Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

An Automated PSO-CFNN Approach for Speaker Classification and Identification Using a Deep Learning Technique

T. S. Mullaivendan[1], R. Thiruvengatanadhan[2], P. Dhanalakshmi[3]

[1]Research Scholar, Department of Computer and Information Science, Annamalai University, tsmullai@outlook.com
[2]Assistant Professor, Department of Computer Science and Engineering, Annamalai University, thiruvengatanadhan01@gmail.com
[3]Professor, Department of Computer Science and Engineering, Annamalai University, abidhana01@gmail.com

Speaker identification plays a crucial role in numerous applications such as security systems, voice-controlled assistants, forensics, and automated customer services systems. The abilities to accurately identify individuals based on their voices enhance security and the user experiences in these applications making speaker identification an essential part of modern technology. Traditional methods often struggle with background noise and require significant pre-processing, which can limit their effectiveness in the real world. To address these challenges, this study introduced a pioneering PSO-CFNN framework designed to authenticate speaker identification tasks effectively. The proposed framework incorporates several advanced techniques to improve accuracy and robustness. Initially, the wavelet denoising technique is applied to eliminate noise interferences in the audio signals, enhancing the quality of the input data. Then, spectrograms are generated from the denoised audio signal and used as inputs for VGGVox architecture, a deep learning model known for its robust feature extraction capabilities in speaker recognition tasks. Following feature extraction, the Particle Swarm Optimization Algorithm is employed to optimize hyperparameters of Convolutional Fuzzy Neural Networks (CFNN). The optimization step ensures CFNN finely tunes to achieve the best possible performance. In the final stage, CFNN architecture is utilized as a classifier to facilitate automatic speech recognition (ASR). The integration of fuzzy logic within CFNN allows for handling ambiguity and uncertainty in data, further enhancing model robustness.

Keywords: Speaker identification, Deep learning, Particle swarm optimization algorithm, VGGVox, Spectrograms.

1. Introduction

Speaker identification and classification are crucial tasks in many domains such, as security systems, voice-controlling assistants forensics, and customer service automation. These applications require identifying individuals based on their voices which enhances security user experiences and operational efficiencies [1]. Traditional methods for speaker identifications often face challenges such as background noise variability in speech patterns, and the need for extensive pre-processing. To address these issues, Deep Learning and Optimization Algorithms have emerged as powerful tools [2].

Deep learning revolutionized the field of speaker identifiers by providing a model that learns complex patterns and features from raw audio data [3]. Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNNs) have demonstrated exceptional performance in extracting high-level features from an audio signal [4]. These models can handle the variabilities in speech patterns and adapt to different speakers, making them highly effective for speaker identification tasks [5]. Deep learning models automatically extract relevant features from audio data eliminating the need for manual feature engineering. These models can learn to distinguish between a speaker's voice and a background noise improving accuracy. Deep learning models can train on large datasets enabling them to generalize well to new unseen data [6]. Optimization algorithms play a crucial role in enhancing the performance of deep learning models. This algorithm is used to fine-tune hyperparameters, assuring those models achieve optimal performances [7]. Algorithms help to select the best set of hyperparameters, like learning rate, and batch size, along with the number of layers that are critical to model performance. These algorithms accelerate the training process [8]. It guides the model to an optimal solution that is more efficient. Techniques such as fuzzy logic, integrating with optimization algorithms, can manage uncertainty and ambiguity in data, next improving model robustness ability [9].

This paper presents a novel approach to Automated Speaker Identification, which is termed the Convolutional Fuzzy Neural Network enhanced by Particle Swarm Optimization Algorithm (PSO-CFNN). In the PSO-CFNN framework, the Wavelet Denoising method is used strategically to eliminate noise from audio signals effectively. After removing noise, spectrogram data feeds into a VGGVox model—a deep convolutional neural network foundation for extracting key features. The PSO algorithm optimizes hyperparameters of VGGVox models to boost performance. For accurate automatic speech recognition and classification, the CFNN framework is robust and offers dynamic analysis tools. The performances and reliabilities of PSO-CFNN models have been validated through various experimental setups, which demonstrate their effectiveness toward meeting objectives.

2. Related Works

In [10], the author proposed a hybrid feature extraction model for speaker recognition utilization with a Deep Believes Network. The processes involved in converting audio scripts into spectrograms that are represented as two-dimensional matrices too capture both time and frequency dimensions. To reduce dimensions, PCA-based techniques were applied, transforming the frequency data into lower-dimension spaces. Latent features from audio signals were extracted using an MFCC and combined with features derived from the unsupervised deep belief network.

In the study [11], some text-independent speaking identification systems based on one Convoluted Network (CNN) were introduced. In such a system, every audio signal sample is converted into spectrogram images, which then are inputted as grayscale images to the network. The CNM model was trained from scratch using three convolutional layers and two pool layers. The performance of this proposed method compared with the MFCC method and the CNN approach applied directly to the signal wave.

The author [12] develops a speaker identification model by incorporating a self-attention layer into two well-known CNN architectures Visual Geometry Group (VGG) nets and Residual Neural Networks (ResNets). By utilizing a structured self-attention layer with many attention hops, the proposed approach can manage variable, and length speech signal segments this model also learned the characteristics from speakers of various aspects of the input sequences including MFCC, FBanks, and spectrogram data. One text-independent speaker recognition system that is capable of operating in noisy, and reverberant conditions was proposed by the author [13]. The system utilizes MFCCs, spectrums, and log-spectrum features extracted from the input speech signal. An LSTM-based neuron network be employed as a classifier to perform speaker-recognition tasks.

A lightweight convolutional Neural Network (CNN) architecture was proposed in [14] for extracting deep features from speech spectrograms. The speech signals were converted into segments of similar lengths using a short-term Fourier transform algorithm. To compute the Fourier spectra, the fast Fourier transform method is employed. A multiples-feature bunch methodology, leverages three advanced feature extraction techniques: a Mel Spectrogram, Mel Frequency Cepstral Coefficients, and Crossing Rate—have been proposed in [15]. Deep learning models which include CNNs, EfficientNet, and MobileNet along with traditional classifiers such as SVMs and perceptron, were used in various ways to train each feature separately, as well as in combinations of two or three features. The performances of each configuration were assessed on accuracy and testing times. The study referenced in [16] explored the application of speaker identifications in a courtroom setting. It investigates if a judge's ability to identify a speaker is more or less accurate compared to the results produced by the forensic voice comparisons system. In [17], the author introduces a novel Convolutional Neural Network (CNN) architecture called the VGG-13 for dependent speaker identification systems. All short segments of audio samples were converted into a log-mel spectrogram and a data augment technique was applied to the segment. These process segments were fed to the VGG-13 architecture comprising 10 convolutional layers. The Rectified Linear Unit (ReLU) acts function was used in all layers. To optimize the architecture, the number of filters in the existing VGG-13 architecture was reduced, significantly reducing training times and memory consumption.

