Blind Source Separation of Audio Signals Using Independent Component Analysis

Mihai BARBU and Lucian ANTON

***Abstract*—Blind source separation represents a method of extracting individual signals from a complex mixture of signals without prior information about the original sources or their mixing pattern. Independent Component Analysis provides an effective framework that can be used to achieve blind separation of audio signals, based on the assumption that the signals from the sources are statistically independent and their mixing is linear. The concept of non-Gaussianity is used to achieve source separation through different methods of measuring it, such as kurtosis or negentropy. This work investigates the application of the FastICA algorithm to audio signal separation, presenting its mathematical foundations, implementation, and optimization strategies. The proposed approach is implemented and evaluated in MATLAB and App Designer, with experimental validation on mixtures of multiple audio signals. Results demonstrate the algorithm's ability to recover independent sources and highlight differences between nonlinear functions used for maximizing non-Gaussianity. The study emphasizes the potential of ICA- based BSS for civilian and military applications, and outlines future directions toward real-time processing and deployment in complex environments.**

***Index Terms*—Blind Source Separation, Independent Component Analysis, FastICA, Audio Signal Processing, Non-Gaussianity, Source Localization.**

I. INTRODUCTION

Advancements in technology have significantly influenced daily life, particularly in audio signal processing. Blind Source Separation (BSS), based on Independent Component Analysis (ICA), enables the extraction of individual signals from mixed audio sources without prior knowledge of the sources or the mixing process. This capability has broad applications in both civilian and military domains [1].

In civilian contexts, BSS can enhance communication device quality by reducing background noise, assist in medical signal analysis such as brain activity or intelligent hearing aids, and facilitate music production through the separation of instruments and vocals.

In military applications, BSS improves operational effectiveness by isolating relevant audio signals in noisy environments, enabling real-time voice recognition, source localization, and extraction of actionable intelligence. Strategically placed microphones can capture and separate signals to support decision-making, security monitoring, and mission coordination.

M. BARBU is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: mihai.barbu@mta.ro).
L. ANTON is with the Military Technical Academy "Ferdinand I", Bucharest, Romania (e-mail: lucian.anton@mta.ro).

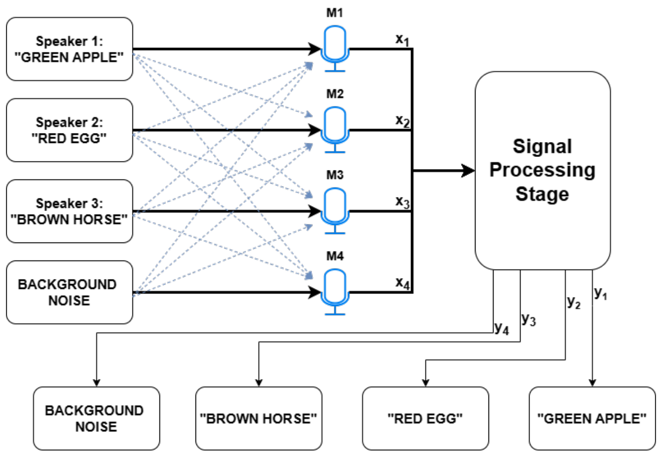

Figure 1. Example of real life Cocktail Party situation

An example that illustrates the goal of this application would be the Cocktail Party concept [1] as shown in Fig. 1, which represents a classic scenario in audio signal processing, where multiple people speak simultaneously in a noisy environment, and the objective is to separate each individual voice from the mixed signals captured by microphones.

This work is driven by the practical application of BSS using ICA for audio signal separation and the creation of tools with wide-ranging, high-impact potential. It underscores the growing significance of blind audio source separation in today's technology and data-intensive environments.

II. RELATED WORK

Over the past three decades, Independent Component Analysis (ICA) has emerged as a probabilistic method aimed at identifying statistically independent components from signal mixtures by exploiting non-Gaussianity. ICA theory was formalized in the 1990s and widely summarized in scientific conferences during the early 2000s [2].

In practical scenarios, measured signals often represent mixtures of underlying sources. For example, EEG or ECG recordings capture weighted sums of neural or physiological activities rather than isolated signals [2]. ICA aims to recover the original latent signals from observed mixtures.

Additionally, W. Du et al. analyzed three ICA algorithms, Infomax, Entropy Bound Minimization (EBM), and Full Blind Source Separation (FBSS), highlighting the importance of incorporating correlation information in functional MRI (fMRI) data analysis for identifying default mode networks (DMN) [3]. Recent studies have extended ICA to address its limitations. A. Mahdi, A. Elbir, and F. Karabiber observed

Digital Object Identifier 10.32754/JMT.2026.1.14

that ICA struggles with real-time separation of convolutive mixtures, and proposed Independent Vector Analysis (IVA) to overcome these challenges [4]. S. He et al. applied FastICA for locating trapped miners, demonstrating its effectiveness in complex acoustic environments where rapid separation of voices is critical [5].

Blind source separation (BSS) and Independent Component Analysis (ICA) remain widely used in modern applications. For example, J. Couchman, O. Kaparounakis, C. Samarakoon, and P. Stanley-Marbell investigated how measurement uncertainty impacts the performance of ICA in removing eyeblink artifacts from EEG signals, highlighting robustness challenges in practical BCI applications [6]. C.A. Musluoglu and A. Bertrand proposed a distributed FastICA algorithm for blind source separation in sensor networks, addressing limitations of centralized processing and demonstrating applicability to real-time, resource-constrained systems [7].

These studies illustrate the evolution of ICA-based methods for both audio and biomedical applications, as well as their adaptations to real-world signal separation challenges.

III. BSS AND ICA CONCEPTS

A. *Blind Source Separation (BSS)*

BSS refers to the problem of recovering unknown and statistically independent source signals from observed mixtures, without prior knowledge of the mixing process or the sources themselves. By assuming that the original sources are relatively independent, BSS techniques aim to estimate the mixing coefficients and, in turn, reconstruct the original signals [8]. The operating scheme of BSS is illustrated in Fig. 2, where two people speak simultaneously, and the algorithm successfully extracts the independent words originating from each speaker. When the number of sources is equal to or less than the number of observations, the problem is termed determined. Also, if the number of observations is smaller than the number of sources, the problem is underdetermined, requiring additional information or model adjustments [9]. BSS was first discussed in 1982 in the context of decoding vertebrate motor control signals, where sensor readings represent mixtures of underlying muscle activities [10].

