Computational intelligence in chromosomal primitive extraction and speaker recognition

Original Scientific Paper

Mohamed Hedi Rahmouni
University of Tunis El Manar
Faculty of Sciences of Tunis. Research Laboratory: LAPER.
Street Bechir Salem Belkhairya - Tunisia
medhedi.rahmouni@fst.utm.tn

Mohamed Salah Salhi
University of Tunis El Manar
National Engineering School of Tunis,
Research Laboratory of Signal Image and Information Technology LR-SITI
Street Bechir Salem Belkhairya -Tunisia
medsalah.salhi@enit.utm.tn

Mounir Bouzguenda
King Faisal University, Department of Electrical Engineering, College of Engineering,
Al Ahsa, 31982, Saudi Arabia
mbuzganda@kfu.edu.sa

*Corresponding author

Hatem Allagui
University of Tunis El Manar
Faculty of Sciences of Tunis. Research Laboratory: LAPER.
Street Bechir Salem Belkhairya -Tunisia
hatem.allagui@fst.utm.tn

Ezzeddine Touti*
Center for Scientific Research and Entrepreneurship,
Northern Border University,
Arar 73213, Saudi Arabia
esseddine.touti@nbu.edu.sa

***Abstract** – This research presents an innovative approach leveraging computational intelligence for chromosomal primitive extraction and speaker recognition. The study emphasizes real-time digital signal processing (DSP) embedded systems integrating chromosomal-inspired techniques to enhance auditory feature extraction and speaker identification accuracy. By applying Gamma chromosomal factors, Mel-Frequency Cepstral Coefficients (MFCC) are refined through convolution, emulating human cochlear functionality. This integration aligns well with the perceptual auditory mechanisms and computational intelligence paradigms. The proposed methodology incorporates feature extraction techniques like Linear Predictive Coding (LPC), Linear Predictive Cepstral Coefficients (LPCC), and MFCC, followed by robust classifiers such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Self-Organizing Maps (RSOM). Experimental results demonstrate superior performance of RSOM, achieving a recognition rate of up to 99.7% with Gamma-enhanced MFCCs, compared to 98.6% for SVM and 91% for SOM. The RSOM model effectively identifies speakers across diverse conditions, albeit with slightly increased response times due to its dynamic recurrence loop. This work addresses challenges like environmental noise and variability in speech styles by introducing the Gamma chromosomal factor, a logarithmic nonlinear enhancement model. The experimental setup, executed on DSP boards using Python, highlights the advantages of computationally intelligent systems in real-world applications such as biometric authentication and decision-making systems. These findings underscore the potential of chromosomal-inspired computational techniques to advance speaker recognition technology, offering high accuracy and reliability in adverse conditions. Future research will focus on optimizing architectural and software frameworks to improve response times and further integrate this approach into constrained real-time systems.*

***Keywords**: Computational intelligence; embedded systems; chromosomal primitive extraction; speaker recognition*

Received: January 17, 2025; Received in revised form: April 26, 2025; Accepted: April 28, 2025

1. INTRODUCTION

Speech is a natural and variable process that serves as the primary means of human communication, conveying information through acoustic signals. Speaker recognition has evolved from statistical models like HMM and GMM-UBM to deep learning approaches such as CNNs, RNNs, and wav2vec 2.0. Computational intelligence techniques, including Recurrent Self-Organizing Maps (RSOM) and Genetic Algorithms (GA), have shown promise in speech processing. However, their integration remains underexplored, particularly for embedded systems. This study considers the foundation of speaker identification, a subset of artificial intelligence (AI), that involves distinguishing individuals based on their speech [1]. It aims to enhance recognition accuracy and efficiency while benchmarking against existing methods. An evolutionary recurrent neural system processes these acoustic vectors through unsupervised learning, associating each vector with a speaker's identity stored in a database. During the control phase, it compares new inputs to its stored data and makes an identification decision [2]. This process is implemented in embedded systems, such as digital signal processors (DSPs), under real-time constraints. Speech and speaker recognition, alongside facial recognition, are critical fields in Industry 4.0, IoT, blockchain, and cloud computing. These technologies are vital for security and decision-making applications [3].

Approaches such as Cochlear coefficients and its derivatives are widely employed, with the selection guided by the specific demands and limitations of the system. This paper introduces, as second contribution, a novel approach to extracting speech signal features and examines the most accurate classification algorithms for speaker identification. The aim is to enhance robotic safety, voice control, and decision-making, targeting zero-error performance, even in challenging environments.

This work is structured as follows: Section 1 contextualizes speaker recognition, tracing key models to the adopted computational intelligence approach. Section 2 s speech feature extraction methods and classifiers, highlighting accuracy, decision speed, and comparative analysis. Finally, sections 3 and 4 focus, respectively, on experimental results and discussion.

2. APPLIED METHODS

2.1. MAIN TECHNIQUES FOR EXTRACTING PRIMITIVES

ASR transcribes speech into text, while speaker identification determines identity using vocal features. Despite differing goals, they share techniques and can be integrated for more robust, personalized systems. [4]. Fig. 1 illustrates the global architecture of the voice recognition system.

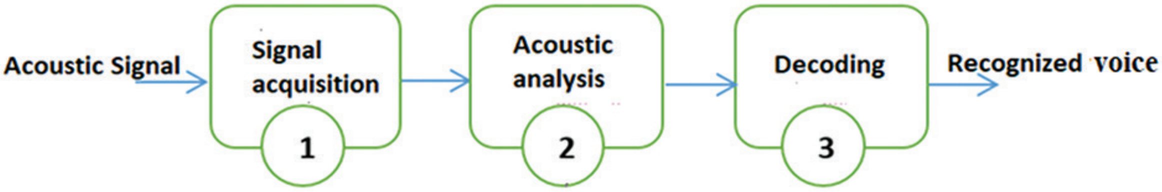


Fig. 1. Global architecture of a voice recognition system

Speaker recognition captures speech, digitizes it via DSP, extracts features (e.g., LPC, MFCC), and matches them to a database. It identifies speakers by balancing feature detail with reduced dimensionality. The primary techniques for extracting speech primitives are as follows:

a. Linear predictive coding (LPC)

Since the 1960s, LPC models speech as a filter with poles, representing the vocal tract, using filter coefficients to describe its transfer function [5].