A study in [18] examines both times-domain and frequency-domain futures to enhance the robustness of a speaker's identification in environments with noises and reverberations. The work proposed in [19] addresses the challenge of identifying speakers in environments that are noisy, stressful, and emotionally charged. The study presented in [20] proposes an ensemble model that integrates a Convolutional Neuronal Network (CNN), Long Short-Term

Memory (LSTM), and the Gated Recurrent Unit (GRU) for speech emotion recognition. This research examined the effectiveness of using an ensemble's deep learning modeling, which combines various deep learning structures, to enhance the understanding of emotions on speech signals.

3. Proposed Work

This paper introduces a novel PSO-CFNN framework for Speaker Identification design. This innovative model utilized a dataset comprising 7500 instances collected from five different speakers, where the audio was recorded at a 16 kHz sampling rate. The dataset is split into training and validation sets—70% is used for training, and the remaining 30% is left for validation. Speech segments within this dataset range between 3 to 5 seconds each. Initially, the PSO-CFNN framework applies the Wavelet Denoising (WD) method to remove noise from the audio signal. Next, spectrograms serve as input to the VGGVox model. Then comes the Particle Swarm Optimization (PSO) algorithm; it fine-tunes the hyperparameters of the CFNN model. In its final phase, CFNN models are utilized to classify tasks related to automatic speech recognition (ASR). Fig. 1 provides an illustration detailing the comprehensive methodology adopted by the PSO-CFNN strategy.

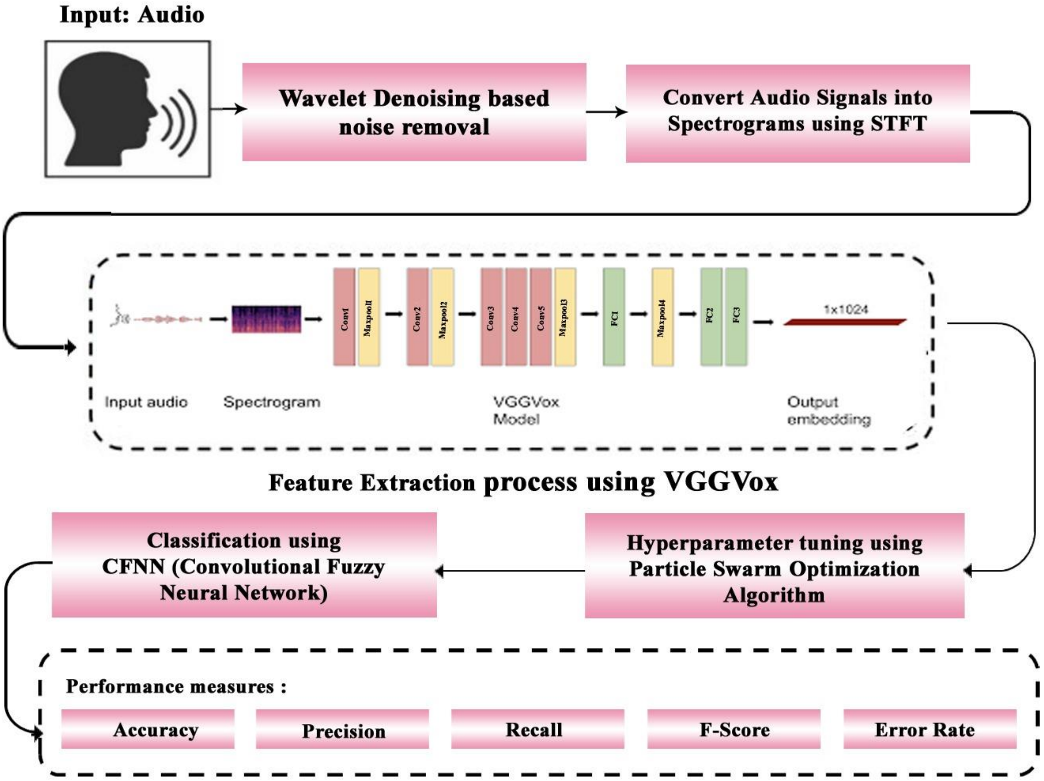
Input: Audio
Wavelet Denoising based noise removal
Convert Audio Signals into Spectrograms using STFT
Input audio
Spectrogram
Conv1
Maxpool1
Conv2
Maxpool2
Conv3
Conv4
Conv5
Maxpool3
FC1
Maxpool4
FC2
FC3
1x1024
VGGVox Model
Output embedding
Feature Extraction process using VGGVox
Hyperparameter tuning using Particle Swarm Optimization Algorithm
Classification using CFNN (Convolutional Fuzzy Neural Network)
Performance measures :
Accuracy
Precision
Recall
F-Score
Error Rate

Fig. 1. Overall architecture of PSO-CFNN Approach

3.1. Wavelet Denoising based noise removal

Wavelet Denoising (WD) is often used for noise removal in audio signals, including in speaker-identifying tasks. It works by decomposition the signal into different frequency bands, utilizing wavelet transform; applying thresholding for attenuating noise in each band, and then reconstructing the denoised signal [21]. Wavelet transform decomposes an input signal into different frequency components at various scales. It provides a time-frequency representation of the signal, letting us analyze both time-localized and frequency-localized features. The wavelet transform of a signal x(t) can be expressed by Eq. (1):

$$\mathrm{W(a,b)} = \int_{-\infty}^{\infty} \mathrm{x(t)}\Psi_{\mathrm{a,b}}^{*}(\mathrm{t})\ \mathrm{dt} \qquad (1)$$

Where $\Psi_{\mathrm{a,b}}^{*}(\mathrm{t})$ is the wavelet function scaled by a and shifted by b, and W(a, b) represents the wavelet co-efficients. After decomposing the signals into wavelet coefficients, thresholding is applied to those co-efficients to remove or reduce noises. The primary concept is that the signal co-efficients are usually bigger than noise coefficients, so it's are distinguishable from each other based on their size. There are various methods for thresholds, such as hard thresholds and soft thresholds. Soft thresholding, for example, sets coefficients that are below a specific threshold to zero- while damping the rest of the coefficients.

Mathematically, soft thresholding can be defined as shown in Eq. (2):

$$\mathrm{T}_{\lambda}(\mathrm{w}) = \mathrm{sign(W)} \cdot \max\left(|\mathrm{W}| - \lambda, 0\right) \qquad (2)$$

Where $\mathrm{T}_{\lambda}(\mathrm{w})$ is the denoised wavelet coefficients, sign (.) returns the sign of W, |W| is the absolute value of W, and λ is the threshold parameter. After thresholding, the denoised wavelet coefficients are reconstructed to obtain the denoised signal. This is achieved by the application of inverse wavelet transform for the denoised coefficients.

