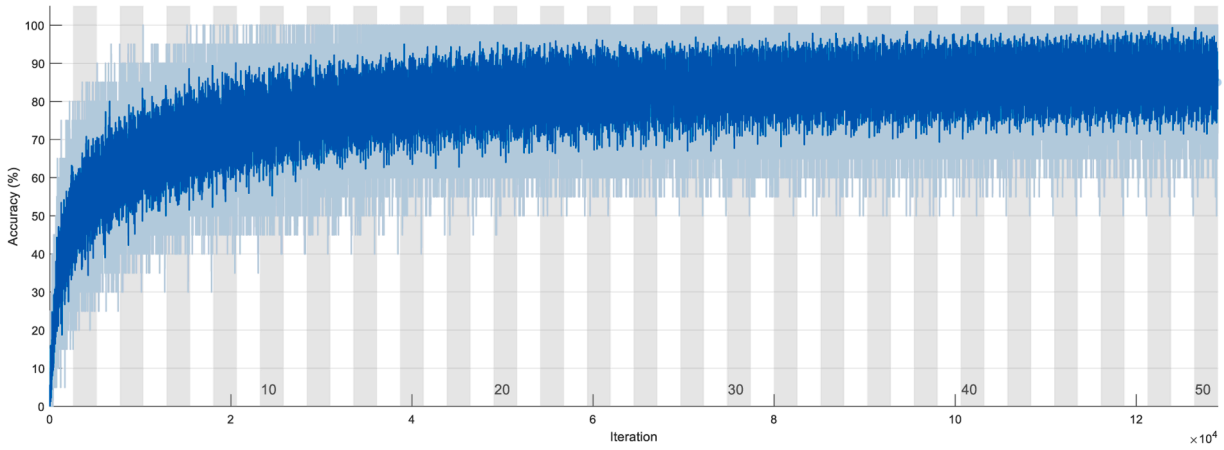


Fig. 13. Training process for RNN-LSTM.

Table 5
Accuracy results that found using MFCC+ GTCC+Pitch+Spectral features for clean speech signals.

	MNIST		TIMIT		Voxceleb1	
Classifiers	(86-dimensional)	(107 -dimensional)	(86-dimensional)	(107 -dimensional)	(86-dimensional)	(107 -dimensional)
AE+Softmax	97.80	99.25	96.42	96.84	55.72	57.64
RNN-LSTM	99.04	99.23	97.75	97.15	59.06	61.25
TDNN	96.49	98.69	95.25	95.94	58.74	59.86
CVA	78.67	81.01	89.12	93.93	53.83	55.48
SF-CVA	98.66	99.00	99.00	99.33	69.62	71.76
HCF	81.05	83.81	92.51	96.18	56.94	58.36
i-vector+PLDA	92.25	94.25	97.10	98.74	53.28	54.72

Table 6
Accuracy results that found using MFCC+ GTCC+Pitch+Spectral features for noisy speech signals.

	MNIST		TIMIT		Voxceleb1	
Classifiers	(86-dimensional)	(107 -dimensional)	(86-dimensional)	(107 -dimensional)	(86-dimensional)	(107 -dimensional)
AE+Softmax	92.50	94.02	92.53	93.41	49.46	51.78
RNN-LSTM	92.56	95.64	86.97	89.91	52.12	53.85
TDNN	84.64	87.12	80.39	82.41	51.17	52.56
CVA	70.20	72.95	83.93	88.16	49.06	51.14
SF-CVA	99.00	99.33	97.91	99.00	64.27	67.46
HCF	75.99	77.23	87.17	90.45	52.44	54.56
i-vector+PLDA	88.36	92.43	86.98	89.14	47.55	48.93

Table 7
Accuracy results that found using Alexnet for clean speech signals.

Classifiers	MNIST	TIMIT	Voxceleb1
Alexnet+Softmax	96.56	91.34	46.88
Alexnet+SVM	85.07	81.20	40.22
Alexnet+KNN	76.43	85.92	41.74
Alexnet+CVA	81.08	77.56	39.78
Alexnet+ SF-CVA	99.52	98.81	72.43
Alexnet+HCF	86.49	84.77	42.25

Table 8
Accuracy results that found using Alexnet for noisy speech signals.

Classifiers	MNIST	TIMIT	Voxceleb1
Alexnet+Softmax	93.06	87.16	41.56
Alexnet+SVM	82.72	74.90	35.96
Alexnet+KNN	73.30	72.44	35.22
Alexnet+CVA	75.29	72.61	33.27
Alexnet+ SF-CVA	99.16	96.66	65.45
Alexnet+HCF	79.87	81.78	37.78

3.3. Performance evaluation of classifiers

In the study, the accuracy rates of the classifiers were presented in Table 3, using MFCC + pitch (13 MFCC + 13 ΔMFCC + 13 ΔΔ MFCC + pitch) and GTCC + pitch (13 GTCC + 13 ΔGTCC + 13 ΔΔ GTCC + pitch) features for the MNIST and TIMIT databases. Table 4 shows the accuracies obtained for noisy speech signals using the same feature types. All accuracy rates were calculated using 5-fold cross-validation.

In Table 3, the classifiers that achieved the highest accuracy were SF-CVA and RNN-LSTM. In Tables 3 and 4, SF-CVA achieved the highest accuracy rates across all datasets. For the MNIST dataset, SF-CVA achieved 99.67% accuracy with MFCC + Pitch and 99.33% with GTCC + Pitch. For the TIMIT dataset, SF-CVA achieved 98.50% accuracy with MFCC + Pitch and 97.50% with GTCC + Pitch. For the TIMIT dataset, RNN-LSTM achieved 94.71% accuracy with MFCC + Pitch and 92.10% with GTCC + Pitch. For the Voxceleb1 dataset, all classifiers had lower accuracy rates. SF-CVA achieved 65.78% accuracy with MFCC + Pitch and 63.53% with GTCC + Pitch.