Linear prediction (LP) is a key tool in speech analysis, modeling the signal $s(n)$ at time n based on p previous samples. The weighted sum of these samples produces a prediction error, $e(n)$, as shown in equation 1.

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \tag{1}$$

Linear Prediction (LPC) computes coefficients ak to minimize the error $e(n)$, commonly using autocorrelation or covariance, with autocorrelation preferred for its efficiency and stability. The LPC technique's block diagram is shown in Fig. 2.

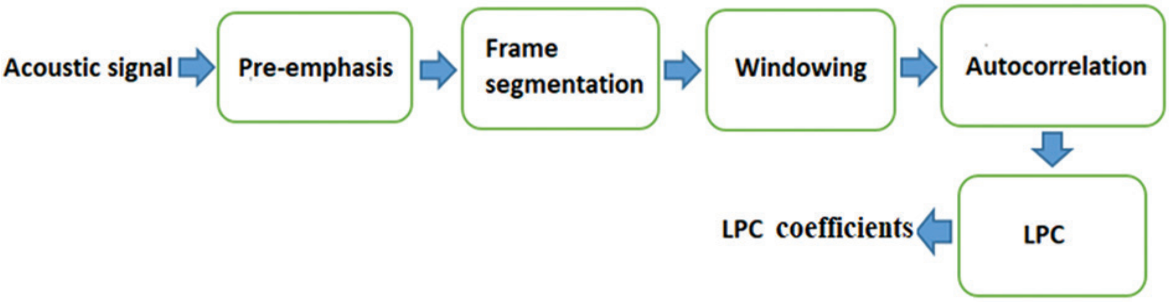


Fig. 2. Schematic representation of the LPC method

Consider $x(t)$ as the speech signal; the temporal autocorrelation function is expressed as:

$$c(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)x(t-\tau)dt \tag{2}$$

This represents the average over time of the signal multiplied by its own version shifted by a time delay τ. For a digital signal xk, sampled with a period Te, the discrete autocorrelation function is calculated using the equation:

$$C_n = \frac{1}{M}\sum_{k=i}^{i+M-1} x_k\, x_{k-n} \qquad (3)$$

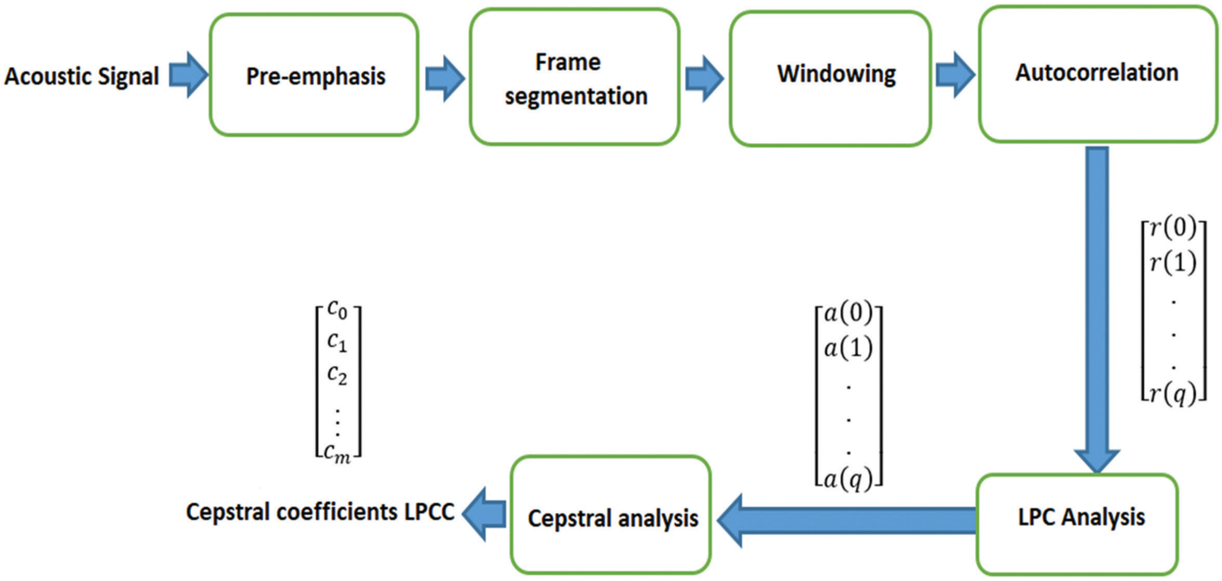
Here, M represents the number of points considered in computing the average, where the total duration is $T{=}M.Te$. The Levinson-Durbin algorithm [6] is used to determine signal coefficients by applying it to the filter signal for linear prediction. It calculates the linear prediction coefficients that minimize the mean squared error, as defined by:

$$E = \frac{1}{N}\sum_{n=0}^{N} e(n) \ where\ e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad (4)$$

The autocorrelation method computes LPC coefficients from windowed frames, precisely modeling the vocal tract's spectral envelope.

b. Linear Predictive Cepstral Coefficients (LPCC)

LPCC smooths the speech signal's spectral envelope while extracting speaker characteristics. Based on LPC analysis, it derives coefficients from the prediction process. Fig. 3 shows the LPCC extraction block diagram.