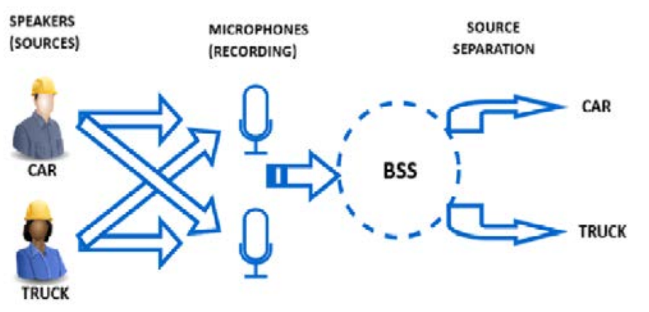

Figure 2. BSS operating scheme

BSS is an important branch of signal processing, with applications across a wide range of fields, including:

- Biomedical applications: Electroencephalography (EEG) recordings often contain signals originating from multiple brain regions, and BSS can be used to separate and identify specific signals associated with neural activity.
- Telecommunications: In noisy environments or in the presence of multiple simultaneous conversations, BSS can isolate and reconstruct individual speech signals, thereby improving call quality and user experience.
- Audio and music processing: Audio recordings may contain multiple instruments or voices, and BSS can separate each source while removing unwanted components, enhancing overall sound quality.

BSS assumes that the observed signals $x_i(t)$, presented in Fig. 3, are linear mixtures of the source signals $s_j(t)$ with unknown mixing coefficients a_{ij} [11]:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \tag{1}$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \tag{2}$$

The goal is to find a demixing matrix W so that the estimated sources $y_i(t)$ are statistically independent and resemble the original sources $s_j(t)$ [11], as in Fig. 4:

$$y_1(t) = w_{11} \cdot x_1(t) + w_{12} \cdot x_2(t) \tag{3}$$

$$y_2(t) = w_{21} \cdot x_1(t) + w_{22} \cdot x_2(t) \tag{4}$$

The signals illustrated in Fig. 3 and Fig. 4 include both the initial MATLAB-generated data and the outcomes obtained through the application of the FastICA algorithm. This method successfully extracts the independent components from the weighted mixtures. The separation process reveals the underlying original signals, namely the signal of interest (pure EEG) and the interfering signal (emotional states influencing medical interpretation). These are identified as independent components, derived solely based on the available information and the principle of statistical independence.

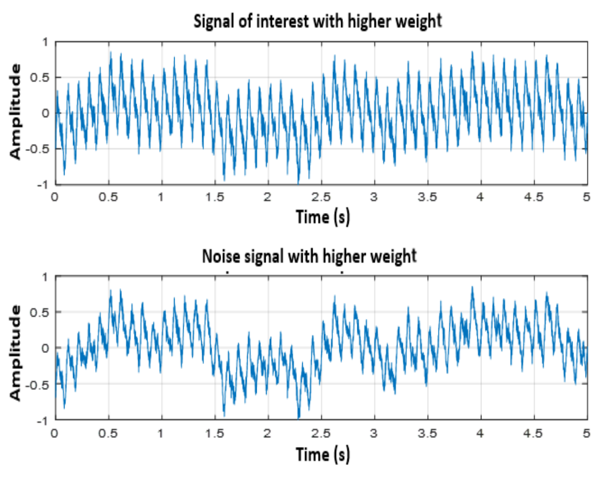

Figure 3. Recordings captured by sensors

 Digital Object Identifier 10.32754/JMT.2026.1.14

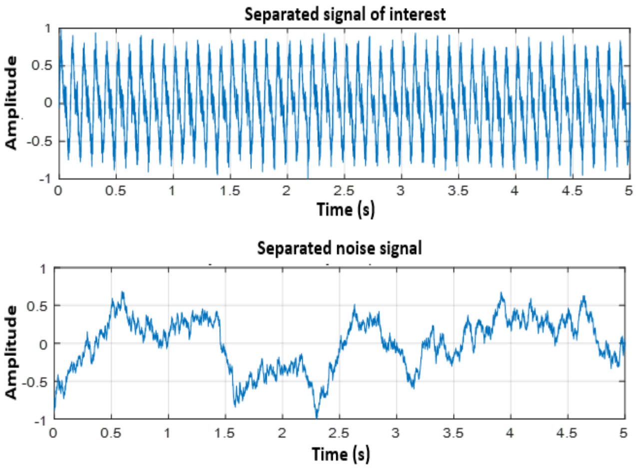

Figure 4. Signals to be extracted

B. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a computational technique in signal processing used to decompose multivariate signals into statistically independent components. The method assumes that at most one component is Gaussian and that the components are mutually independent, making ICA a specialized form of blind source separation [12]. The objective of ICA is to separate multivariate signals into non-Gaussian independent components, such as audio signals, which typically represent a sum of contributions from different sources at each time point. The main challenge lies in isolating these source signals from the observed mixtures. When the assumption of statistical independence holds, ICA can achieve nearly precise separation results [1].

The effectiveness of ICA relies on three key assumptions [11]:

- The source signals are mutually independent.
- The distributions of each source signal are non-Gaussian.
- The mixing matrix is square, implying an equal number of source and observed signals.

These assumptions give rise to three observable effects in the mixed signals [11]:

- Independence (source signals are independent, while observed signals are not).
- Normality (the Central Limit Theorem predicts that the sum of independent variables tends toward a Gaussian distribution).
- Complexity (observed signals exhibit higher temporal complexity than the original sources).

Proper implementation of these principles underpins ICA, and BSS methods based on ICA have been continuously developed and refined to achieve more effective signal separation [13].

A fundamental principle in determining the demixing matrix W in ICA is statistical independence, meaning that each recovered component y_i should provide no information about the others [11]. For example, two scalar random variables x and y are independent if knowing y does not give any information about x, which can occur in unrelated physical processes or events [11].