The inverse wavelet transformation of the denoised coefficients, $\mathrm{T}_{\lambda}(\mathrm{w})$ yields the denoised signal $\hat{\mathrm{x}}(\mathrm{t})$ is as shown in Eq. (3):

$$\hat{\mathrm{x}}(\mathrm{t}) = \int_{-\infty}^{\infty} \mathrm{T}_{\lambda}(\mathrm{w})\Psi_{\mathrm{a,b}}(\mathrm{t})\ \mathrm{da} \qquad (3)$$

Wavelet denoising effectively removes noises from sound signals while preserving the important signal features. By adaptive thresholds of wavelet coefficients, it can attenuate noises without significant distortions of the underneath signals. This makes it especially suitable for speaker identification tasks where the preservation of the Integrity of the speech signals is essential for adequate analyses and classifications.

3.2. Convert audio signals into spectrograms using STFT

After the noise has been eliminated from the audio signals, the spectrograms are converted by using a Short-Time Fourier Transform (STFT). Following the denoising process, those denoised audio signals need segmenting into Short overflowing frames [22]. These frames ought to be tiny enough to grasp the local propensities of the signals but also provide enough frequency resolutions. Then, the windowing function is applied to reduce the spectral leakage. Now, compute the Short-Time Fourier Transform (STFT) for every frame using those windowed segments; STFT computes the Fourier transform of every segment and encapsulates it into the time-frequency domains.

The STFT, of a Signals $x(t)$ uses a window function, $\omega(t)$, at time t and Frequency, ω, it is given by in Eq. (4):

$$X(\omega, t) = \int_{-\infty}^{\infty} x(\tau)\omega(\tau - t)e^{-j\omega T}\, d\tau \quad (4)$$

Where $X\ (\omega, t)$ represents the STFT magnitude at the frequency ω and time t, $x(\tau)$ is the original signal, and $\omega(\tau -\ t)$ is the window function centered at time t.

The magnitude of STFT coefficients stands for the energy distribution across various frequencies and time frames. Square the magnitude of each STFT coefficient to snag the power spectrum. Stack the power spectrum of all the frames over time, to create the spectrogram-where the x-axis shows time. The Y-axis shows frequency, and the color intensity represents the magnitude of the power spectrum. When converting denoised audio signals into spectrograms utilizing STFT, we obtain a time-frequency representation of the signal that captures, both temporal and spectral information. This spectrogram representation is often utilized as an input feature for speaker identification tasks; it Allows machine learning algorithms to analyze the spectral characteristic of the speech signals, across different frequency bands and time frames.

3.3. Feature Extraction using VGGVox

After obtaining the spectrograms from the denoised audio signals; feature extraction using VGGVox involves processing these spectrograms through a convolution neural network- CNN architecture [23]. VGGVox is a specific CNN architecture designed for speaker identification tasks, typically consisting of convolution layers followed by fully connected layers as depicted in Fig. 2. The inputs to VGGVox are singularly a spectrogram derived from denoised audio signals, representing the times-varying frequency content of the audio signal. They are often computed using Short-Time Fourier Transforms (STFT) or similar methodologies. The input spectrogram, get passes through a series of convolutional layers in VGGVox. Each convolutional layer applies a group of learnable filters. These filters, also known as kernels do convolutions, to extract features from the input spectrogram. The depth of these convolution layers deepens gradually, letting the network learn more complex and abstract features. After each convolutional operation, an activation function gets applied on an element-by-element basis, to toss in some non-linear feels into the network. Some typical activation functions embarking like the Rectified Linear Unit, ReLU, up-jumps sparsity, and speed the convergence. Meanwhile, Max-pooling layers are often inserted between the convolutional layers to squash that spatial enlightenment of the features maps while preserving important information.

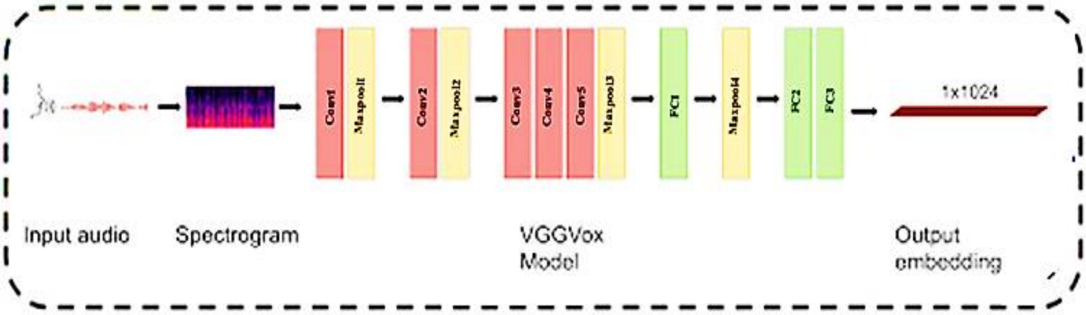

Fig.2. Feature extraction using VGGVox

Max-pooling extracts the maximum value found across all areas of the features map. These pooling layers, aid in making the learned features much more invariant to small translations and distortions in the input spectrogram. The output feature maps of the convolutional layers get flat into a one-dimensional feature vector. This Vector has the high-level representations from the input spectrogram learned in the convolutional layers. Flattening collapses the space dimensions of those feature maps into a single vector, ready to be processed by fully connected layers. The flattened feature vectors are being passed through one or more fully connected layers in VGGVox. This layer learns to map the extracted feature to speaker identities. Fully connected layers introduce complex interaction between features learned by convolutional layers. They perform the final mapping to speaker identities. The output from the last fully connected layer was getting passed through a softmax activation function; which normalizes output scores into probability distributions over speech identities. The softmax layer assigns probabilities to each speaker's identity to indicate the likelihood that the input spectrogram belongs to that speaker. The final result of VGGVox is the probability distributions over Speaker Identities. However, the almost last layer- right before the softmax- could also be used like the extracted features for downstream tasks like similarity comparisons or clustering. By processing, and inputting spectrograms through convolutional layers, pooling layers, and fully connected layers, VGGVox learning extracts top-level representations of the discriminative audio signal for speaker identification. The learned features capture local and global patterns in the spectrograms, enabling accurate speaker recognition in several conditions.

3.4. Hyperparameter tuning using PSO

Tuning the hyperparameter of a Convolutional Fuzzy Neural Network (CFNN) by utilizing Particle Swarm Optimization (PSO) involves optimizing parameters such as filter sizes, numbers of filters, learning rates, and fuzzy membership functionaries to max the performance on specific tasks of the CFNN. Particle Swarm Optimization (PSO) is a naturally inspired optimization algorithm that is used for finding optimal solutions to problems by simulating the social behaviors of bird flocks or fish schools [24]. In PSO, a group of potential solutions, what's known as particles, is moving around in a search space to find the best solution based on their own experiences and the neighbor’s experiences.