Acoustic Signal
Pre-emphasis
Frame segmentation
Windowing
Autocorrelation
$\begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(q) \end{bmatrix}$
LPC Analysis
$\begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(q) \end{bmatrix}$
Cepstral analysis
Cepstral coefficients LPCC
$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$

Fig. 3. Schematic illustrating the process of LPCC feature extraction.

The parameters are determined using the following equation:

$$\begin{gathered} c_n = \sum_{k=1}^{n-1}(1-\tfrac{k}{n})a_k c_{n-k} + a_n \\ c_1 = a_1 \quad 1 < n \le p \end{gathered} \qquad (5)$$

C_n : the n^{th} coefficient of cepstrum
A_n : the n^{th} linear predictor coefficient LPC

c. Mel Frequency Cepstral Coefficient (MFCC)

In 1980, Davis and Mermelstein introduced Mel-Frequency Cepstral Coefficients (MFCC) analysis [7], a robust parameter extraction method based on the Mel scale. It uses FFT and DCT to derive decorrelated coefficients that closely simulate human auditory perception. The Mel scale, reflecting the human ear's sensitivity, is linear at low frequencies and logarithmic at high frequencies, as defined by the following equation [8]:

$$Mel(f) = 2595 log_{10}\left(1+\frac{f}{700}\right) \qquad (6)$$

MFCC extraction involves pre-emphasis, segmentation with a Hamming window, FFT, and Mel-scaled filter banks. The first 12 coefficients from 20-30 ms overlapping windows are used for analysis.

$$s(n) = x(n) * w(n) \qquad 0 \le n \le N-1 \qquad (7)$$

The Fast Fourier Transform (FFT) is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) of a discrete signal $x(n)$.

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j2\pi nf} \qquad (8)$$

Proceed to the discretization of the frequency on N points among [-F_e/2 ,F_e/2] by putting:

$$f = \frac{k}{N} \ avec\ k = 0,1 \ldots\ldots N-1$$

We write in this case:

$$X\left(\frac{k}{N}\right) = X(k) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j2\pi\frac{k}{N}n} \qquad (9)$$

The discrete Fourier transform (DFT) for N frequency points of a discrete signal is expressed as:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi k\frac{n}{N}} \qquad (10)$$

Where, $X(k)$ is the DFT output.

N is the sample count per frame, enabling time-to-frequency conversion. The Mel scale (1937) models auditory spectra using triangular filters, crucial for cepstral coefficient calculation. [9 -10]. Fig. 4 shows the general shape of the Mel-scale filter bank.

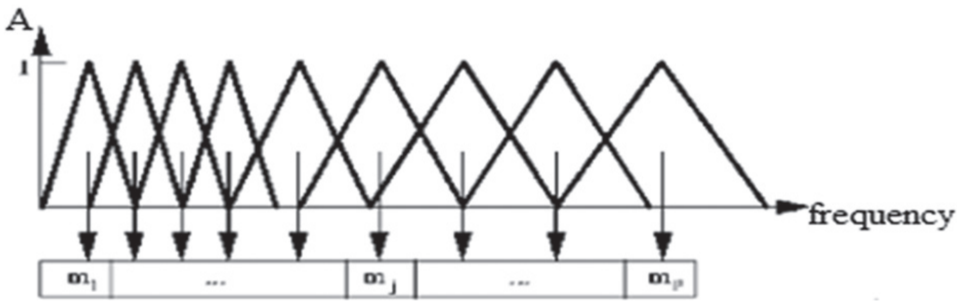


Fig. 4. Mel Scale Filter Bank

To ensure a smooth, stable spectrum, the energy logarithm (amplitude spectrum logarithm) is computed as follows:

$$s(m) = 20log_{10}\left(\sum_{k=0}^{N-1}|X(k)|\, H(k)\right) \tag{11}$$

Here, m represents the number of Mel scale filters, ranging from 20 to 40.

$X(k)$ denotes the FFT of the frame, while $H(k)$ refers to the transfer function of the Mel filter.

The Discrete Cosine Transform (DCT) is applied to filter coefficients, enhancing discriminative power and noise robustness for speech recognition. The coefficients $c(n)$ are calculated using the following equation [11]:

$$c(n) = \sum_{m=0}^{N-1} \mathrm{s(m)cos}\left[\frac{\mathrm{n\pi(m-0.5)}}{\mathrm{M}}\right] \quad 0 \le \mathrm{n} \le \mathrm{M} \tag{12}$$

In this context $c(n)$ represents the MFCC coefficients. s(m) denotes the logarithmic spectrum. N indicates the number of samples within each frame. M refers to the number of filter banks.

MFCC dynamic features are captured by delta and acceleration coefficients, reflecting temporal changes, with typical speech systems sampling at 16 kHz. and extracts these features [12].

$$x_{\boldsymbol{k}} = \begin{cases} c_{\boldsymbol{k}} \\ \Delta c_{\boldsymbol{k}} \\ \Delta\Delta c_{\boldsymbol{k}} \end{cases}$$

Where, c_k : is the MFCC vector of the kth frame

$\Delta ck = c_{k+2} - c_{k-2}$: first derivative of the MFCCs calculated from the MFCC vectors of the kth+2 frame and kth-2 frame;

$\Delta\Delta c_k = \Delta c_{k+1} - \Delta c_{k-1}$: second derivative of MFCC.

d. Comparative study between primitives' extraction techniques

A comparative study of MFCC, PCA, and ARMA for voice feature extraction assesses their effectiveness, efficiency, and suitability in speech processing. Additional methods like spectral subtraction, LPC, Wiener filtering, and independent component analysis ICA aid in noise separation. Key comparison factors include computational complexity, noise robustness, and speech quality [13].