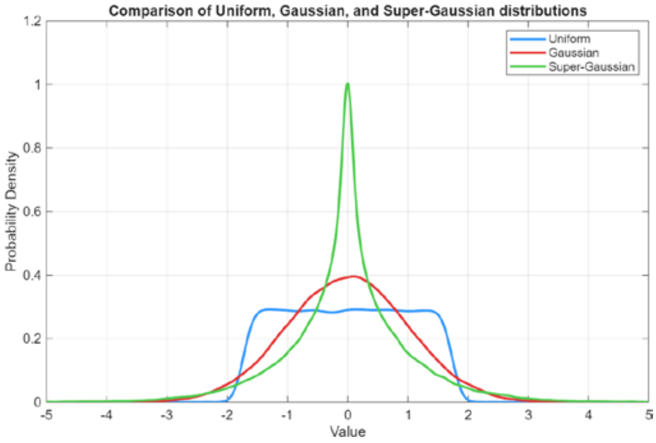

Figure 5. Comparison of Gaussian, Super-Gaussian, and Uniform Distributions

In practice, data often exhibit non-Gaussian distributions, typically super-Gaussian (Laplace), characterized by a high probability of values near zero and heavier tails compared to Gaussian distributions. This property is essential for ICA, as non-Gaussianity enables the identification of statistically independent components. Fig. 5 illustrates the differences between Gaussian, super-Gaussian and Uniform distributions, highlighting the higher likelihood of extreme values in super-Gaussian data.

Thus, non-Gaussianity and statistical independence form the basis for ICA, providing the starting point for extracting independent components from observed mixtures [11].

Based on the Central Limit Theorem, mixtures of independent signals tend to approximate Gaussian distributions, whereas the original sources are typically more non-Gaussian [2].

Signal separation in ICA is achieved by maximizing non-Gaussianity. Measures such as kurtosis and negentropy are widely used for this purpose. Kurtosis, the normalized fourth-order cumulant, indicates whether a distribution is sub-Gaussian or super-Gaussian but is sensitive to outliers [14]. Negentropy, grounded in information theory, provides a more robust measure, being zero for Gaussian variables and strictly positive otherwise. Since direct entropy estimation is computationally complex, practical approximations are derived using higher-order statistics or nonlinear [11].

By applying these principles, ICA efficiently recovers original source signals from their mixtures, with significant applications in biomedical signal processing, speech enhancement, and data analysis.

C. Non-Gaussianity and Its Maximization

Non-Gaussianity is a fundamental concept in signal processing, utilized to identify the original source signals through Independent Component Analysis (ICA). It relies on the statistical independence of the original components of the signal. In probability theory, a set of pseudo-random variables is generally assumed to follow a Gaussian distribution [1]. According to the Central Limit Theorem, the sum of independent random variables tends toward a Gaussian distribution. To put it differently, the sum of two independent variables is generally more Gaussian than the original variables themselves [11].

Digital Object Identifier 10.32754/JMT.2026.1.14

Consider the observed data matrix x modeled according to the ICA mixing model:

$$x = A \cdot s \tag{5}$$

where A contains the mixing coefficients and s represents the original source signals. By inverting this relation, the original sources can be expressed as:

$$s = A^{-1} \cdot x \tag{6}$$

To estimate an independent component, a linear combination of the observed signals is considered:

$$y = w^{\mathrm{T}} \cdot x = w^{\mathrm{T}} \cdot A \cdot s \tag{7}$$

Here, w is a weight vector that will later be optimized. This can also be written in terms of the original sources and a coefficient vector q:

$$y = w^{\mathrm{T}} \cdot x = \left(w^{\mathrm{T}} \cdot A\right) \cdot s = q^{\mathrm{T}} \cdot s \tag{8}$$

Thus, the vector $y = w^{\mathrm{T}} x$, representing an independent component, can be expressed either as a linear combination of w^{T} and x or as a linear combination of q and the original sources s. Given that the sum of independent variables is more Gaussian than the individual variables, $y = q^T s$ will be more Gaussian than any individual source component, and it becomes least Gaussian when y equals one of the original components, corresponding to a single nonzero element in q.

In practice, q is unknown, but from (8), we have $q^T s = w^T x$. This allows w to vary, and the distribution of $w^T x$ can be analyzed. By choosing a w that maximizes the non- Gaussianity of $w^T x$, the corresponding $q = A^T w$ will have a single nonzero element, identifying an independent component.

Since non-Gaussianity is processed in an n-dimensional space of the vector w, each independent component has two local maxima, $\pm w$, resulting in $2n$ local maxima in total. Maximizing non-Gaussianity leads to ideal values associated with these local maxima, representing the desired independent components. Identification of all such maxima is necessary to determine multiple components [1].

D. FastICA

Fast Independent Component Analysis (FastICA), introduced by A. Hyvärinen, J. Karhunen, and E. Oja in the 1990s, is one of the most widely adopted ICA algorithms due to its computational efficiency and robustness. It is based on a fixed-point iteration scheme that maximizes non-Gaussianity, typically using approximations of negentropy as the contrast function. Unlike gradient descent-based ICA methods, FastICA offers cubic convergence, leading to faster and more stable results [11].

Preprocessing is essential for algorithm performance and involves centering (forcing zero mean) and whitening (unit variance and decorrelation), commonly implemented via singular value decomposition (SVD). Whitening reduces the estimation problem to orthogonal matrices, significantly lowering computational complexity.

The algorithm can be applied to extract either a single component or multiple components. In the single-component case, the update rule iteratively refines the demixing vector using nonlinear functions, defined in (9), (10) and (11). For multiple components, symmetric decorrelation via SVD ensures orthogonality between weight vectors, preventing convergence to the same maximum

$$g_1(y) = y^3 \tag{9}$$

$$g_2(y) = tanh(a \cdot y) \tag{10}$$

$$g^3(y) = y \cdot e^{-y^2/2} \tag{11}$$

Convergence is defined by the stability of the demixing matrix w typically measured by the scalar product between successive iterations. FastICA's nonlinear update strategy yields faster convergence than linear methods, with improved robustness when combined with SVD-based orthogonalization.