Particle Swarm Optimization (PSO) consists of several phases or core components that collectively enable the optimization process. These phases describe the sequences of action and interactions between particles within a Swarm. The main phases of the PSO included initialization, movement updates, fitness evaluations, and termination.

- Initialization Phase

PSO starts by initializing a population of particles into the search space, each particle represents a potential solution to an optimization problem. Next, Particle positions and velocities are randomly initialized in within the specified bounds of the search space. Then, each particle's personal best position (pbest) was initially set to its current position: and the global best position (gbest) was initialized to the best positions among all particles.

- Movement and Position Update Phase

In every iteration, particles update their velocity based on their current velocity, cognitive component (being influenced by personal Best), and social component (being under the influence of a global best) as shown in Eq. (5):

$$\mathbf{v_i(t+1) = wv_i(t) + c_1r_1\left(pbest_i - X_i\,(t)\right) + c_2\,r_2\,\left(gbest - X_i(t)\right)} \quad \mathbf{(5)}$$

Where:

○ $\mathbf{v_i(t)}$ is the velocity of the particle ‘i’ at iteration ‘t’,

○ $\mathbf{X_i(t)}$ is the position of a particle ‘i’ at iteration ‘t’,

○ ‘w’ is the inertia weight,

○ $\mathbf{c_1}$ and $\mathbf{c_2}$ are acceleration coefficients (cognitive and social components),

○ $\mathbf{r_1}$ and $\mathbf{r_2}$ are random numbers between 0 and 1 (random exploration components).

After updating velocities, particles adjust their positions based on the new velocities using the Eq. (6):

$$\mathbf{X_i(t+1) = X_i(t) + v_i\,(t+1)} \quad \mathbf{(6)}$$

- Fitness Evaluation Phase

After updating positions, each particle's fitness is evaluated using a fitness function $\mathbf{f(X_i(t+1))}$ that measures a corresponding solution's quality or performance in search space. Each particle compares its current positioning with its personal best positioning (pbest) as shown in Eq. (7):

$$\mathbf{f\left(X_i(t+1)\right) < f(pbest_i)} \quad \mathbf{(7)}$$

If the current positioning yields better fitness value, the particle updating it being pbest by $\mathbf{X_i(t+1)}$. Similarly, the global best positioning (gbest) is updated based on the best fitness value among all particles as shown in Eq. (8):

$$\mathbf{f(pbest_i) < f(gbest)} \quad \mathbf{(8)}$$

- Termination Phase

PSO keeps iterating through the moving update and the fitness evaluation phases until the termination criteria are met. Common termination conditions include reaching a maximum number of iterations, achieving the desired fitness threshold, or observing minuscule improvement over successive iterations. PSO converges when the particles collectively gravitate toward promising regions of the search space, idealistic converging toward an optimal solution.

To maximize the accuracy of a classification model, let x be the vector of hyperparameters we were optimizing, and let accuracy(x) be the accuracy of the model with hyperparameters x. The fitness function f(x) can be defined as shown in Eq. (9):

$$\mathbf{f(x) = Accuracy(x)} \qquad (\mathbf{9})$$

3.5. Classification using CFNN

To classify speakers, the feature vector is provided to a CFNN model. The Convolutional Fuzzy Neural Network (CFNN) is designed to leverage better the strengths of Convolutional Neural Network (CNN) and fuzzy logic for speaker identification [25]. CFNN model typically consists of three main components: a convolution network for feature extractions, a fuzzy layer for handling uncertainties and ambiguity in data, and a fully connected (FC) layer for classifications. Fig. 3 illustrates the architecture of CFNN.

1. Network 1(Convolutional layer & Pooling layer): A convolutional network is in charge of extracting high-level features from inputted audial data. Audio signals are first converted into a spectrogram, which serves as inputs to the CNN. The convolutional networks consist of multiple layers, including convolution layers, activation functions, and pooling layers. The feature maps are defined as shown in Eq. (10) and the pooling operation is done using Eq. (11):

$$\mathbf{f = (W * X + b)} \qquad (\mathbf{10})$$

$$\mathbf{P_{i,j} = max\left(X_{i+m,j+n}\right)} \qquad (\mathbf{11})$$

In Eq. (10), W - denotes the Kernel/filter, X - denotes the input Spectrogram, b is the bias, * denotes the convolution operation and f is the activation function. In Eq. (11), the pooling operation takes the maximum value over a defined window size (m, n).

2. Network 2 (Fuzzy layer): The fuzzy layer processes the features extracted by a convolutional network. It does fuzzy clustering to handle ambiguity and uncertainty in data. Each neuron in the fuzzy layer represents a fuzzy membership function that on the degree indicates which feature vector belongs to a specific cluster. The fuzzy membership function is defined in Eq. (12):

$$\boldsymbol{\mu_l(x) = exp\left(-\frac{\| x - m_l \|^2}{2\sigma_l{}^2}\right)} \qquad (\mathbf{12})$$

In Eq. (12), $\boldsymbol{\mu_l(x)}$ is the membership value of input x to cluster l, $\mathbf{m_l}$ is the center of the l-th cluster, $\boldsymbol{\sigma_l}$ is the standard deviation to control the fuzziness. The normalization condition ensures that the membership values sum to 1 for all clusters as shown in Eq. (13):

$$\boldsymbol{\sum_{l=1}^{L} \mu_l(x) = 1} \qquad (\mathbf{13})$$

3. Network 3(Fully Connected Layer): The output of the fuzzy layer is fed into the fully connected (FC) layer, for the final classification. The FC layer assigns class labels to input based on the membership value as shown in Eq. (14):

$$\mathbf{z_j} = \sum\nolimits_i \mathbf{w_{ij}} \boldsymbol{\mu}_\mathbf{i}(\mathbf{x}) + \mathbf{b_j} \qquad (14)$$

$$\textbf{Activation function } (\mathbf{y_j}) = \frac{\mathbf{exp}\ (\mathbf{z_j})}{\sum_\mathbf{k} \mathbf{exp}\ (\mathbf{z_k})} \qquad (\mathbf{15})$$

In Eq. (14), $\mathbf{z_j}$ denotes the input to the activation function for class j, $\mathbf{w_{ij}}$ is the weight connecting fuzzy neuron i to output neuron j, and $\mathbf{b_j}$ is the bias for class j. In Eq. (15) $\mathbf{y_j}$ is the probability of the input belonging to class j.