In Table 4, for the MNIST dataset, SF-CVA achieved 99.67% accuracy with GTCC + Pitch. For the TIMIT dataset, SF-CVA achieved 94.45% accuracy with MFCC + Pitch and 93.33% with GTCC + Pitch. RNN-

Table 9
Accuracy results that found using i-vector for clean speech signals.

	MNIST			TIMIT		
Classifiers	40- dimensional	50-dimensional	60-dimensional	40-dimensional	50-dimensional	60-dimensional
i-vector + PLDA	92.65	95.00	94.97	88.48	88.83	92.41
i-vector + KNN	98.83	98.98	99.16	81.89	78.02	82.38
i-vector + BiLSTM	95.33	95.42	96.58	98.26	98.20	98.86
i-vector + SVM	89.13	96.34	96.53	76.20	75.67	76.58
i-vector + CVA	90.76	91.67	92.86	97.01	98.59	99.25
i-vector + SF-CVA	99.00	99.33	99.66	99.16	99.33	99.66
i-vector + HCF	97.08	98.01	97.06	98.13	99.32	99.29

Table 10
Accuracy results that found using i-vector for clean speech signals.

	Voxceleb1		
Classifiers	40- dimensional	50-dimensional	60-dimensional
I-vector + PLDA	62.75	63.24	65.38
i-vector + KNN	59.42	59.85	61.86
i-vector + BiLSTM	63.35	63.71	64.58
i-vector + SVM	70.69	71.21	72.59
i-vector + CVA	55.28	55.86	58.28
i-vector + SF-CVA	79.14	80.76	81.25
i-vector + HCF	56.02	57.34	60.77

LSTM achieved 93.58% accuracy with MFCC + Pitch and 93.40% with GTCC + Pitch on the MNIST dataset. For the TIMIT dataset, TDNN achieved 82.94% accuracy with MFCC + Pitch and 84.75% with GTCC + Pitch. For the Voxceleb1 dataset, SF-CVA achieved 60.85% accuracy with MFCC + Pitch and 59.46% with GTCC + Pitch. Fig. 13 shows an example of the RNN-LSTM training process using 107-dimensional features, 50 epochs, the Adaptive Moment (Adam) optimization algorithm, and a learning rate 0.001.

Tables 5 and 6 present the accuracy results obtained by combining MFCC + GTCC + Pitch + Spectral features. The i-vector + PLDA model consists of 128 components, and for 86- and 107-dimensional features, the total variability space (TVS) ranks are 80 and 90, respectively.

In Table 5, the AE + Softmax, RNN-LSTM, SF-CVA, and TDNN classifiers achieved the highest accuracy rates. Among them, SF-CVA provided the highest accuracy compared to the other three classifiers. For the Voxceleb1 dataset, SF-CVA achieved 69.62% accuracy with MFCC + Pitch and 71.76% with GTCC + Pitch.

Table 6 gives the accuracy results for noisy speech signals using the same feature sets. AE + Softmax achieved 94.02% accuracy for the MNIST dataset and 92.53% for the TIMIT dataset. RNN-LSTM also performed well on noisy data, achieving 95.64% accuracy for MNIST and 89.91% for TIMIT. SF-CVA again performed the best on the MNIST and TIMIT datasets, achieving 99.33% and 99% accuracy, respectively. Tables 7 and 8 present the accuracy results of the 4096-dimensional AlexNet features tested on clean and noisy speech datasets classified by various classifiers. In Table 7, AlexNet + SF-CVA achieved the highest accuracy, reaching 99.52% in MNIST and 98.81% in TIMIT, showing strong performance in speaker identification tasks. AlexNet + Softmax also performed well, achieving 96.56% accuracy for MNIST and 91.34% for TIMIT. AlexNet + HCF outperformed other classifiers, such as KNN and SVM, achieving an accuracy of about 86.49%. For the Voxceleb1 dataset, SF-CVA achieved an accuracy of 67.46%.

In Table 8, AlexNet+SF-CVA continued to perform strongly, maintaining high accuracy rates of 99.16% for MNIST and 96.66% for TIMIT, demonstrating robustness to noise. AlexNet+Softmax also performed well under noisy conditions, achieving 93.06% accuracy for MNIST and 87.16% for TIMIT, though with a slight decrease compared to clean data.

Speaker identification was performed by extracting 40-, 50-, and 60-dimensional i-vector features, as shown in Tables 9 to 12. The universal background model (UBM) for the i-vector+PLDA system consists of 64 components. The total variability space (TVS) rank was set to 40, 50, and 60, extracting 40-, 50-, and 60-dimensional i-vectors for each speech signal.

Table 9 presents the performance of i-vector combined with various classifiers. I-vector + SF-CVA consistently achieved excellent accuracy, reaching 99.66% on both the MNIST and TIMIT datasets. For MNIST, i-vector + KNN achieved 99.16% accuracy. The i-vector + BiLSTM also performed well, particularly for TIMIT, which achieved 98.86% accuracy with 60-dimensional i-vectors. The i-vector + HCF demonstrated strong performance across clean and noisy environments, particularly for the TIMIT dataset. Table 10 shows the results for the Voxceleb1 dataset. The highest accuracies were 72.59% for I-vector + SVM and 85.25% for i-vector + SF-CVA.

Table 11 gives the accuracy rates of i-vectors obtained from noisy speech signals.

Table 12
Accuracy results that found using i-vectors for noisy speech signals.

	Voxceleb1		
Classifiers	40- dimensional	50-dimensional	60-dimensional
i-vector + PLDA	59.25	60.41	62.03
i-vector + KNN	54.27	55.23	57.54
i-vector + BiLSTM	58.73	59.48	60.37
i-vector + SVM	66.84	67.45	69.12
i-vector + CVA	51.58	52.12	55.89
i-vector + SF-CVA	71.40	72.71	74.78
i-vector + HCF	55.43	55.86	59.75

Table 11
Accuracy results that found using i-vectors for noisy speech signals.