Each method has its strengths depending on the application. For speech recognition, MFCC and LPCC are commonly used. PCA is mainly for dimensionality reduction. LPC and ARMA are more relevant for speech synthesis and modeling. [14 -15].

2.2. CLASSIFICATION MODELS

Classification identifies speakers by matching features with a database using classifiers like HMM, SVM, k-means, PCA, and ANN [16].

a. Artificial Neural Networks ANN

An artificial neural network (ANN), inspired by the human brain, processes and produces information. Multi-layer perceptron (MLP) networks have three layers: input, hidden (for non-linear processing), and output (for results). See Fig. 5 [17 - 18].

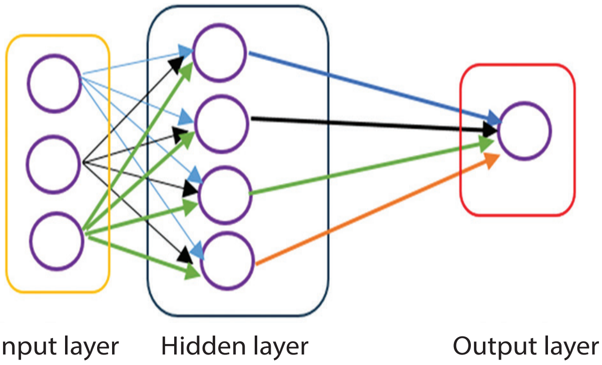


Fig. 5. Representative of ANN structure

ANN is widely used in speech and speaker recognition under artificial intelligence applications (Deep learning and Q-learning). The self-organizing map (SOM), as a static tool, achieves 85-90% speaker recognition, while the recurrent dynamic neural map (RSOM) improves this to 97-100% under optimal conditions. See Fig. 6.

The layer consists of neurons functioning as interconnected, fundamental processing units, operating through the following sequence of steps:

Unsupervised Learning: The neurons are trained by processing MFCC vectors that represent the speech signals of known individuals.

Neuron Count Estimation: The total number of neurons, N_n, in the RSOM map is calculated using the formula $N_n = 2.5 \times C$, where C is the number of individuals (or vectors) involved in training. For example, recognizing 40 speakers typically requires around 100 neurons.

Neuron Specialization: After multiple training iterations, each neuron becomes fine-tuned to a specific input vector. In our case, the stop criterion is set at 100 iterations.

Testing and Identification: Once trained, the RSOM map can process any speech samples, analyze them, and visualize potential identification outcomes.

Weight Vector Representation: The weight vector associated with a specific neuron, indexed as iii, is described using the expression provided below [19]:

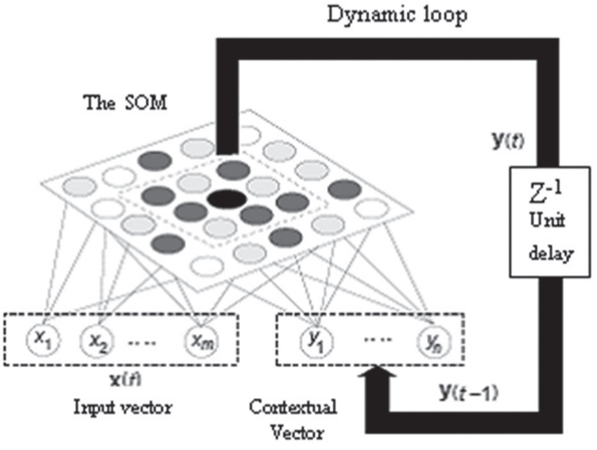


Fig. 6. Representation of Recurrent Self-Organizing Map RSOM

$$v_{pij} = \{w_{i1}; w_{i2}; w_{i3}; \ldots; w_{ij}\} \tag{13}$$

Each signal vector is sent to all neurons of the RSOM map. The neuron whose weight vector has the smallest Euclidean distance to the input vector is activated, determining whether the input corresponds to a known or unknown individual. The Euclidean distance between the input $x(t)$ and the weight w_i is calculated as follows:

$$\begin{gathered} E_i = \|x(t) - w_i\| \\ E_v = \min E_i \quad ; i \in N \end{gathered} \tag{14}$$

The winning neuron is the one that minimizes this Euclidean distance.

b. Support Vector Machine (SVM)

In the early 1990s, Vladimir Vapnik introduced the support vector machine (SVM), which projects data into a higher-dimensional space to find the best hyperplane for classification or regression. SVM solves the discrimination problem by constructing a function f that maps an input vector x to an output vector y [20].

$$y = f(x) = wx + b \tag{15}$$

The linear discriminant function is derived as a linear combination of the input vector $x=(x_1, x_2., x_N)$ and the weight vector $w : f(x)=wx+b$, $b \in R$ a scalar referred to as the bias.

If $f(x)>0$, it is decided that x is of class 1, otherwise, if $f(x)<0$, we decide x of class -1.

For classifying speech primitives using SVM, the following criteria are taken into account:

$$\begin{gathered} \text{class of (x)} = \text{sign f(x)} \\ = \text{sign (wx + b)=} \begin{cases} -1, \text{if f(x)} < 0 \\ 0, \text{if f(x)} = 0 \\ 1, \text{if f(x)} > 0 \end{cases} \end{gathered} \tag{16}$$

The margin of a hyperplane is defined as the shortest distance between the hyperplane and the closest data points. Let dis (x, w, b) denote the distance between a point x located on the plane H1 and the hyperplane defined by $f(x) = 0$. The margin M can be expressed as:

$$M = min \{dis \,(w \cdot x + b\,)\} \tag{17}$$

This distance is calculated as: $(f(x))/\|w\| = 1/\|w\|$, resulting in the distance between the two planes H1 and H2 being $2/\|w\|$.

The vectors w and b define the separating hyperplane, also known as the optimal hyperplane. Optimizing this hyperplane involves minimizing the squared norm $\|w\|^2$, leading to the objective: min(1/2 $\|w\|^2$).