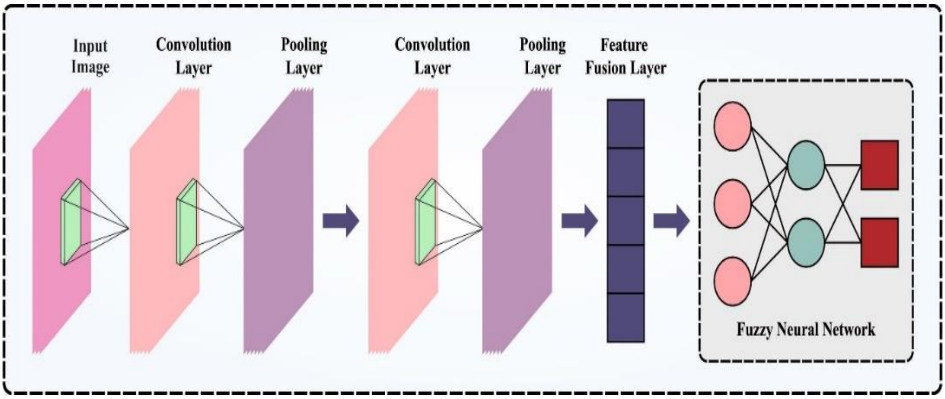

Fig. 3. Structure of CFNN

By combining these layers, the CFNN model effectively extracts, processes, and classifies audio features for robust speaker identification, handling both variabilities and uncertainty within input data.

4. Experimental Results

4.1 Implementation Setup

In this section, the PSO-CFNN model was subjected to various experimental validation to identify the speakers by analyzing an audio file, considering all kinds of aspects. Testing phases used Python 3.6.5 on a system equipped with an i5-8600K CPU & 250GB SSD, a GeForce 1050Ti 4GB graphics card, 16GB RAM, and a 1TB hard drive. For validations, a benchmark dataset from Kaggle containing audio files was utilized [26]. The total number of test samples is detailed in Table 1. The evaluation of the PSO-CFNN model's performance included key metrics such as accuracy, precision, recall, F-scores, & error rate as defined in Eqs. (13-16). A True Positive (TP) happens when the Model correctly identifies a positive category, True Negative (TN) occurs when the negative class is accurately identified by the model. False Positives (FP) are instances where the model incorrectly predicts the positive class, and False Negatives (FN) are cases where the model incorrectly predicts the negative class. The definition of these measures is as follows:

$$\mathbf{Accuracy} = \frac{\mathbf{TP + TN}}{\mathbf{TP + FP + FN + TN}} \times \mathbf{100} \qquad (\mathbf{13})$$

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP + FP}} \qquad (\mathbf{14})$$

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \qquad (\mathbf{15})$$

$$\mathbf{F - Score} = \frac{\mathbf{2 * Precision * Recall}}{\mathbf{Precision + Recall}} \qquad (\mathbf{16})$$

Error Rate: An error rate means the portion of wrong answers seen in a method, network, or evaluation. It's often shown as percentages or ratios that compare error counts with total observations or tries conducted.

Table 1. Dataset Details

Class	Number. of. Samples
Speaker 1	1500
Speaker 2	1500
Speaker 3	1500
Speaker 4	1500
Speaker 5	1500
Total	7500

Fig. 4 and Fig. 5 illustrate the performance results of PSO-CFNN strategy usage in a 70:30 train/test datasets split. Data clearly showed that the PSO-CFNN method achieves the highest accuracy for train and validation. Noticeably, test accuracy appears to surpass train accuracy. Also, Fig. 4 & Fig. 5 display the train and validation loss figures associated with the PSO-CFNN approach using the same 70:30 train/test split. These results show that the PSO-CFNN algorithm achieved the lowest score for both train loss and validation loss, with validation loss being lower than training loss.

Table 2. Average Outcomes of the PSO-CFNN Model with Various Metrics a 70:30 Split of the TR/TS Data

Class	Accuracy (%)	Precision (%)	Recall (%)	Error Rate (%)	F-Score (%)
Training Phase (70%)					
Speaker – 1	96.92	100.00	88.24	3.08	93.75
Speaker – 2	98.46	92.31	100.00	1.54	96.00
Speaker – 3	100.00	100.00	100.00	0	100.00
Speaker – 4	98.46	93.75	100.00	1.54	96.77
Speaker – 5	100.00	100.00	100.00	0	100.00
Average	98.77	97.21	97.65	1.23	97.30
Testing Phase (30%)					
Speaker – 1	98.00	100.00	98.24	2.00	96.75
Speaker – 2	98.46	95.31	100.00	1.54	97.00
Speaker – 3	100.00	100.00	100.00	0	100.00
Speaker – 4	100.00	96.75	100.00	0	97.77

Speaker – 5	100.00	100.00	97.15	0	100.00
Average	99.29	98.41	99.07	0.71	98.30

Fig. 6 and Fig. 7 present the confusion matrices, Precision-Recall, and ROC curves produced by the PSO-CFNN model for different sizes of training (TR) and test (TS) datasets. The results indicate that the PSO-CFNN model excels in speaker recognition. Illustrated in Fig. 8 are the aggregate outcomes for classifying speakers utilizing the PSO-CFNN methodology on 70% of training data. Such outcomes suggest that the technique achieves enhanced performance across all categories. Specifically, Within the Speaker 1 category, the PSO-CFNN strategy accomplishes an accuracy of 96.92%, precision of 100.00%, Recall of 88.24%, Error rate of 3.08%, and F_score of 93.75%. Regarding the Speaker 2 category, this approach secures an accuracy of 98.46%, precision of 92.31%, Recall of 100.00%, Error rate of 01.54%, and F_score of 96.00%. In the context of Speaker 3, the strategy achieves an accuracy of 100.00%, precision of 100.00%, Recall of 100.00%, Error rate of 0%, and F_score of 100.00%. For the Speaker 4 category, the model achieves an accuracy of 98.46%, precision of 93.75%, Recall of 100.00%, Error rate of 01.54%, and F_score of 96.77%. Regarding the Speaker 5 category, this approach secures an accuracy of 100.00%, precision of 100.00%, Recall of 100.00%, Error rate of 0%, and F_score of 100.00%. Table 2 depicts the overall performances of the PSO-CFNN model for speaker identification, using 70% of the data for training and 30% for testing. The result highlights the model's ability to identify each speaker category accurately.