	MNIST			TIMIT		
Classifiers	40-dimensional	50-dimensional	60-dimensional	40-dimensional	50-dimensional	60-dimensional
i-vector + PLDA	88.07	89.51	91.34	78.17	79.38	82.76
i-vector + KNN	94.51	95.04	96.31	72.83	72.83	74.86
i-vector + BiLSTM	87.69	90.85	93.23	85.83	87.33	90.28
i-vector + SVM	87.41	92.74	94.24	71.16	69.33	72.83
i-vector + CVA	88.92	91.19	91.85	91.67	92.23	93.46
i-vector + SF-CVA	98.33	99.33	99.33	95.66	98.33	99.16
i-vector + HCF	91.87	93.10	92.10	93.33	95.45	95.88

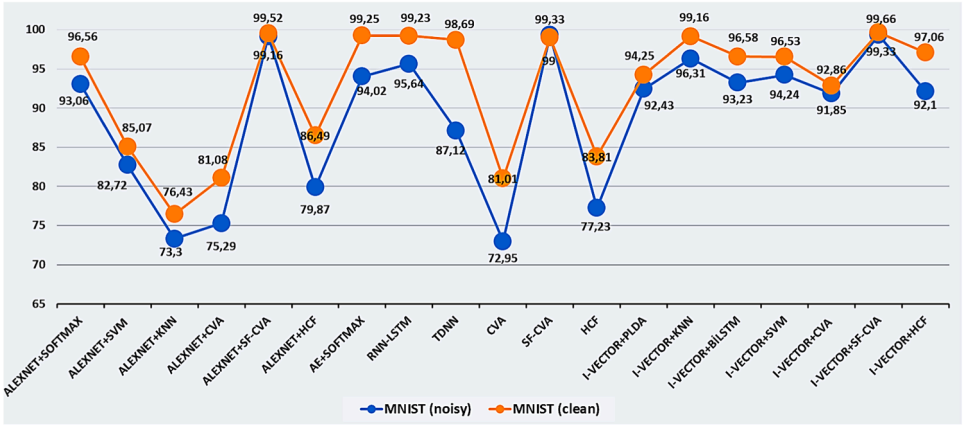


Fig. 14. The highest accuracy rates of classifiers for the MNIST database.

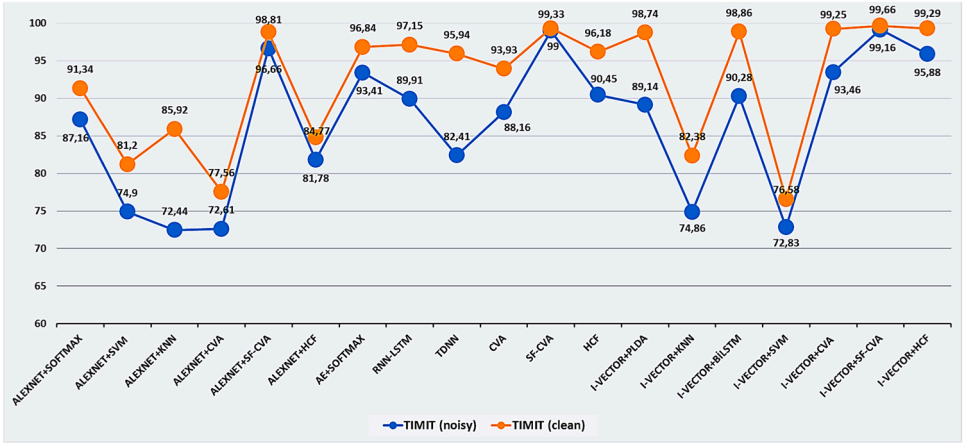


Fig. 15. The highest accuracy rates of classifiers for the TIMIT database.

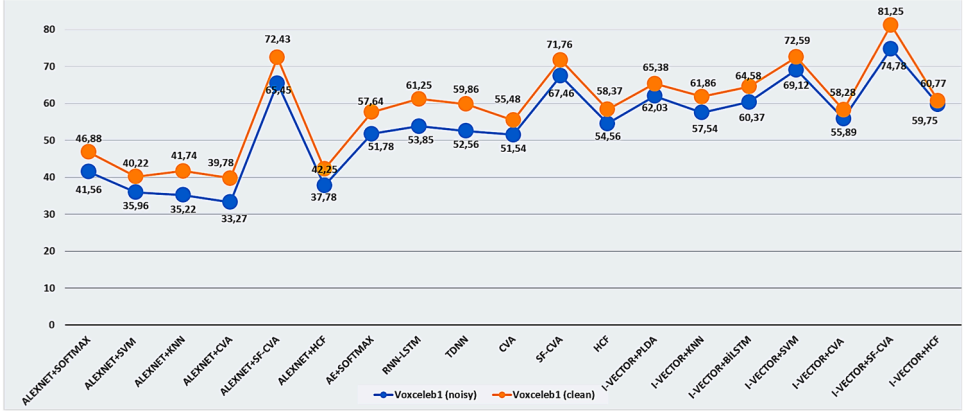


Fig. 16. The highest accuracy rates of classifiers for the Voxceleb1 database.

According to Table 11, the two best-performing classifiers for MNIST are i-vector + SF-CVA and i-vector + KNN. For the two databases, SF-CVA achieves accuracies between 98.33% and 99.66% for MNIST and 95.66% to 99.16% for TIMIT. In addition, i-vector + HCF consistently achieved over 91% accuracy for both databases. In Table 12, the highest accuracies were 69.12% for i-vector + SVM and 78.78% for i-vector + SF-CVA.

Fig. 14 indicates the highest accuracy rates obtained with the classifiers using clean and noisy speech signals for the MNIST database.

Fig. 15 and 16 show the highest accuracy rates found for the TIMIT and Voxceleb1 databases.

Table 13 provides the computational complexity and the average time per sample (TPS) of the classifiers for the MNIST, TIMIT, and Voxceleb1 databases. Due to the nature of their algorithms, the classifiers with the lowest TPS are SF-CVA, CVA, and HCF.