Key properties of FastICA include:

- Cubic convergence rate, outperforming gradient-based approaches.
- Flexibility in nonlinear function selection.
- Reduced memory and computational demands, making it suitable for large datasets.
- Ability to estimate independent components individually.

Due to these advantages, FastICA is widely applied in Blind Source Separation, biomedical signal analysis, and high-dimensional data processing.

IV. System Design and Implementation

The basic structure of the FastICA algorithm [15] can be summarized in the following steps:

1) **Data acquisition:** the mixtures of signals recorded.
2) **Preprocessing:** centering and whitening of the data.
3) **Initialization:** setting the initial demixing matrix W.
4) **Update step:** iterative update of the demixing matrix W using nonlinear functions.
5) **Orthogonalization:** applying Singular Value Decomposition (SVD) to orthogonalize W.
6) **Convergence check:** verifying whether the convergence criterion is satisfied.
7) **Iteration:** if the criterion is not met, repeat steps 5–7. In case the algorithm fails to converge to acceptable results, W can be reinitialized with new random values and the whole process restarted.
8) **Output:** extraction of the independent components matrix S.

In the upgraded version of the FastICA algorithm implemented in this work, an additional mechanism was introduced to handle cases where the algorithm fails to converge satisfactorily. In such situations, the demixing matrix W is automatically reinitialized with new random values and the process restarts, which often results in improved convergence.

 Digital Object Identifier 10.32754/JMT.2026.1.14

A. Data Preprocessing: Centering and Whitening

After data acquisition, represented by observed and mixed signals, a preprocessing step is required to prepare the data for the FastICA algorithm. This preprocessing consists of two essential stages:

1) Centering the variables (the mean must be zero).
2) Whitening the variables (the variance must be unitary).

The centering step removes the mean of the observed variable *x*, resulting in a zero-mean variable x_c [15]:

$$x_c = x - E\{x\} \tag{12}$$

After ICA, the mean can be reintroduced by adding $A^{-1}E\{x\}$ to the recovered independent components [11].

Whitening transforms the centered variable x_c into x_a, such that the covariance matrix is the identity [15]:

$$E\left\{x_a x_a^{\mathrm{T}}\right\} = I \tag{13}$$

This is achieved using the eigenvalue decomposition of the covariance matrix, leading to the whitening matrix:

$$V = ED^{-1/2}E^{\mathrm{T}}, \tag{14}$$

where D is the diagonal matrix of eigenvalues and E the corresponding eigenvectors:

$$D^{-1/2} = diag\left(d_1^{-1/2}, d_2^{-1/2}, ..., d_n^{-1/2}\right) \tag{15}$$

The whitened variable is then:

$$x_{ca} = Vx = A_a s, \tag{16}$$

where A_a is an orthogonal mixing matrix. This reduces the ICA problem to the estimation of an orthogonal transformation, simplifying computations and reducing the parameter space [11].

Whitening solves half of the ICA problem by simplifying the search space to orthogonal transformations. It is therefore a fundamental preprocessing step that reduces the overall computational complexity of ICA [15].

B. Extraction of a Single Component

To extract independent components, an algorithm is required to maximize these contrast functions, such as the nonlinear functions or the approximation of negentropy.

For a single component, the FastICA algorithm maximizes the non-Gaussianity in a single projection direction $w^{\mathrm{T}}x$ to estimate the demixing vector *w*. The non-Gaussianity is measured using the negentropy approximation $J\left(w^{\mathrm{T}}x\right)$ [15].

The fixed-point iterative structure is based on a gradient-ascent approach:

$$w \leftarrow E\left\{z\, g\left(w^{\mathrm{T}}z\right)\right\} \tag{17}$$

Algebraic manipulation yields the FastICA update rule:

$$w \leftarrow E\left\{z\, g\left(w^{\mathrm{T}}z\right)\right\} - E\left\{g'\left(w^{\mathrm{T}}z\right)\right\} w \tag{18}$$

The derivatives of nonlinear functions G used to find directions that maximize non-Gaussianity are:

$$g_1(y) = 4y^3, \qquad g_1'(y) = 12y^2 \tag{19}$$

$$g_2(y) = tanh(a\, y), \quad g_2'(y) = a\left(1 - tanh(a\, y)\right) \tag{20}$$

$$g_3(y) = ye^{-y^2/2}, \qquad g_3'(y) = \left(1 - y^2\right)e^{-y^2/2} \tag{21}$$

Here, *a* is typically 1 but may range from 1 to 2 [11].

The basic FastICA algorithm for a single component is summarized in this manner:

1) Center the data to achieve *zero* mean.
2) Whiten the data to obtain the matrix *z*.
3) Initialize w randomly.
4) Update w using (18).
5) Normalize $w \leftarrow w/\,|w|$.
6) Repeat steps 4–5 until convergence

The convergence criterion relies on the modulus of the dot product between the old and new *w* vectors approaching 1.

C. Extraction of Multiple Components

To extract n components, FastICA iteratively estimates vectors w_1, w_2, ..., w_n. To prevent convergence to the same maximum, orthogonalization of $w_i^{\mathrm{T}}x$ is performed after each iteration. Symmetric decorrelation ensures orthogonality using:

$$W \leftarrow \left(W\, W^{\mathrm{T}}\right)^{-1/2} W \tag{22}$$

This is implemented via Singular Value Decomposition (SVD):

$$W\, W^{\mathrm{T}} = E\, D\, E^{\mathrm{T}} = \left(W\, W^{\mathrm{T}}\right)^{-1/2} = E\, D^{-1/2} E^{\mathrm{T}} \tag{23}$$

The FastICA algorithm for multiple components is summarized in this manner:

Center the data.

1) Whiten the data to obtain *z*.
2) Choose m, the number of independent components to estimate.
3) Initialize w_1, w_2, ..., w_m randomly.
4) For each i = 1, ..., m, update
   $$w_i \leftarrow E\left\{zg\left(w_i^T z\right)\right\} - E\left\{g'\left(w_i^T z\right)\right\} w_i$$
5) Orthogonalize *W* using SVD (22).
6) Repeat steps 5–6 until convergence.