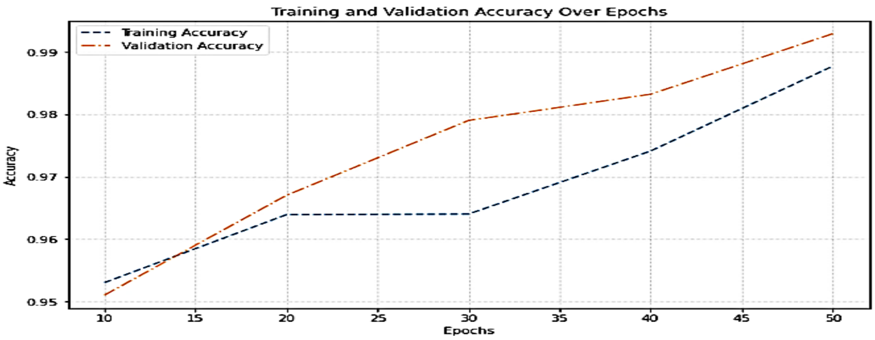

Fig. 4. Accuracy Graph Based on Training and Testing Set

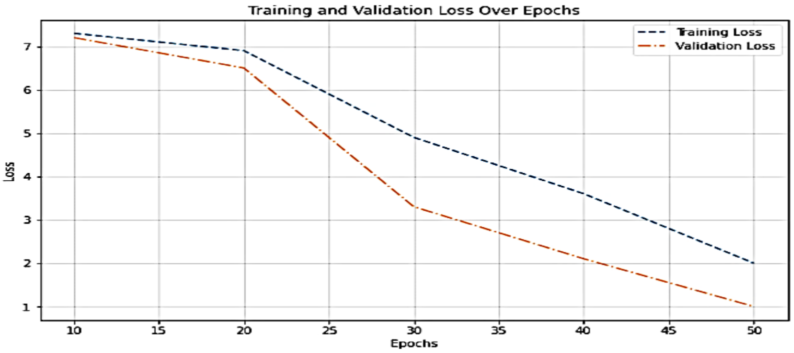

Fig. 5. Loss Graph Based on Training and Testing Set

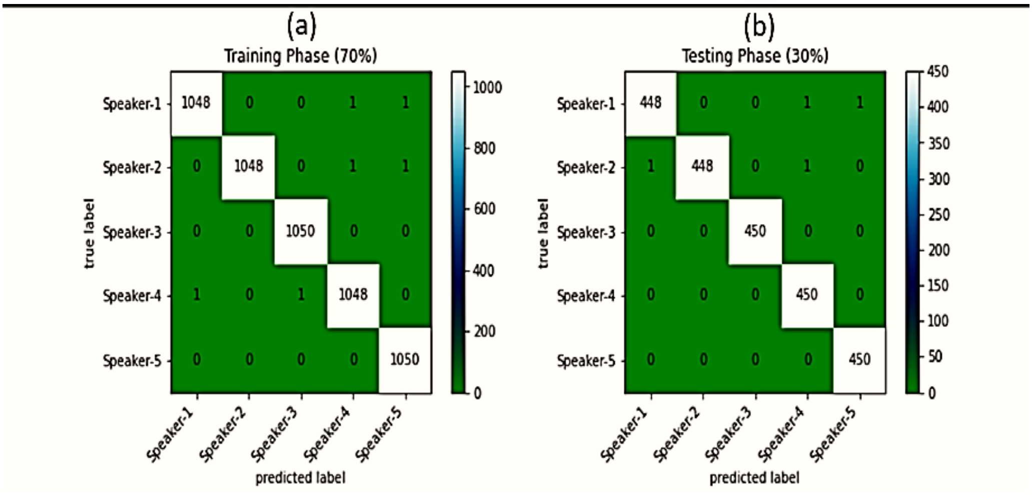

Fig. 6. (a) Confusion Matrix based on TR set (b) Confusion Matrix based on TS set

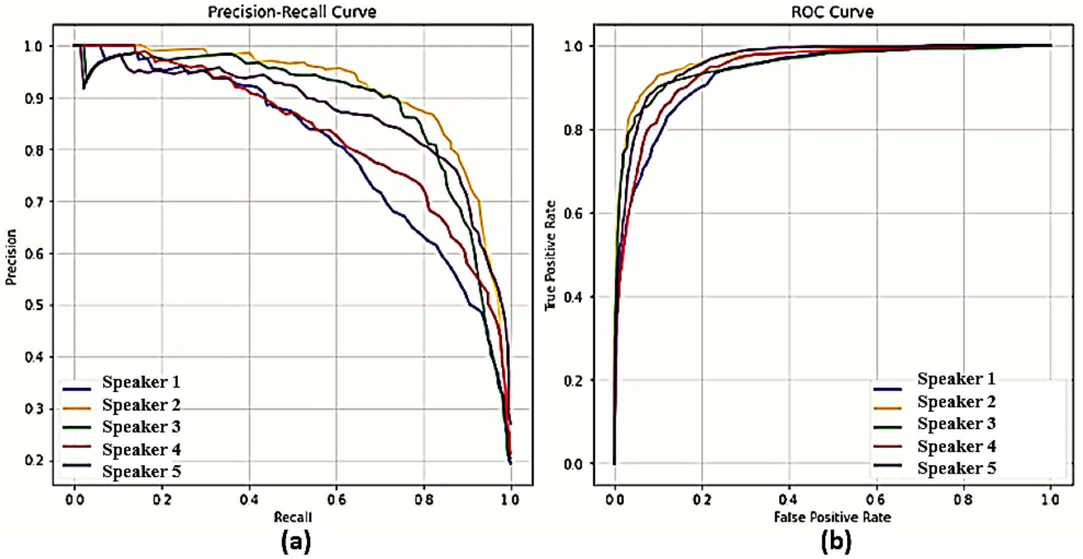

Fig. 7. (a) Precision-Recall Curve (b) ROC Curve

Illustrated in Fig. 9 are the aggregate outcomes for classifying speakers utilizing the PSO-CFNN methodology on 30% of testing data. Such outcomes suggest that the technique achieves enhanced performance across all categories. Specifically, Within the Speaker 1 category, the PSO-CFNN strategy accomplishes an accuracy of 98.00%, precision of 100.00%, Recall of 98.24%, Error rate of 2.00%, and F_score of 96.75%. Regarding the Speaker 2 category, this approach secures an accuracy of 98.46%, precision of 95.31%, Recall of 100.00%, Error rate of 01.54%, and F_score of 97.00%. In the context of Speaker 3, the strategy achieves an accuracy of 100.00%, precision of 100.00%, Recall of 100.00%, Error rate of 0%, and F_score of 100.00%. For the Speaker 4 category, the model achieves an accuracy of 98.46%, precision of 93.75%, Recall of 100.00%, Error rate of 01.54%, and

F_score of 96.77%. Regarding the Speaker 5 category, this approach secures an accuracy of 100.00%, precision of 100.00%, Recall of 97.15%, Error rate of 0%, and F_score of 100.00%.