The results of our study are compared with those of other studies using similarly sized databases. Table 14 gives our methods' speaker identification accuracy rates and other studies in [49,55,76-91]. As can

Table 13
The values of computational time and complexity for the test.

Classifier	Average TPS (second)	Time Complexity	Explanation
Alexnet+Softmax	0.071	$O\left(n^2.d^2.f\right)$	n: The input dimension d: The kernel/filter size f: The number of filters
Alexnet+SVM	0.42	$O(n_s.d)$	n_s : The number of support vectors d: the dimensionality of the input
Alexnet+KNN	0.28	$O(n.k)$	n: the number of samples k: the number of neighbors
Alexnet+CVA	0.035	$O(k.n^2)$	n: the dimensionality of the input k: the number of classes
Alexnet+SF-CVA	0.019	$O(n^2)$	n: the dimensionality of the input
Alexnet+HCF	0.092	$O(k.n^2) + O(f)$	$O(f)$: the extra step from FLDA
AE+Softmax	0.0053	$O(n.d) + O(k)$	n: the number of frames d: the number of neurons in the hidden layer $O(k)$: Softmax
RNN-LSTM	0.122	$O\left(T\cdot\left(n\cdot h + h^2\right)\right)$	T: number of frames n: the dimensionality of the input h: the number of hidden units
TDNN	0.148	$O\left(T\cdot\sum_{i=1}^{L}\left(k_i\cdot w_i\cdot k_{i-1}\right)\right)$	T: number of frames k_i: Number of neurons in layer i. w_i: Width of the time window in layer i. k_{i-1}:Number of neurons in the previous layer.
CVA	0.0069	$O(k.n^2)$	n: the dimensionality of the input k: the number of classes
SF-CVA	0.0051	$O(n^2)$	n: the dimensionality of the input
HCF	0.0128	$O(k.n^2) + O(f)$	$O(f)$: the extra step from FLDA
i-vector+PLDA	0.028	$O(T\cdot K\cdot n) + O\left(n^2\cdot F\right)$	F: The dimensionality of the i-vectors T: The number of frames in the test utterance K: The number of components in the UBM n: The feature dimension
i-vector+KNN	0.0061	$O(n.k)$	n: the number of samples k: the number of neighbors
i-vector+BiLSTM	0.0026	$O\left(T\cdot\left(n\cdot h + h^2\right)\right)$	T: number of frames n: the dimensionality of the input h: the number of hidden units
i-vector+SVM	0.23	$O(n_s.d)$	n_s: The number of support vectors d: the dimensionality of the i-vectors
i-vector+CVA	0.00049	$O(k.n^2)$	n: the dimensionality of the input k: the number of classes
i-vector+SF-CVA	0.00029	$O(n^2)$	n: the dimensionality of the input
i-vector+HCF	0.0032	$O(k.n^2) + O(f)$	$O(f)$: the extra step from FLDA

be seen from Table 14, the accuracy rate of most of the classifiers used in our proposed study is higher than in other studies.

4. Conclusions

This study compares machine and deep learning classifiers for speaker identification, including developing novel subspace classifiers such as SF-CVA and HCF. Our findings reveal that SF-CVA consistently achieved superior accuracy across multiple datasets, particularly excelling in noisy environments, where it reached 99.66% accuracy on the MNIST dataset and high accuracy on the TIMIT dataset. In addition to SF-CVA, other classifiers also demonstrated strong performance. For instance, RNN-LSTM and AE + Softmax exhibited high accuracy rates, particularly in clean speech conditions, making them valuable alternatives in scenarios where deep learning approaches are preferred. The TDNN also performed competitively, especially on the TIMIT dataset, demonstrating its effectiveness in sequential data processing tasks. For the Voxceleb1, the highest accuracy rate was 81.25% using i-vector SF-CVA. However, other classifiers have lower accuracy rates.

SF-CVA stood out not only for its accuracy but also for its computational efficiency, enabling fast processing. Moreover, its ability to isolate noise in the difference vectors while preserving the essential features of the speaker in the common vectors further enhanced its robustness in noisy speech signals. Overall, SF-CVA emerged as a leading classifier, while the strong performance of RNN-LSTM, AE + Softmax, and TDNN emphasizes the importance of considering multiple approaches depending on the application context. The results show that subspace classifiers and advanced deep learning techniques are robust tools for speaker identification, especially when accuracy and computational efficiency are crucial.

Credit author statement

SK did the writing, reviewing, editing, and software. EG prepared references, prepared figures, and checked sentences.

Funding

No funding was received to assist with the preparation of this manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial

Table 14
Comparison of the proposed study with studies in the literature for similar-sized databases.