D. Convergence

Convergence describes how an algorithm approaches a stable solution or fixed point. In FastICA, nonlinear functions are used to maximize non-Gaussianity, leading to faster convergence compared to linear methods based on gradient descent. Nonlinear convergence allows exponential progress toward the solution, achieving the desired components in fewer iterations. Orthogonalization via SVD ensures independent directions, promoting stable and rapid convergence. The convergence criterion is evaluated using the modulus of the dot product between successive updates of *W*, which must exceed a predefined threshold [15].

The threshold value must be chosen carefully depending on the application, as it represents a trade-off between faster processing and higher accuracy. A smaller threshold leads to more iterations and more precise weight computations, but if set too low, the algorithm may fail to converge [16].

V. RESULTS

To assess the functionality of the implemented codes, several pairs of audio signals were digitally mixed in MATLAB to emulate realistic microphone-captured mixtures, and an application was developed in App Designer to visualize the results. The selected test cases include:

- EEG-pure / emotions – simulating the removal of parasitic signals (e.g., blinking, emotional fluctuations) from EEG recordings to enable more accurate neurological analysis by researchers and clinicians.
- Pilot / helicopter-noise-gunfire – demonstrating separation of the pilot's speech from helicopter engine noise, surrounding explosions, and gunfire.
- Voice / music – illustrating the extraction of relevant intelligence from an environment where enemy personnel are in a state of relaxation.
- Terrorist / terrorist leader – simulating the separation of voices in enemy communications (e.g., inside a room or vehicle), enabling both identification of strategic discussions and potential localization of speakers in the operational environment.

A. EEG-pure – emotions Results:

The FastICA algorithm applied to the EEG-pure and emotion signal pair, theoretically originating from two electrodes placed on a patient's scalp during electroencephalography, aims to extract the signal corresponding strictly to brain activity, free of interfering components. This process was implemented using three nonlinear functions for updating the demixing matrix W. The best resulting separated signals are illustrated in Fig. 6, where the $G = 4{\cdot}S^3$ function was used. Also, the corresponding indices of the matrices S and W, along with the algorithm runtime for the three functions used, are reported in Table I.

The Central Limit Theorem states that the distribution of the sum of multiple signals tends to be more Gaussian than the distribution of any of the original signals. This property is evident in Fig. 6, where the distributions of the combined signals *X_combinat* and *Y_combinat* are shown to be more Gaussian compared to the initial signals *X_initial* and *Y_initial*.

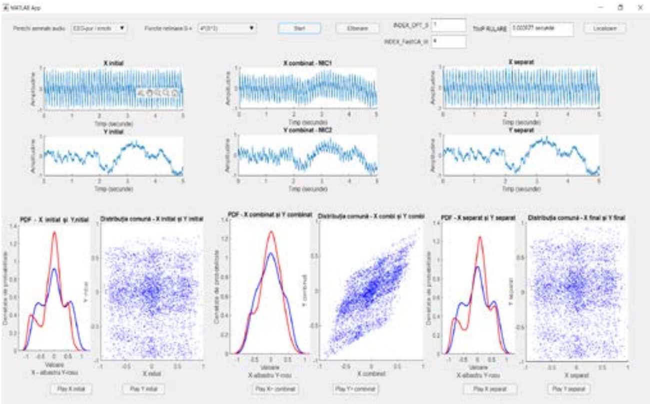

Figure 6. The FastICA algorithm applied to the 'EEG-pure / emotions' signal pair using the nonlinear function $G = 4S^3$

TABLE I. RESULTS FOR THE SIGNAL PAIR "EEG-PURE – EMOTIONS"

Nonlinear function	S Matrix Iter.	W Matrix Iter.	Runtime (s)
$G = 4S^3$	1	4	0.0036
$G = tanh(S)$	1	5	0.0047
$G = S \cdot exp(-S^2/2)$	1	6	0.0069

In Fig. 6, the upper plots display the three pairs of signals in their initial, combined, and separated states, whereas the lower plots present the corresponding probability density functions (PDF) alongside the joint distributions for each pair.

From Table I, it can be observed that the source component matrix S satisfied the imposed convergence criteria from the very first iteration for each nonlinear function used, making repeated execution of the optimization loop unnecessary. The only differences arise in the number of iterations required by the demixing matrix W and in the overall execution time of the algorithm.

For this model of mixed signals, the nonlinear function $G = 4{\cdot}S^3$ achieved the fastest separation after 4 iterations in 0.0036 seconds, followed by $G = tanh(S)$ with 5 iterations in 0.0047 seconds, and finally $G = S \cdot exp(-S^2/2)$ with 6 iterations in 0.0069 seconds, the latter proving less suitable for this case.

In all three cases, however, the separation was performed effectively, with only minor and negligible differences among the algorithms.

B. Pilot – Helicopter Results

The following signal pair, shown in Fig. 7, simulates a military helicopter pilot, whose voice is sequentially separated from both the aircraft noise and surrounding gunfire and explosions.

The same three functions are applied, with a summary of the results shown in Table II, which indicates that the number of iterations required to optimize matrix S is consistently 1, with the FastICA algorithm reaching convergence on the first run for all three nonlinear functions.

Differences among algorithms appear in the iterations needed to update the matrix W. Here, the efficiency order is reversed compared to the previous scenario.

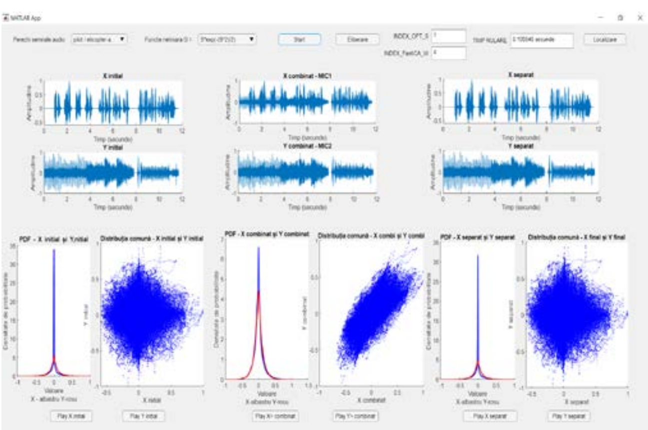

Figure 7. The FastICA algorithm applied to the 'Pilot / Helicopter' signal pair using the nonlinear function $G = S \cdot exp(-S^2/2)$

 Digital Object Identifier 10.32754/JMT.2026.1.14

TABLE II. RESULTS FOR THE SIGNAL PAIR "PILOT / HELICOPTER"

Nonlinear function	*S* Matrix Iter.	*W* Matrix Iter.	Runtime (s)
$G = 4S^3$	1	8	0.244
$G = tanh(S)$	1	7	0.144
$G = S \cdot exp(-S^2/2)$	1	4	0.106

The function corresponding to the third function in Table II is the most efficient, requiring 4 iterations and 0.106 seconds, followed by the second function with 7 iterations and 0.144 seconds, and lastly the first one with 8 iterations and 0.244 seconds.