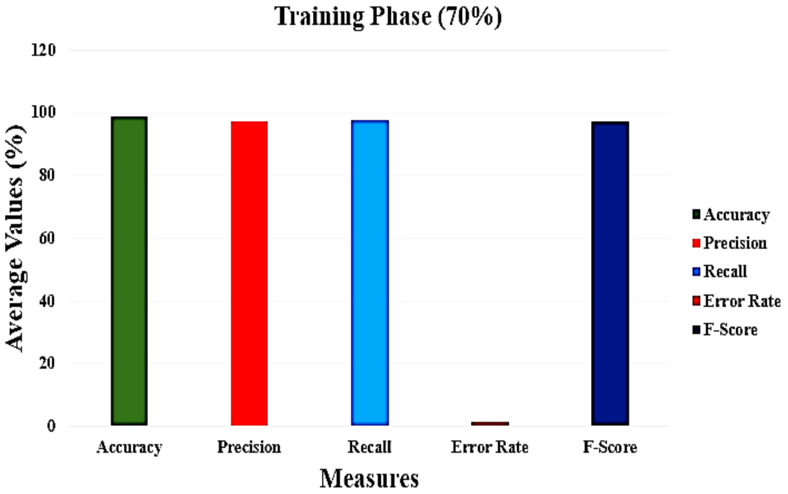

Fig. 8. Average outcome of PSO-CFNN approach on 70% training of data

The comparison of the experimental results for the feature parameters used in this work with other existing works [27] tested in this paper is shown in Table 3. It indicates that the proposed model outperforms other existing methods. Here, we tested MFCC, IMFCC, MFCC+IMFCC, & the feature parameters detailed in this paper.

Table 3. Evaluating the ERR of the Feature Extraction Ablation Experiments for PSO-CFNN Model versus Contemporary Techniques

Parameters	ERR (%)	t-DCF
MFCC	5.26	0.181
IMFCC	8.09	0.240
MFCC + IMFCC	3.56	0.153
PSO-CFNN	1.71	0.100

A comparative analysis was performed to prove the superior performance of the PSO-CFNN configurations presented in Table 4 [27]. Also, Fig. 10 gives accuracy comparisons among the PSO-CFNN approach and other modern methods. The analysis reveals that the MFCC-SOFM-MLP-GD framework records the lowest rates of success and accuracy, at 96.92%. Little better performance was noted in MFCC-SOFM-MLP-GDM, MFCC-SOFM-MLP-BR, MFCC-FW, and fusion various methods; it is achieving accuracies rates 97.05%, 97.62%, 97.32%, and 97.81%. However, the proposed method PSO-CFNN showed better results, reaching a high accuracy rate of 99.29%.

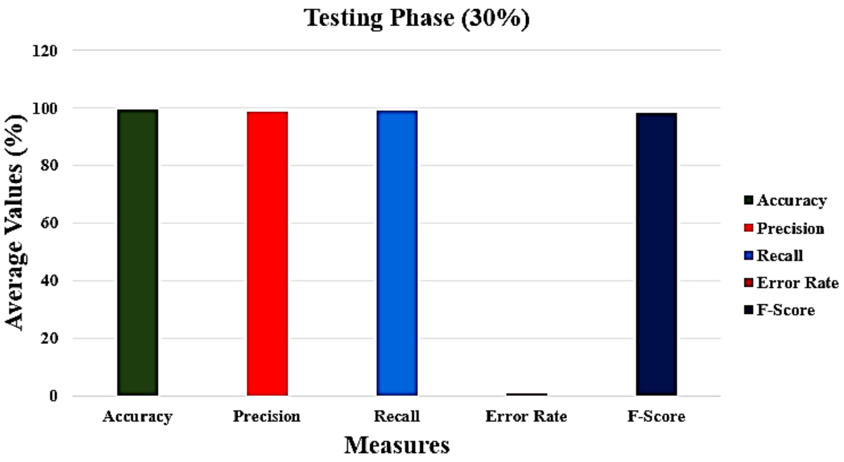

Fig. 9. Average outcome of PSO-CFNN approach on 30% of testing data

Table. 4. Evaluating the Accuracy of the PSO-CFNN Model Versus Contemporary Techniques

Models	Accuracy (%)	Error rate (%)
MFCC-SOFM-MLP-GD	96.92	03.08
MFCC-SOFM-MLP-GDM	97.05	02.95
MFCC-SOFM-MLP-BR	97.62	02.38
MFCC-FW	97.32	02.68
FUSION	97.81	02.19
Proposed PSO-CFNN	99.29	00.71

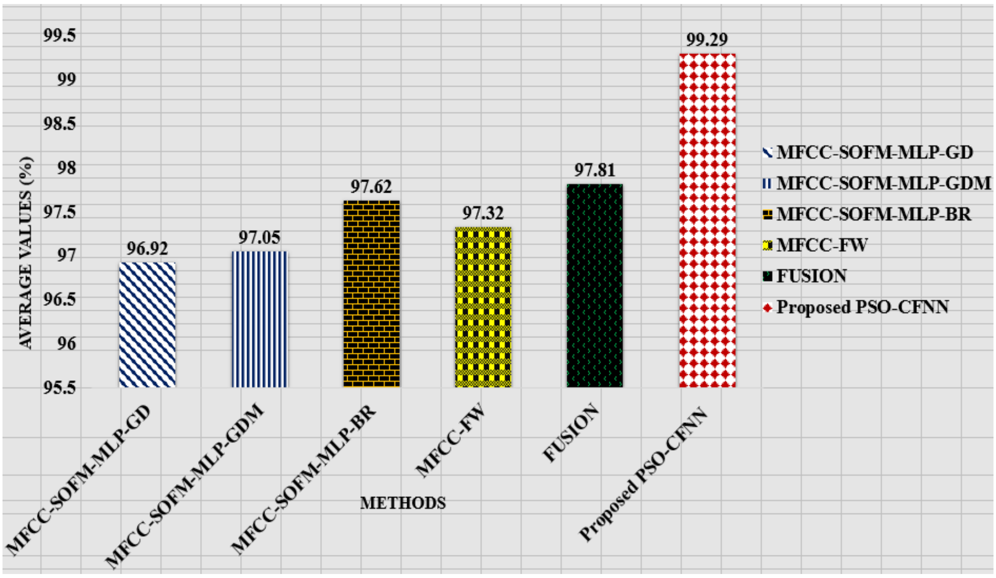

Fig. 10. Comparisons of the Accuracy of the PSO-CFNN Framework with existing methodologies

5. Conclusion

In the study, a novel PSO-CFNN model effectively authenticates speaker applications. This cutting-edge PSO-CFNN mechanism starts with the first use of Wavelet Denoising (WD) approaches to get rid of noise interference in audio signals. Then, a spectrogram is used as an input for the VGGVox architecture. Afterward, the Particle Swarm Optimization (PSO) Algorithm is applied to refine the hyperparameters linked to the CFNN design. Finally, the CFNN architecture functions like a classifier to aid automatic speech recognition (ASR). The efficacy of PSO-CFNN frameworks is tested through a detailed series of tests. A comparative analysis highlights the superior performance of the PSO-CFNN framework over other existing methods. This demonstrates the potential for strong ASR in real-time speaker identification scenarios. Looking ahead, a combined approach employing a fusion-based deep learning model might be explored to further enhance PSO-CFNN framework performance.