Reference	Method	Dataset	Feature Extraction	Highest Accuracy (%)
[49]	Extreme learning machine (ELM) and Backpropagation NN based the i-vector	TIMIT (120 speakers)	PNCC and MFCC	95.83, (20 dB noise: 92.5)
[55]	DNN	LibriSpeech (100 speakers)	MFCCT	89
[76]	BPNN	10 speakers	MFCC	92
[77]	Vector Quantization	20 speakers	Weighted LPCC	94.67
[78]	CNN, KNN, SVM	SUSAS (32 speakers), RAVDESS (24 speakers), Emirati-accented speech database (50 speakers)	MFCCs-delta and MFCCs delta-delta	CNN: 96
[79]	MKMFCC–SVM	80 speakers	Multiple Kernel Weighted Mel Frequency Cepstral Coefficient (MKMFCC)	97.5
[80]	GMM-UBM	MEPCO (50 speakers)	RASTA–MFCC	97.67
[81]	GMM	CMU (50 speakers)	MFFC	86
[82]	ANN	ELDSR (22 speakers)	Clustering based MFCC	93
[83]	GMM	CHAIN (36 speakers)	Discrete Wavelet Transform (DWT)	96.31
[84]	i-vector, vector quantization	15 speakers	MFFC	92.38
[85]	FCNN	MNIST (60 speakers)	The gammatone filter banks	98.77
[86]	i-vector with ELM approach	TIMIT (120 speakers)	PNCC and MFCC	96.67
[87]	i-vector + PLDA	TIMIT (100 speakers)	PNCC and RASTA PLP	85
[88]	Extreme learning machine (ELM)	TIMIT (124 speakers)	MFCC and PNCC	97.52 (20 dB noise: 69.98)
[89]	Wav2Vec2 +Augmentation	LibriSpeech, Voxceleb1	MFCC +CNN	99.64–99.99 (Top-1%-Top-10%)
[90]	CNN	Voxceleb1	MFCC	90 (Top-1 %)
[91]	Bidirectional LSTM	Voxceleb1	MFCC	76.88 %
Proposed Methods (for clean speech signals)	Proposed classifiers	MNIST (60 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+KNN:99.31 RNN-LSTM: 99.23 i-vector+SF-CVA:99.66
Proposed Methods (for noisy speech signals)	Proposed classifiers	MNIST (60 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+KNN:96.43 Alexnet+ SF-CVA: 99.16 i-vector+SF-CVA:99.33
Proposed Methods (clean speech signals)	Proposed classifiers	TIMIT (120 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+SF-CVA:99.66 i-vector+HCF:99.29 SF-CVA:99.33
Proposed Methods (noisy speech signals)	Proposed classifiers	TIMIT (120 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+SF-CVA: 99.16 SF-CVA: 99 Alexnet+ SF-CVA: 96.66 i-vector +HCF: 95.88
Proposed Methods (clean speech signals)	Proposed classifiers	Voxceleb1 (1251 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+SF-CVA: 81.25
Proposed Methods (noisy speech signals)	Proposed classifiers	Voxceleb1 (1251 speakers)	GTCC, MFCC, i-vectors, Alexnet	i-vector+SF-CVA:74.78

interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] A. Caranica, H. Cucu, C. Burileanu, F. Portet, M. Vacher, Speech recognition results for voice-controlled assistive applications, in: 2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), IEEE, 2017, pp. 1–8.

[2] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, H. Li, Spoofing and countermeasures for speaker verification: a survey, Speech. Commun. 66 (2015) 130–153, https://doi.org/10.1016/j.specom.2014.10.005.

[3] V. Tiwari, M.F. Hashmi, A. Keskar, N.C. Shivaprakash, Virtual home assistant for voice based controlling and scheduling with short speech speaker identification, Multimed. Tools. Appl. 79 (2020) 5243–5268, https://doi.org/10.1007/s11042-018-6358-x.

[4] R.M. Hanifa, K. Isa, S. Mohamad, A review on speaker recognition: technology and challenges, Comput. Electric. Eng. 90 (2021) 107005, https://doi.org/10.1016/j.compeleceng.2021.107005.

[5] J. Ding, J.Y. Shi, Kinect microphone array-based speech and speaker recognition for the exhibition control of humanoid robots, Comput. Electric. Eng. 62 (2017) 719–729, https://doi.org/10.1016/j.compeleceng.2015.12.010.

[6] Z. Bai, X.L. Zhang, Speaker recognition based on deep learning: an overview, Neural Netw. 140 (2021) 65–99, https://doi.org/10.1016/j.neunet.2021.03.004.

[7] S.S. Tirumala, S.R. Shahamiri, A.S. Garhwal, R. Wang, Speaker identification features extraction methods: a systematic review, Expert. Syst. Appl. 90 (2017) 250–271, https://doi.org/10.1016/j.eswa.2017.08.015.

[8] N. Shome, A. Sarkar, A.K. Ghosh, R.H. Laskar, R. Kashyap, Speaker recognition through deep learning techniques: a comprehensive review and research challenges, Period. Polytech. Electric. Eng. Comput. Sci. (2023), https://doi.org/10.3311/PPee.20971.

[9] Y. Huang, K. Tian, A. Wu, G. Zhang, Feature fusion methods research based on deep belief networks for speech emotion recognition under noise condition, J. Ambient. Intell. Humaniz. Comput. 10 (5) (2019) 1787–1798, https://doi.org/10.1007/s12652-017-0644-8.

[10] F. Ye, J. Yang, A deep neural network model for speaker identification, Appl. Sci. 11 (8) (2021) 3603, https://doi.org/10.3390/app11083603.

[11] M. Tamazin, A. Gouda, M. Khedr, Enhanced automatic speech recognition system based on enhancing power-normalized cepstral coefficients, Appl. Sci. 9 (10) (2019) 2166, https://doi.org/10.3390/app9102166.

[12] G. Sharma, K. Umapathy, S. Krishnan, Trends in audio signal feature extraction methods, Appl. Acoust. 158 (2020) 107020, https://doi.org/10.1016/j.apacoust.2019.107020.

[13] E. Bachir Tazi, Fusion approach for robust speaker identification system, Int. J. Comput. Sci. Inf. Secur. (IJCSIS) 15 (8) (2017).

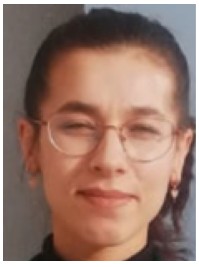
[14] A.A. Alashban, M.A. Qamhan, A.H. Meftah, Y.A. Alotaibi, Spoken language identification system using convolutional recurrent neural network, Appl. Sci. 12 (18) (2022) 9181, https://doi.org/10.3390/app12189181.

[15] W. Li, T. Fu, J. Zhu, An improved i-vector extraction algorithm for speaker verification, EURASIP. J. Audio Speech. Music. Process. 2015 (2015) 1–9, https://doi.org/10.1186/s13636-015-0061-x.