C. Voice – Music Results

The separation of an audio signal containing voice from background music was analyzed to enhance vocal clarity and extract relevant information. Fig. 8 illustrates this signal pair, showing that the PDFs of the combined signals tend to a Gaussian distribution after mixing. Table III summarizes the results and efficiency of the nonlinear functions used.

In all three cases, FastICA achieved convergence in a single iteration, successfully separating the independent components on the first attempt. The $G = S \cdot exp(-S^2/2)$ function performed best, corresponding to Fig. 8, requiring 4 iterations and 0.049 seconds, followed by the $G = tanh(S)$ function with 5 iterations (0.057 s) and the $G = 4S^3$ function with 6 iterations (0.081 s).

All three nonlinear functions matched the mixing models well, providing optimal separation and high-quality results, confirmed by audio playback.

The separation of overlapping voice and music signals using FastICA has potential military applications in surveillance, communications intelligence, and battlefield monitoring.

By isolating individual voices from background noise or music, the technique can enhance speech intelligibility in intercepted communications, support automated threat detection in crowded environments, and improve the clarity of audio captured by unmanned systems or reconnaissance devices. Additionally, it can be employed in training simulations to evaluate auditory scene analysis under complex signal conditions.

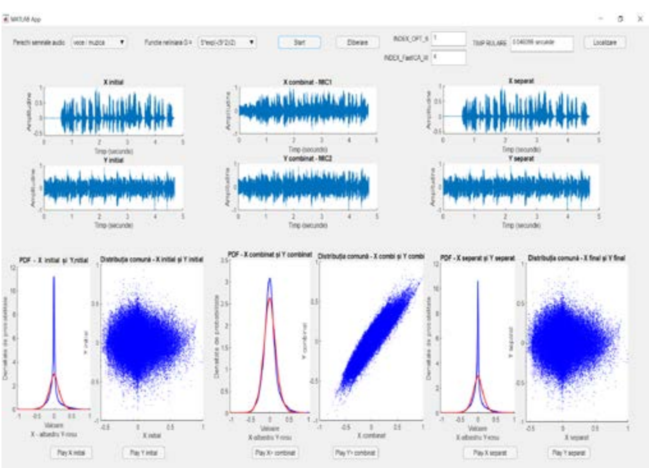

Figure 8. The FastICA algorithm applied to the 'Voice / Music' signal pair using the nonlinear function $G = S \cdot exp(-S^2/2)$

TABLE III. RESULTS FOR THE SIGNAL PAIR "VOICE / MUSIC"

Nonlinear function	S Matrix Iter.	W Matrix Iter.	Runtime (s)
$G = 4S^3$	1	6	0.081
$G = tanh(S)$	1	5	0.057
$G = S \cdot exp(-S^2/2)$	1	4	0.049

D. Terrorist – Terrorist Leader Results

This particular separation case demonstrates a key application of FastICA: separating two voices to extract independent information and estimate source positions based on microphone weights.

Fig. 9 and 10 illustrate the separation processes for two different cases using the nonlinear functions $G = tanh(S)$ and $G = S \cdot exp(-S^2/2)$. Combined signal PDFs decrease in amplitude according to the Central Limit Theorem, with simulation results summarized in Table IV.

To evaluate the separation performance of the proposed functions, two microphones (MIC1 and MIC2) were employed, each positioned to capture different proportions of the original input signals. The principle of localization was used to determine the dominant contribution of each source to the recorded signals. In this setup, MIC1 and MIC2 do not capture the sources in isolation but rather mixtures with varying weights, which can then be analyzed through the separation and optimization functions.

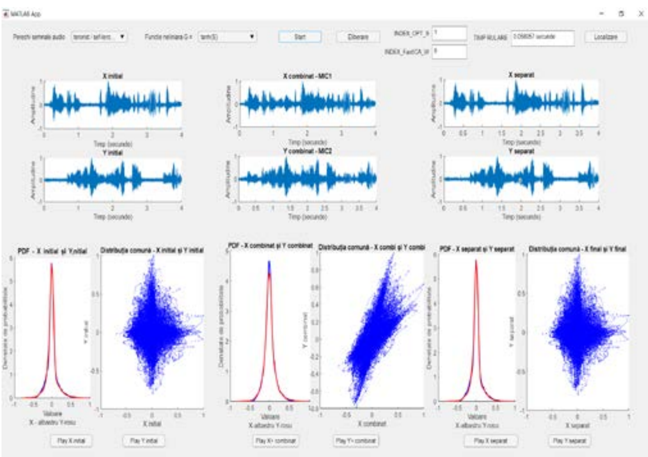

Figure 9. The FastICA algorithm applied to the 'Terrorist / Terrorist Leader' signal pair using the nonlinear function $G = tanh(S)$

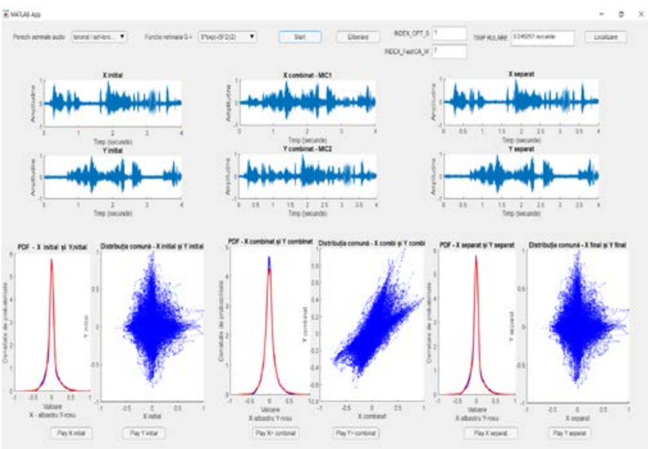

Figure 10. The FastICA algorithm applied to the 'Terrorist / Terrorist Leader' signal pair using the nonlinear function $G = S \cdot exp(-S^2/2)$

Digital Object Identifier 10.32754/JMT.2026.1.14

By observing the localization window, it becomes possible to identify which microphone is more strongly associated with a given source, providing a clear indication of how the separation functions distribute the original signals across the two channels.