This problem is typically solved using the Lagrange multipliers method. The classification function is represented as: $class(x) = sign(w \cdot x + b)$. The indicator function can also be reformulated based on the following expression [21].

$$\begin{gathered} w = \textstyle\sum_{i=1}^{l} \alpha_i y_i x_i \\ \text{So, } class\text{ (x)= } sign \left(\textstyle\sum_{i=1}^{l} (\alpha_i y_i x_i . x) + b\right) \end{gathered} \tag{18}$$

In practical classification scenarios, data frequently necessitates separation via a nonlinear decision boundary. This is accomplished by applying a kernel-based transformation $K(x, y)$, which optimizes the input data and is represented in the following form:

$$f(x) = \left(\textstyle\sum_{i=1}^{l} \alpha_i y_i K(x_i . x) + b\right) \tag{19}$$

Among the kernels used are:

$$\begin{cases} \text{linear:} \quad K(x,y) = x.y \\ \text{polinomial:} \quad K(x,y) = [(x,y) + 1]^d \\ \text{radial basis function RBF:} \quad K(x,y) = exp\{-\Psi(|x.y|^2\} \end{cases}$$

c. Comparative study of main classifier models

A comparative study of classifiers like HMM, RSOM, CSOM, SVM, and DNN assesses their effectiveness in speech recognition. HMMs achieve around 90% accuracy, while RSOMs capture temporal speech dynamics and CSOMs handle classification. CNNs excel in visual data analysis, and DNNs surpass 95% recognition rates.

SVMs, effective with non-linear boundaries, achieve 85-95% accuracy. X-vector and i-vector methods, combined with DNNs, exceed 98%, while Deep Speaker Embeddings (DES) can achieve over 99%, though environmental conditions can reduce performance to 92%. Each method's choice depends on application requirements and data quality.

Various performance parameters are used to evaluate the suggested model. The RSOM in its evolutionary form (hybridized with GA) demonstrates a strong balance between recognition accuracy and computational efficiency in speaker recognition tasks. It achieves high precision (93-99%), recall (89-97%), and F1-score (92-96%), making it competitive with deep learning models while maintaining a lower computational cost. Compared to traditional models like HMM (Precision: 75-82%, Recall: 72-80%, F1-score: 73-81%) and SVM (Precision: 78-85%, Recall: 75-83%, F1-score: 76-84%), RSOM outperforms in handling dynamic speech variations. However, modern deep learning approaches such as CNN (Precision: 88-94%, Recall: 86-93%, F1-score: 87-93%), LSTM (Precision: 90-96%, Recall: 89-95%, F1-score: 89-95%), and wav2vec 2.0 (Precision: 93-97%, Recall: 92-96%, F1-score: 92-96%) achieve higher recognition accuracy but at the expense of increased computational complexity. In terms of time efficiency, RSOM outperforms deep learning models, with processing times comparable to HMM and SVM, making it a viable choice for embedded systems and real-time speaker recognition applications.

The Table 1 below highlights some parameter scores supported by TIMIT database.

Table 1. Compared performances over existing models

Model	Recognition Accuracy (%)	Computational Cost (ms)	Dataset Used
HMM	81	100	TIMIT
SVM	87	200	TIMIT
CNN	92	350	TIMIT
DNN	92.5	500	TIMIT
i-vector	89	275	TIMIT
Deep RSOM Embeddings	96	850	TIMIT

2.3. ADOPTED METHOD

Speaker recognition on embedded systems uses machine learning tailored to resources, speed, and accuracy. Lightweight models like SVM suit low-resource systems, while complex algorithms work on high-resource systems. This work employs Python-programmed DSP cards for a comparative study of SVM and RSOM, with Fig. 7 illustrating the SVM algorithm.

This approach introduces Gamma Chromosomal Factors, a novel technique convolved with MFCC primitives to enhance speech feature extraction, especially in adverse environments. The resulting convolutional output is then fed into an evolutionary RSOM model embedded on a DSP. A comparative analysis with an embedded SVM model, known for its lightweight nature, underscores the advantages of our approach.

The experimental setup includes:

- a PC running the Code Composer Studio CCS software environment for programming a DSP board.
- a Texas Instruments TMS 320 DSP.
- a USB cable for downloading the program describing the model to be implemented to the DSP.
- a screen interface for viewing results and curves