[16] M. Li, S. Narayanan, Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification, Comput. Speech. Lang. 28 (4) (2014) 940–958, https://doi.org/10.1016/j.csl.2014.02.004.
[17] P. Verma, P.K. Das, i-Vectors in speech processing applications: a survey, Int. J. Speech. Technol. 18 (2015) 529–546, https://doi.org/10.1007/s10772-015-9295-3.
[18] Z. Wu, Z. Cao, Improved MFCC-based feature for robust speaker identification, Tsinghua Sci. Technol. 10 (2) (2005) 158–161, https://doi.org/10.1016/S1007-0214(05)70048-1.
[19] C. Xie, X. Cao, L. He, Algorithm of abnormal audio recognition based on improved MFCC, Procedia Eng. 29 (2012) 731–737, https://doi.org/10.1016/j.proeng.2012.01.032.
[20] D. Salvati, C. Drioli, G.L. Foresti, A late fusion deep neural network for robust speaker identification using raw waveforms and gammatone cepstral coefficients, Expert. Syst. Appl. 222 (2023) 119750, https://doi.org/10.1016/j.eswa.2023.119750.
[21] N.M. Almarshady, A.A. Alashban, Y.A. Alotaibi, Analysis and investigation of speaker identification problems using deep learning networks and the YOHO english speech dataset, Appl. Sci. 13 (17) (2023) 9567, https://doi.org/10.3390/app13179567.
[22] M. Sarma, K.K. Sarma, Vowel phoneme segmentation for speaker identification using an ANN-based framework, J. Intell. Syst. 22 (2013) 111–130, https://doi.org/10.1515/jisys-2012-0050.
[23] S. Sekkate, M. Khalil, A. Adib, Speaker identification for OFDM-based aeronautical communication system, Circuits. Syst. Signal. Process. 38 (8) (2019) 3743–3761, https://doi.org/10.1007/s00034-019-01026-z.
[24] P.K. Ajmera, D.V. Jadhav, R.S. Holambe, Text-independent speaker identification using Radon and discrete cosine transforms based features from speech spectrogram, Pattern. Recognit. 44 (10–11) (2011) 2749–2759, https://doi.org/10.1016/j.patcog.2011.04.009.
[25] J. Medikonda, H. Madasu, Higher order information set based features for text-independent speaker identification, Int. J. Speech. Technol. 21 (3) (2018) 451–461, https://doi.org/10.1007/s10772-017-9472-7.
[26] T.B. Mokgonyane, T.J. Sefara, M.J. Manamela, T.I. Modipa, The effects of data size on text-independent automatic speaker identification system, in: 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), IEEE, 2019, pp. 1–6.
[27] C. Zhang, K. Koishida, J.H. Hansen, Text-independent speaker verification based on triplet convolutional neural network embeddings, IEEE/ACM Trans. Audio, Speech Lang. Process. (TASLP) 26 (2018) 1633–1644.
[28] K.A. Abdalmalak, A. Gallardo-Antolín, Enhancement of a text-independent speaker verification system by using feature combination and parallel structure classifiers, Neural Comput. Appl. 29 (3) (2018) 637–651, https://doi.org/10.1007/s00521-016-2470-x.
[29] S. Keser, Improvement of face recognition performance using a new hybrid subspace classifier, Signal. Image Video Process. 17 (5) (2023) 2511–2520, https://doi.org/10.1007/s11760-022-02468-w.
[30] S. Gunal, R. Edizkan, Subspace based feature selection for pattern recognition, Inf. Sci. (Ny) 178 (19) (2008) 3716–3726, https://doi.org/10.1016/j.ins.2008.06.001.
[31] M.B. Gülmezoğlu, V. Dzhafarov, R. Edizkan, A. Barkana, The common vector approach and its comparison with other subspace methods in case of sufficient data, Comput. Speech. Lang. 21 (2) (2007) 266–281, https://doi.org/10.1016/j.csl.2006.06.002.
[32] R. Jahangir, Y.W. Teh, H.F. Nweke, G. Mujtaba, M.A. Al-Garadi, I. Ali, Speaker identification through artificial intelligence techniques: a comprehensive review and research challenges, Expert. Syst. Appl. 171 (2021) 114591, https://doi.org/10.1016/j.eswa.2021.114591.
[33] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEe Trans. Audio Speech. Lang. Process. 19 (4) (2010) 788–798.
[34] P.K. Nayana, D. Mathew, A. Thomas, Comparison of text independent speaker identification systems using GMM and i-vector methods, Procedia Comput. Sci. 115 (2017) 47–54, https://doi.org/10.1016/j.procs.2017.09.075.
[35] S.J. Abdallah, I.M. Osman, M.E. Mustafa, Text-independent speaker identification using hidden Markov model, World Comput. Sci. Inf. Technol. J. (WCSIT) 2 (6) (2012) 203–208.
[36] A.A. Mallouh, Z. Qawaqneh, B.D. Barkana, New transformed features generated by deep bottleneck extractor and a GMM–UBM classifier for speaker age and gender classification, Neural Comput. Appl. 30 (2018) 2581–2593, https://doi.org/10.1007/s00521-017-2848-4.
[37] M. Jayanth, B.R. Reddy, Speaker Identification based on GFCC using GMM-UBM, Int. J. Eng. Sci. Invent. 5 (5) (2016) 62–65.
[38] Đ.T. Grozdić, S.T. Jovičić, Whispered speech recognition using deep denoising autoencoder and inverse filtering, IEEE/ACM. Trans. Audio Speech. Lang. Process. 25 (12) (2017) 2313–2322, https://doi.org/10.1109/TASLP.2017.2738559.
[39] A. Srinivasan, Speaker identification and verification using vector quantization and mel frequency cepstral coefficients, Res. J. Appl. Sci., Eng. Technol. 4 (1) (2012) 33–40.
[40] N. Almaadeed, A. Aggoun, A. Amira, Speaker identification using multimodal neural networks and wavelet analysis, IET. Biom. 4 (1) (2015) 18–28, https://doi.org/10.1049/iet-bmt.2014.0011.
[41] V.L. Lajish, S.R. Kumar, P. Vivek, Speaker identification using a nonlinear speech model and ANN, Int. J. Adv. Inf. Technol. 2 (5) (2012) 15.
[42] G. Nijhawan, M.K. Soni, Speaker recognition using support vector machine, Int. J. Comput. Appl. 87 (2) (2014).