In the graphical representations, the first signal is denoted by the color green and the second signal by the color purple, enabling a clear and immediate identification of their respective locations, within Fig. 11 and Fig. 12 for example.

In this way, localization can be achieved, even in real time, within a room where multiple people are engaged in a conversation.

In the first two separation functions, including the one with $G = tanh(S)$, the first microphone (MIC1) predominantly captures the *X_initial* signal, while the second microphone (MIC2) primarily records the *Y_initial* signal, as illustrated in the localization window inf Fig. 11. For the final function used in optimizing the matrix *W*, $G = S \cdot exp\left(-S^2/2\right)$, this distribution is reversed: MIC1 captures a greater proportion of *Y_initial*, whereas MIC2 records more of *X_initial*, as shown in Fig. 12.

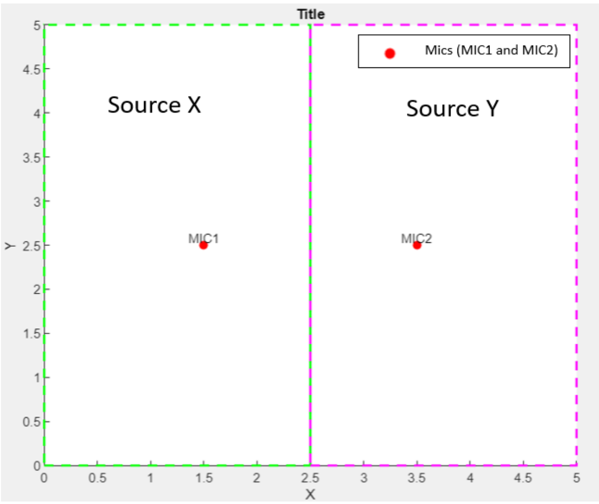

Figure 11. Source localization for the cases where the functions $G = 4S^3$ and $G = tanh(S)$ were used

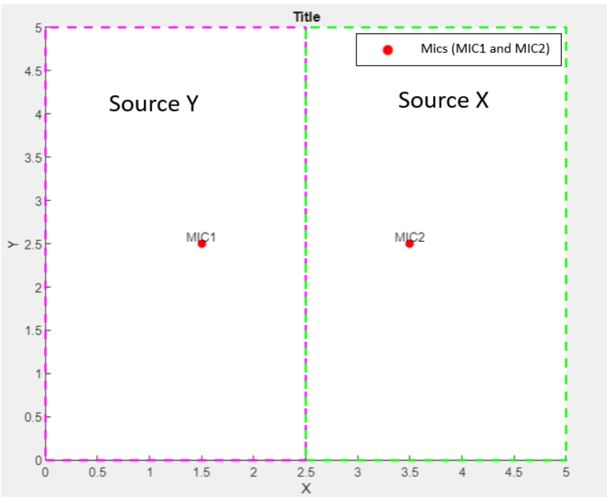

Figure 12. Source localization for the cases where the functions $G = 4S^3$ and $G = S \cdot exp(-S^2/2)$ was used

TABLE IV. RESULTS FOR THE SIGNAL PAIR "TERRORIST / TERRORIST LEADER"

Nonlinear function	*S* Matrix Iter.	*W* Matrix Iter.	Runtime (s)
$G = 4S^3$	1	9	0.096
$G = tanh(S)$	1	8	0.058
$G = S \cdot exp(-S^2/2)$	1	7	0.046

Table IV shows that the third function, used in Fig. 10, is the most efficient, requiring 7 iterations and 0.046 seconds, while the first function is the slowest, with 8 iterations and 0.058 seconds and the second function, from Fig. 9, requires 9 iterations and 0.096 seconds. All nonlinear functions are suitable for these signals, yielding perfect final separation.

E. Cocktail Party Situation (3 signals) Results

In the case of the Cocktail Party scenario with three signals (the voice of a man, the voice of a woman and music), the separation and localization process was carried out following the same methodology as in the previous examples. The initial signals are presented in Fig. 13, while the mixed signals are shown in Fig. 14. In these mixtures, signal *X* dominates in the first case, signal *Y* in the second, and signal *Z* in the third. The results are presented in Fig. 15, where each separated signal, using the $G = 4S^3$ function, is shown. This is not the most suitable nonlinear function, as evidenced in Table V.