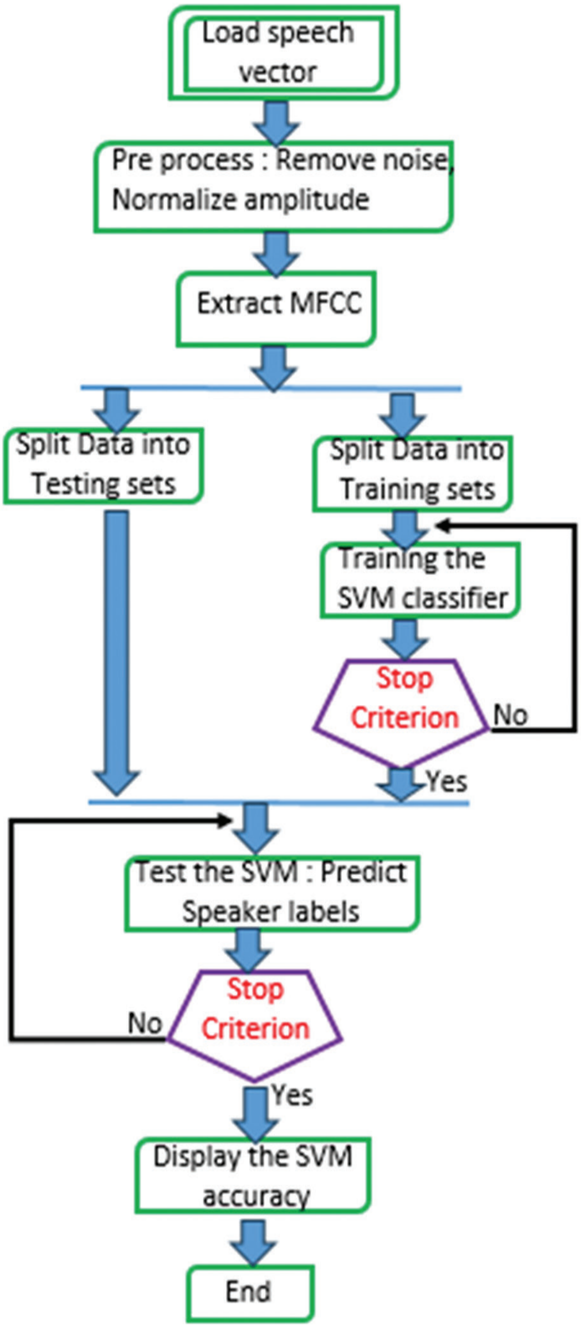

Load speech vector
Pre process : Remove noise, Normalize amplitude
Extract MFCC
Split Data into Testing sets
Split Data into Training sets
Training the SVM classifier
Stop Criterion
No
Yes
Test the SVM : Predict Speaker labels
Stop Criterion
No
Yes
Display the SVM accuracy
End

Fig. 7. Algorithm of experimented SVM Function

The algorithm assumes pre-processed, labeled speech signals split into training and testing sets. MFCC extraction functions are pre-implemented.

Compared to RSOM, it offers limited execution time and accuracy. Fig. 8 illustrates the optimized RSOM, where the BMU (Best Matching Unit) acts as a small intelligent processor, identifying the speaker.

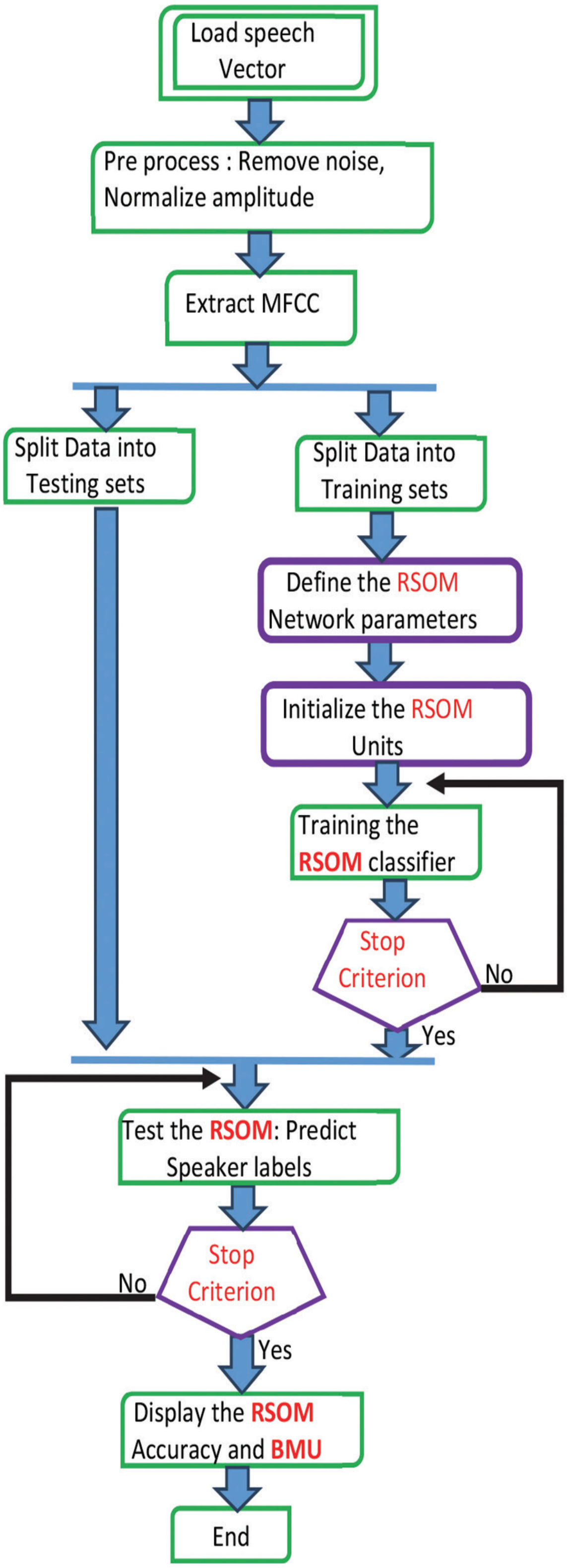

Load speech Vector
Pre process : Remove noise, Normalize amplitude
Extract MFCC
Split Data into Testing sets
Split Data into Training sets
Define the RSOM Network parameters
Initialize the RSOM Units
Training the RSOM classifier
Stop Criterion
No
Yes
Test the RSOM: Predict Speaker labels
Stop Criterion
No
Yes
Display the RSOM Accuracy and BMU
End

Fig. 8. Algorithm of adopted RSOM Function

3. RESULTS

MFCC coefficients in speaker recognition can be affected by environmental conditions, causing errors. Our approach uses a chromosomal factor, Gamma, to refine and enhance these coefficients. Tests on diverse speakers show Gamma ranges from 0.1 to 1.0. The results of this study are mentioned in Table 2 below.

Table 2. Chromosomal Gamma scores over conditions

Gamma	Day	Night
Men	0.9	1.0
Women	0.6	0.78
Children	0.4	0.55

Our contributed chromosomal factor Gamma is calculated using a logarithmic, non-linear model developed through experiments and validations.

$$Gamma\ (\gamma)= \alpha * log(\beta+\lambda) \qquad (19)$$

Alpha (α) represents a membership factor that characterizes the state of an individual speaker. Its value ranges between 0 and 1.