[43] S. Sadıç, M. Gülmezoğlu, Common vector approach and its combination with GMM for text-independent speaker recognition, Expert. Syst. Appl. 38 (9) (2011) 11394–11400, https://doi.org/10.1016/j.eswa.2011.03.009.
[44] S. Bunrit, T. Inkian, N. Kerdprasop, K. Kerdprasop, Text-independent speaker identification using deep learning model of convolution neural network, Int. J. Mach. Learn. Comput. 9 (2) (2019) 143–148, https://doi.org/10.18178/ijmlc.2019.9.2.778.
[45] Y. Lukic, C. Vogt, O. Dürr, T. Stadelmann, Speaker identification and clustering using convolutional neural networks, in: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2016, pp. 1–6.
[46] A.B. Nassif, I. Shahin, S. Hamsa, N. Nemmour, K. Hirose, CASA-based speaker identification using cascaded GMM-CNN classifier in noisy and emotional talking conditions, Appl. Soft. Comput. 103 (2021) 107141, https://doi.org/10.1016/j.asoc.2021.107141.
[47] Z. Liu, Z. Wu, T. Li, J. Li, C. Shen, GMM and CNN hybrid method for short utterance speaker recognition, IEEe Trans. Industr. Inform. 14 (7) (2018) 3244–3252, https://doi.org/10.1109/TII.2018.2799928.
[48] R. Djemili, M. Bedda, H. Bourouba, A hybrid gmm/svm system for text independent speaker identification, Int. J. Electric. Comput. Eng. 1 (4) (2007) 721–727.
[49] M.T. Al-Kaltakchi, R.R.O. Al-Nima, M.A. Abdullah, Comparisons of extreme learning machine and backpropagation-based i-vector approach for speaker identification, Turk. J. Electric. Eng. Comput. Sci. 28 (3) (2020) 1236–1245.
[50] L. Xu, Z. Yang, X. Shao, Dictionary design in subspace model for speaker identification, Int. J. Speech. Technol. 18 (2015) 177–186, https://doi.org/10.1007/s10772-014-9258-0.
[51] M.K. Singh, A text independent speaker identification system using ANN, RNN, and CNN classification technique, Multimed. Tools. Appl. 83 (16) (2024) 48105–48117.
[52] E. Seke, K. Özkan, A new speech signal denoising algorithm using common vector approach, Int. J. Speech. Technol. 21 (2018) 659–670.
[53] L. Calz` a, G. Gagliardi, R.R. Favretti, F. Tamburini, Linguistic features and automatic classifiers for identifying mild cognitive impairment and dementia, Comput. Speech. Lang. 65 (2020). Article 101113.
[54] J. Fierrez, A. Morales, R. Vera-Rodriguez, D. Camacho, Multiple classifiers in biometrics. Part 1: fundamentals and review, Inf. Fus. 44 (2018) 57–64.
[55] R. Jahangir, Y.W. TEh, N.A. Memon, G. Mujtaba, M. Zareei, U. Ishtiaq, I. Ali, Text-independent speaker identification through feature fusion and deep neural network, IEEe Access. 8 (2020) 32187–32202.
[56] N.N. An, N.Q. Thanh, Y. Liu, Deep CNNs with self-attention for speaker identification, IEEe Access. (2019).
[57] D. Byrd, Preliminary results on speaker-dependent variation in the TIMIT database, J. Acoust. Soc. Am. 92 (1) (1992) 593–596.
[58] Si, S., Wang, J., Sun, H., Wu, J., Zhang, C., Qu, X., … & Xiao, J. (2021). Variational information bottleneck for effective low-resource audio classification. arXiv preprint arXiv:2107.04803.
[59] E. Tsalera, A. Papadakis, M. Samarakou, Comparison of pre-trained CNNs for audio classification using transfer learning, J. Sens. Actuat. Netw. 10 (4) (2021) 72, https://doi.org/10.3390/jsan10040072.
[60] S.M. Kulkarni, G. Sundari, Comparative analysis of performance of deep cnn based framework for brain mri classification using transfer learning, J. Eng. Sci. Technol. 16 (4) (2021) 2901–2917.
[61] Desplanques, B., Thienpondt, J., Demuynck, K. (2020). Ecapa-tdnn: emphasized channel attention, propagation and aggregation in tdnn based speaker verification. arXiv preprint arXiv:2005.07143.
[62] Y.Q. Yu, W.J. Li, Densely connected time delay neural network for speaker verification, in: INTERSPEECH, 2020, pp. 921–925.
[63] S. Hu, X. Xie, S. Liu, J. Yu, Z. Ye, M. Geng, H. Meng, Bayesian learning of LF-MMI trained time delay neural networks for speech recognition, IEEE/ACM. Trans. Audio Speech. Lang. Process. 29 (2021) 1514–1529, https://doi.org/10.1109/TASLP.2021.3069080.
[64] H.J. Kim, K.S. Shin, A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets, Appl. Soft. Comput. 7 (2) (2007) 569–576, https://doi.org/10.1016/j.asoc.2006.03.004.
[65] J.H. Wang, Y.T. Lai, T.C. Tai, P.T. Le, T. Pham, Z.Y. Wang, P.C. Chang, Target speaker extraction using attention-enhanced temporal convolutional network, Electronics. (Basel) 13 (2) (2024) 307, https://doi.org/10.3390/electronics13020307.
[66] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[67] D. Neil, M. Pfeiffer, S.C. Liu, Phased lstm: accelerating recurrent network training for long or event-based sequences, Adv. Neural Inf. Process. Syst. 29 (2016).
[68] F. Curreri, L. Patanè, M.G. Xibilia, RNN-and LSTM-based soft sensors transferability for an industrial process, Sensors 21 (3) (2021) 823, https://doi.org/10.3390/s21030823.
[69] H. Choi, M. Kim, G. Lee, W. Kim, Unsupervised learning approach for network intrusion detection system using autoencoders, J. Supercomput. 75 (2019) 5597–5621, https://doi.org/10.1007/s11227-019-02805-w.
[70] N.S. Ibrahim, D.A. Ramli, I-vector extraction for speaker recognition based on dimensionality reduction, Procedia Comput. Sci. 126 (2018) 1534–1540, https://doi.org/10.1016/j.procs.2018.08.126.
[71] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, N. Dehak, State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and speakers in the wild evaluations, Comput. Speech. Lang. 60 (2020) 101026, https://doi.org/10.1016/j.csl.2019.101026.