References

1. Machado, T.J.; Vieira Filho, J.; de Oliveira, M.A. Forensic speaker verification using ordinary least squares. Sensors 2019, 19, 4385.
2. Pawar, R. V., Kajave, P. P., & Mali, S. N. (2005, August). Speaker Identification using Neural Networks. In Iec (prague) (pp. 429-433).
3. Hossain, M. M., Ahmed, B., & Asrafi, M. (2007, December). A real time speaker identification using artificial neural network. In 2007 10th international conference on computer and information technology (pp. 1-5). IEEE.
4. Devi, K. J., & Thongam, K. (2023). Automatic speaker recognition from speech signal using bidirectional long-short-term memory recurrent neural network. Computational Intelligence, 39(2), 170-193.
5. Hamsa, S., Shahin, I., Iraqi, Y., Damiani, E., Nassif, A. B., & Werghi, N. (2023). Speaker identification from emotional and noisy speech using learned voice segregation and speech VGG. Expert Systems with Applications, 224, 119871.
6. Shah, S. H., Saeed, M. S., Nawaz, S., & Yousaf, M. H. (2023, February). Speaker recognition in realistic scenario using multimodal data. In 2023 3rd International Conference on Artificial Intelligence (ICAI) (pp. 209-213). IEEE.
7. Bharti, S., Kumari, A., Verma, D., & Kumar, A. (2024, January). Speaker Identification Using Various Machine Learning Algorithms. In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 714-719). IEEE.
8. Guo, X., Qin, X., Zhang, Q., Zhang, Y., Wang, P., & Fan, Z. (2023). Speaker recognition based on dung beetle optimized CNN. Applied Sciences, 13(17), 9787.
9. Nagarajan, D., Chourashia, K., & Udhayakumar, A. (2023, February). Neuro-Fuzzy Logic Application in Speech Recognition. In International Conference on Mathematical Modeling and Computational Science (pp. 1-9). Singapore: Springer Nature Singapore.
10. H. Ali, S. N. Tran, E. Benetos, and A. S. d'Avila Garcez, ''Speaker recognition with hybrid features from a deep belief network,'' Neural Comput. Appl., vol. 29, no. 6, pp. 13–19, Mar. 2018, doi:10.1007/s00521-016-2501-7.
11. S. Bunrit, T. Inkian, N. Kerdprasop, and K. Kerdprasop, ''Text-independent speaker identification using deep learning model of convolution neural network,'' Int. J. Mach. Learn. Comput., vol. 9, no. 2, pp. 143–148, Apr. 2019, doi: 10.18178/ijmlc.2019.9.2.778.
12. N. N. An, N. Q. Thanh, and Y. Liu, ''Deep CNNs with self-attention for speaker identification,'' IEEE Access, vol. 7, pp. 85327–85337, 2019, doi:

10.1109/ACCESS.2019.2917470.

13. S. A. El-moneim, M. A. Nassar, M. I. Dessouky, and N. A. Ismail, ‘‘Textindependent speaker recognition using LSTM-RNN and speech enhancement,’’ Multimed. Tools Appl., vol. 79, pp. 24013–24028, Sep. 2020.
14. T. Anvarjon and S. Kwon, ‘‘Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features,’’ Sensors, vol. 20, no. 18, p. 5212, Sep. 2020, doi: 10.3390/s20185212.
15. M. Turab, T. Kumar, M. Bendechache, and T. Saber, ‘‘Investigating multi-feature selection and ensembling for audio classification,’’ Int. J. Artif. Intell. Appl., vol. 13, no. 3, pp. 69–84, May 2022, doi:10.5121/ijaia.2022.13306.
16. N. Basu, A. S. Bali, P. Weber, C. Rosas-Aguilar, G. Edmond, K. A. Martire, and G. S. Morrison, ‘‘Speaker identification in courtroom contexts—Part I: Individual listeners compared to forensic voice comparison based on automatic-speaker-recognition technology,’’ Forensic Sci. Int., vol. 341, Dec. 2022, Art. no. 111499, doi: 10.1016/j.forsciint.2022.111499.
17. S. Farsiani, H. Izadkhah, and S. Lotfi, ‘‘An optimum end-to-end text independent speaker identification system using convolutional neural network,’’ Comput. Electr. Eng., vol. 100, May 2022, Art. no. 107882, doi: 10.1016/j.compeleceng.2022.107882.
18. D. Salvati, C. Drioli, and G. L. Foresti, ‘‘A late fusion deep neural network for robust speaker identification using raw waveforms and gamma tone cepstral coefficients,’’ Exp. Syst. Appl., vol. 222, Jul. 2023, Art. no. 119750, doi: 10.1016/j.eswa.2023.119750.
19. S. Hamsa, I. Shahin, Y. Iraqi, E. Damiani, A. B. Nassif, and N. Werghi, ‘‘Speaker identification from emotional and noisy speech using learned voice segregation and speech VGG,’’ Exp. Syst. Appl., vol. 224, Aug. 2023, Art. no. 119871, doi: 10.1016/j.eswa.2023.119871.
20. M. Rayhan Ahmed, S. Islam, A. K. M. Muzahidul Islam, and S. Shatabda, ‘‘An ensemble 1D-CNN-LSTM-GRU model with data augmentation for speech emotion recognition,’’ Exp. Syst. Appl., vol. 218, May 2023, Art. no. 119633, doi: 10.1016/j.eswa.2023.119633.
21. Elsaid, A., Khalaf, A. A., El-Rabaie, S., & El-Samie, A. (2024). An efficient Automatic Speaker Identification System based Pitch Frequency Estimation in Degraded Environmental Conditions. Journal of Advanced Engineering Trends, 43(1), 221-229.
22. Manfron, E., Teixeira, J. P., & Minetto, R. (2023, September). Deep Learning and Machine Learning Techniques Applied to Speaker Identification on Small Datasets. In International Conference on Optimization, Learning Algorithms and Applications (pp. 195-210). Cham: Springer Nature Switzerland.
23. Al-Dulaimi, H. W., Aldhahab, A., & Al Abboodi, H. M. (2023). Speaker Identification System Employing Multi-resolution Analysis in Conjunction with CNN. International Journal of Intelligent Engineering & Systems, 16(5).
24. Thukroo, I. A., Bashir, R., & Giri, J. K. (2023). Improved Support Vector-Recurrent Neural Network with Optimal Feature Selection-based Spoken Language Identification System. Indian Journal of Science and Technology, 16(10), 680-697.
25. Almoussawi, Z. A., Rasool, H. A., Taher, N. A., Al-Attabi, K., Khalid, R., & Abdulhussain, Z. N. (2023, July). Improved Arithmetic Optimization with Deep learning Driven Contactless Biometric Verification on Iris Images. In 2023 6th International Conference on Engineering Technology and its Applications (IICETA) (pp. 502-508). IEEE.
26. https://www.kaggle.com/code/auishikpyne/speaker-identification/input
27. Gaurav, Bhardwaj, S., & Agarwal, R. (2023). Two-Tier Feature Extraction with Metaheuristics-Based Automated Forensic Speaker Verification Model. Electronics, 12(10), 2342.