Beta (β) serves as an indicator of geographic conditions and atmospheric pressure, varying within the interval [0, 10].

Lambda (λ) denotes a coefficient associated with neighborhood noise. This coefficient is typically negligible, ensuring that ($\beta+\lambda$) ≤ 10.

Gamma chromosome, derived from deoxyribonucleic acid (DNA) composition, influences human biological traits. Its variation with day-night cycles can impact pronunciation.

Speaker recognition results on a DSP using Python depend on speech quality, feature extraction, classification algorithms, and system parameters. Accuracy ranges from 70% to 99%, and performance is assessed through metrics like precision, recall, and F1-score. Experimentation results for the sentence "I am Happy" spoken by five public people, using SOM, SVM, and RSOM without MFCC filtering, are presented in Table 3. Results may slightly vary with databases like TIMIT or VoxCeleb, but the performance gap between models remains consistent.

Table 3. Comparison of recognition rates across models using MFCC, excluding chromosomal Gamma

Models/ speakers	SOM rates in %	SVM rates in %	RSOM rates in %
Person 1	81.5	86.9	92
Person 2	83.2	88	94.5
Person 3	90.1	89.9	97.4
Person 4	87.6	95	99.2
Person 5	89	96.8	98

These results are illustreted in Fig. 9 below.

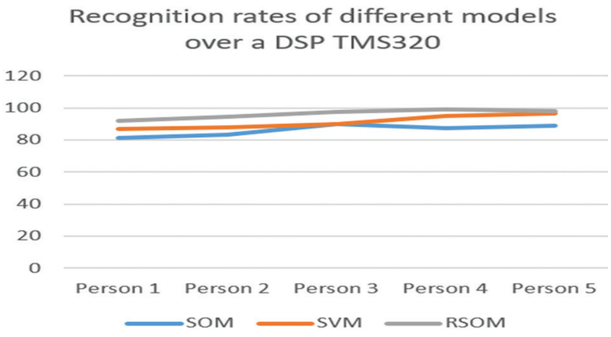


Fig. 9. Representation of recognition rates for different models over DSP

Nevertheless, RSOM exhibits a slightly longer response time of 30 ms, attributed to its dynamic recurrence loop, in contrast to 22 ms for SOM and 17 ms for SVM. The results highlight a trade-off: RSOM offers higher precision, while SVM is faster. RSOM remains the preferred choice, as real-time efficiency can be optimized through DSP hardware enhancements and software acceleration. RSOM's diverse neuron weights enhance adaptability to Deep Learning and Q-learning, while its recurrence loop adds dynamism, further improving results. When these models are tested using chromosomal Gamma MFCC primitives, an improvement in the results is observed, as shown in Table 4 below:

Table 4. Comparison of recognition rates across models using chromosomal Gamma MFCC primitives

Models/ speakers	SOM rates in %	SVM rates in %	RSOM rates in %
Person 1	82.1	87.4	92.7
Person 2	83.6	90	95
Person 3	91	91.5	98
Person 4	88.3	97,2	99.7
Person 5	89.5	98.6	98.5

Fig. 10 below illustrates the response of each model on their respective embedded systems, showing recognition rates when chromosomal Gamma is applied to MFCC primitives.

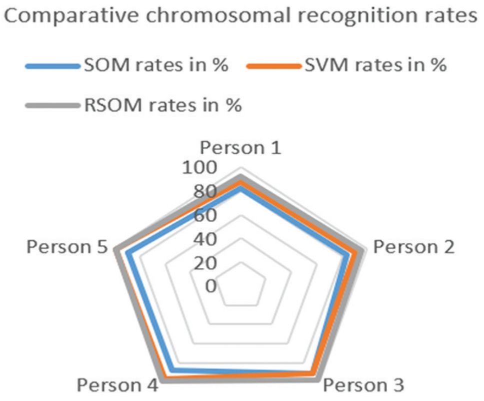


Fig.10. Visualization of recognition rates with chromosomal Gamma across various models on DSP

4. DISCUSSION

Speaker recognition is challenging due to factors like speech style, background noise, and microphone variations. Therefore, thorough evaluation under diverse conditions is crucial for reliability. As shown in Fig. 9, the RSOM model outperforms the SVM and SOM with a maximum recognition rate of 99.2%, compared to 96.8% for SVM and 90.1% for SOM, all without the chromosomal factor Gamma applied to the MFCC primitives.

By applying the convolutional method of MFCC primitives combined with the chromosomal factor Gamma to the spoken sentence during the testing phase of the various models, the following outcomes were observed:

- RSOM achieved the highest speaker recognition rate of approximately 99.7% in 34 ms.
- SVM reached a peak recognition rate of around 98.6% in 20 ms.
- SOM attained a maximum recognition rate of about 91% in 25 ms.

The embedded DSP model, utilizing computational sensors with the RSOM classifier, demonstrates superior speaker recognition performance compared to other models. However, it requires more processing time to generate its response. Consequently, architectural and software optimizations are suggested to enhance its suitability for a real-time, constrained system.