[72] Dehak, Najim, Réda Dehak, James R. Glass, Douglas A. Reynolds and Patrick Kenny. "Cosine Similarity Scoring without Score Normalization Techniques." Odyssey (2010).
[73] E.Y. Boateng, J. Otoo, D.A. Abaye, Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: a review, J. Data Anal. Inf. Process. 8 (4) (2020) 341–357, https://doi.org/10.4236/jdaip.2020.84020.
[74] S. Keser, R. Edizkan, Phonem-based isolated Turkish word recognition with subspace classifier, in: 2009 IEEE 17th Signal Processing and Communications Applications Conference, IEEE, 2009, pp. 93–96.
[75] M. Awad, R. Khanna, M. Awad, R. Khanna, Support vector machines for classification, Effic. Learn. Mach.: Theories, Concepts, Appl. Engineers Syst. Designers (2015) 39–66.
[76] S.S. Wali, S.M. Hatture, S. Nandyal, MFCC based text-dependent speaker identification using BPNN, Int. J. Signal Process. Syst. 3 (1) (2014) 30–34.
[77] L. Zhu, Q. Yang, Speaker recognition system based on weighted feature parameter, Phys. Procedia 25 (2012) 1515–1522, https://doi.org/10.1016/j.phpro.2012.03.270. Jan.
[78] I. Shahin, A.B. Nassif, N. Hindawi, Speaker identification in stressful talking environments based on convolutional neural network, Int. J. Speech. Technol. 24 (2021) 1055–1066, https://doi.org/10.1007/s10772-021-09869-1.
[79] O.S. Faragallah, Robust noise MKMFCC–SVM automatic speaker identification, Int. J. Speech Technol. 21 (2) (2018) 185–192, https://doi.org/10.1007/s10772-018-9494-9. Jun.
[80] S. Selva Nidhyananthan, R. Shantha Selva Kumari, T. Senthur Selvi, Noise robust speaker identification using RASTA–MFCC feature with quadrilateral filter bank structure, Wirel. Pers. Commun. 91 (2016) 1321–1333, https://doi.org/10.1007/s11277-016-3530-3.
[81] J.C. Liu, F.Y. Leu, G.L. Lin, H. Susanto, An MFCC-based text-independent speaker identification system for access control, Concurr. Computat.: Pract. Exp. 30 (2) (2018) e4255.
[82] M. Soleymanpour, H. Marvi, Text-independent speaker identification based on selection of the most similar feature vectors, Int. J. Speech Technol. 20 (1) (2017) 99–108, https://doi.org/10.1007/s10772-016-9385-x. Mar.
[83] K.A. VD, Wavelets for speaker recognition using GMM classifier, Int. J. Adv. Signal Image Sci. 3 (1) (2017) 13–18, https://doi.org/10.29284/ijasis.3.1.2017.13-18.
[84] S.S. Bharali, S.K. Kalita, Speaker identification using vector quantization and I-vector with reference to Assamese language, in: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, 2017, pp. 164–168, https://doi.org/10.1109/WiSPNET.2017.8299740.
[85] X.Y. Cai, S.W. Ko, Development of parametric filter banks for sound feature extraction, IEEe Access. (2023), https://doi.org/10.1109/ACCESS.2023.3321798.
[86] M.T. Al-Kaltakchi, M.A. Abdullah, W.L. Woo, S.S. Dlay, Combined i-vector and extreme learning machine approach for robust speaker identification and evaluation with SITW 2016, NIST 2008, TIMIT databases, Circuits. Syst. Signal. Process. 40 (10) (2021) 4903–4923.
[87] P.K. Nayana, D. Mathew, A. Thomas, Comparison of text independent speaker identification systems using GMM and i-vector methods, Procedia Comput. Sci. 115 (2017) 47–54.
[88] B. KP, ELM speaker identification for limited dataset using multitaper based MFCC and PNCC features with fusion score, Multimed. Tools. Appl. 79 (39) (2020) 28859–28883.
[89] O.H. Anidjar, R. Marbel, R. Yozevitch, Harnessing the power of Wav2Vec2 and CNNs for Robust Speaker Identification on the VoxCeleb and LibriSpeech Datasets, Expert. Syst. Appl. 255 (2024) 124671.
[90] N.N. An, N.Q. Thanh, Y. Liu, Deep CNNs with self-attention for speaker identification, IEEe Access. 7 (2019) 85327–85337.
[91] M.K. Nammous, K. Saeed, P. Kobojek, Using a small amount of text-independent speech data for a BiLSTM large-scale speaker identification approach, J. King Saud Univ. -Comput. Inf. Sci. 34 (3) (2022) 764–770.

Serkan Keser received M.S. and Ph.D. degrees from Eskişehir Osmangazi University in Electrical-Electronics Engineering in 2008 and 2018, respectively. He is working as an Asst. Prof. at the Department of Electrical and Electronics Engineering, Kırşehir Ahi Evran University. His current research interests are signal and systems, digital signal processing, speech and image recognition, signal coding and artificial neural networks.



Esra GEZER received her master's degree from Kırşehir Ahi Evran University, Department of Advanced Technologies in 2023. She continues her PhD program at Uludağ University, Department of Electrical and Electronics Engineering.