5. CONCLUSION

This research demonstrates the significant advancements made in the field of speaker recognition by leveraging computational intelligence and chromosomal-inspired techniques. The study developed an embedded real-time system using a combination of Mel-Frequency Cepstral Coefficients (MFCC) and Gamma chromosomal factors for feature extraction, along with advanced classifiers such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Self-Organizing Maps (RSOM). The results underscore the potential of these methods to significantly enhance recognition accuracy, even in challenging environmental conditions. Speaker recognition is inherently complex due to various challenges, including variability in speech styles, environmental noise, and device inconsistencies. The introduction of the Gamma chromosomal factor addresses these issues by providing a non-linear enhancement to MFCC, inspired by biological auditory processes. This factor adapts to variations in environmental conditions and speaker characteristics, enabling robust feature extraction and improving recognition rates. Experimental results demonstrate the superiority of RSOM in achieving a maximum recognition rate of 99.7% with Gamma-enhanced MFCCs, compared to 98.6% for SVM and 91% for SOM. However, RSOM's slightly increased response time due to its dynamic recurrence loop highlights a trade-off between accuracy and computational efficiency. The study also emphasizes the versatility and adaptability of computational intelligence techniques. The integration of MFCC primitives with the Gamma factor not only improves recognition performance but also aligns with human auditory perception, bridging the gap between biological inspiration and technological application. The real-time implementation on DSP boards demonstrates the feasibility of deploying these advanced techniques in embedded systems, making them suitable for various practical applications, including security, robotics, and voice-controlled systems.

Data availability statements

The data used to support the findings of this research are available from the corresponding author upon request.

Declaration of interest

The authors confirm that there are no conflicts of interest associated with this Paper.

Acknowledgements

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this research work through project number "NBU-CRP-2025-2448".

6. REFERENCES:

[1] T. Voegtlin, "Recursive Principal Components Analysis", Inria Campus Scientifique, Nancy, France, 2002.

[2] Q.-B. Hong, C.-H. Wu, H.-M. Wang, "Speaker-Specific Articulatory Feature Extraction Based on Knowledge Distillation for Speaker Recognition", Journal APSIPA Transactions on Signal and Information Processing, Vol. 12, No. 2, 2023.

[3] Z. Wang et al. "A hybrid model of sentimental entity recognition on mobile social media", EURASIP Journal on Wireless Communications and Networking, 2016, p. 253.

[4] M. S. Salhi, El M. Barhoumi, Z. Lachiri, "Effectiveness of RSOM Neural Model in Detecting Industrial Anomalies", Diagnostyka, Vol. 23, No. 1, 2023.

[5] M. S. Salhi, N. Khalfaoui, H. Amiri, "Evolutionary Strategy of Chromosomal RSOM Model on Chip for Phonemes Recognition", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.

[6] Z. Chen, P. Li, R. Xiao, T. Li, W. Wang, "A Multiscale Feature Extraction Method for Text-independent Speaker Recognition", Journal of Electronics & Information Technology, Vol. 43, No. 11, 2021.

[7] H. Liang, X. Sun, Y. Sun, Y. Gao, "Text feature extraction based on deep learning: a review", EURASIP Journal on Wireless Communications and Networking, 2017, p. 211.

[8] C. Hema, F. P. G. Marquez, "Emotional speech Recognition using CNN and Deep learning techniques", Applied Acoustics, Vol. 211, 2023, p. 109492.

[9] V. S. Dharun M. E, "Intelligent system speech recognition", Manonmaniam Sundaranar University, Tamil Nadu, India, 2012, PhD thesis.

[10] B. Medhi, P. H. Talukdar, "Different acoustic feature parameters ZCR, STE, LPC and MFCC analysis of Assamese vowel phonemes", Proceedings of the ICFM International Conference on Frontiers in Mathematics, Assam, India, 26-28 March 2015.

[11] S. Chen, Z. Luo, H. Gan, G. Mesnil, X. He, L. Deng, Y. Bengio, "An entropy fusion method for feature extraction of EEG", Neural Computing and Applications, Vol. 29, 2018, pp. 857-863.

[12] K. Bharti, P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering", Expert Systems with Applications, Vol. 42, No. 6, 2015, pp. 3105-3114.

[13] Y. Shen, X. He, J. Gao, "Learning semantic representations using convolutional neural networks for web search", WWW '14 Companion: Proceedings of the 23rd International Conference on World Wide Web, Seul, Korea, 7-11 April 2014, pp. 373-374.

[14] A. Severyn, A. Moschitti, "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks", Proceedings of the 38th International ACM SIGIR conference on research and development in Information Retrieval, Santiago, Chile, 9-13 August 2015, pp. 373-382.

[15] H. A. Elharati, M. Alshaari, V. Z. Këpuska, "Arabic Speech Recognition System Based on MFCC and HMMs", Journal of Computer and Communications, Vol. 8 No. 3, 2020.

[16] B. M. Tarabya, As. Khateb, S. Andria, "Processing Printed Words in Literary Arabic and Spoken Arabic: An fNIRS Study", Open Journal of Modern Linguistics, Vol. 11 No. 3, 2021.

[17] P. J. Worth, "Word Embeddings and Semantic Spaces in Natural Language Processing", International Journal of Intelligence Science, Vol. 13 No. 1, 2023.

[18] M. Koizumi, M. Maeda, Y. Saito, M. Kojima, "Correlations between Syntactic Development and Verbal Memory in the Spoken Language of Children with Autism Spectrum Disorders and Down Syndrome: Comparison with Typically Developing Children", Psychology, Vol. 11 No. 8, 2020.

[19] M. S. Salhi, S. Kashoob, Z. Lachiri, "Progress in Smart Industrial Control based on Deep SCADA Applied to Renewable Energy System", Turkish Online Journal of Qualitative Inquiry, Vol. 12, No. 8, 2021, pp. 871-882.

[20] F. A. Zadeh, A. R. Salehi, A. H. Mohammed, "An Analysis of New Feature Extraction Methods Based on Machine Learning Methods for Classification Radiological Images", Computational Intelligence and Neuroscience, 2022.

[21] Z. Yang, S. Serikawa, "Optimizing Speech Emotion Recognition with Hilbert Curve and convolutional neural network", Cognitive Robotics, Vol. 4, 2024, pp. 30-41.