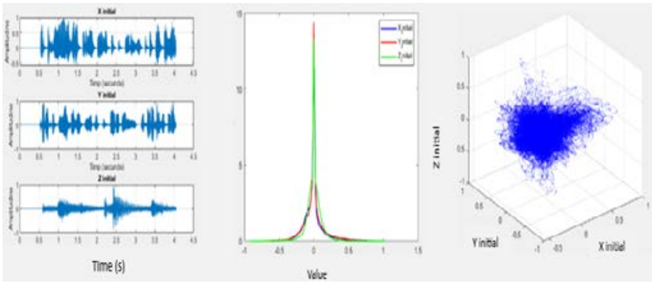

Figure 13. Initial signals from 3 sources

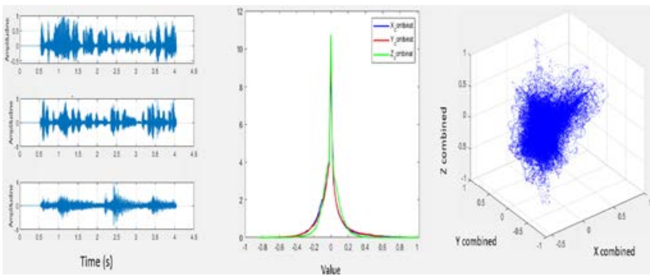

Figure 14. The resulted combined signals

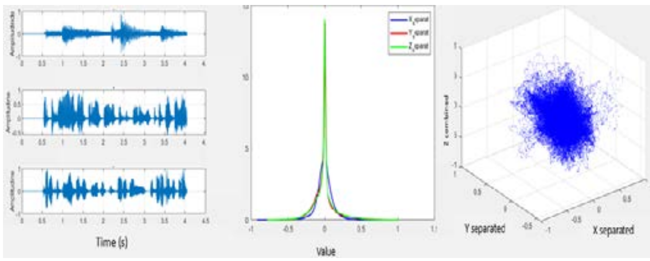

Figure 15. Results of the FastICA algorithm applied to the three signals using the function $G = 4S^3$

 Digital Object Identifier 10.32754/JMT.2026.1.14

TABLE V. RESULTS FOR THE SIGNAL PAIR "MAN / WOMAN / MUSIC"

Nonlinear function	S Matrix Iter.	W Matrix Iter.	Runtime (s)
$G = 4S^3$	18	19	13.58
$G = tanh(S)$	6	17	0.81
$G = S \cdot exp(-S^2/2)$	27	19	27.12

The comparison shows that the $G = tanh(S)$ function provides the best performance, achieving accurate separation with the lowest runtime (0.81 seconds). In contrast, the cubic function $G = 4S^3$ and the Gaussian function $G = S \cdot exp\left(-S^2/2\right)$ require considerably more iterations for convergence and result in significantly higher runtimes (13.58 and 27.12 seconds, respectively). These findings indicate that the choice of nonlinearity directly influences both efficiency and computational cost in the FastICA algorithm.

This representation evaluates the algorithm's ability to separate overlapping signals in a realistic setting.

The algorithm for audio source localization was modified so that the three microphones were placed on the circumference of an ellipse at equal distances, with each of the three sources being assigned a specific area of microphone coverage, as illustrated in Fig. 16.

As observed in Fig. 13 and 15, the separated signals do not always correspond to the positions of the original signals. For example, the first separated component corresponds to the last original signal, the second separated component matches the first original signal, and the third separated component corresponds to the second initial signal. Also, the position of the sources in the room is different now, as it can be seen in Fig. 16. This reflects that FastICA optimizes for statistical independence and non Gaussianity rather than preserving the original signal order.

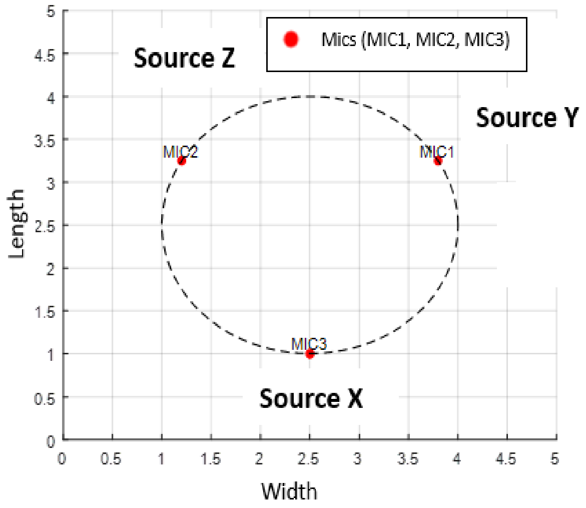

Figure 16. Source localization for the case where the function $G = 4S^3$ was used

VI. CONCLUSIONS AND FUTURE WORK

This work highlights the effectiveness and limitations of blind audio source separation using Independent Component Analysis (ICA) in both civilian and military contexts. The study demonstrates that careful selection and recording of audio signals is critical, as ICA assumes linear, independent sources, and real-world conditions, such as reverberation, propagation delays, and source movement, can significantly affect separation performance. FastICA was implemented with three different nonlinear functions, and results indicate that the choice of nonlinearity strongly influences convergence speed and computational efficiency. Additionally, source localization based on Euclidean distances between separated and sensor signals proved feasible, supporting applications in surveillance, communications monitoring, and battlefield scenarios. Limitations were observed in scenarios involving convolutive or nonlinear mixtures, as well as in the initialization of the demixing matrix. Despite these challenges, the work confirms that FastICA provides reliable separation in controlled simulations and establishes a foundation for future research on real-time and nonlinear source separation applications.

REFERENCES

[1] P. Comon and C. Jutten, "Handbook of Blind Source Separation - Independent Component Analysis and Applications," Academic Press, Oxford, UK, February, 2010.

[2] A. Hyvärinen, "Independent component analysis: recent advances," Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci., vol. 371, no. 1984, pp. 20110534, Feb. 2013.

[3] W. Du, H. Li, X.-L. Li, V. D. Calhoun, and T. Adali, "ICA of fMRI data: Performance of three ICA algorithms and the importance of taking correlation information into account," in Proc. IEEE International Symposium on Biomedical Imaging (ISBI), Chicago, IL, 2011, pp. 3025–3030. doi: 10.1109/ISBI.2011.5872582

[4] A. Mahdi, A. Elbir, and F. Karabiber, "Blind audio source separation using independent component analysis and independent vector analysis," Int. J. Appl. Math. Electron. Comput., vol. 4, no. Special Issue, pp. 174–177, 2016.

[5] S. He, Z. Tong, M. Tong, S. Tang, M. Li, and L. Liang, "Research on sound separation and identification of trapped miners based on FastICA algorithm," Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), pp. 3025–3030, Macau, 2017. doi: 10.1109/ICIT.2017.7915220.

[6] J. Couchman, O. Kaparounakis, C. Samarakoon, and P. Stanley-Marbell, "Simulated Eyeblink Artifact Removal with ICA: Effect of Measurement Uncertainty," arXiv preprint arXiv:2410.03261, 2024. [Online].

[7] C. A. Musluoglu and A. Bertrand, "Distributed Blind Source Separation Based on FastICA," IEEE Signal Processing Letters, vol. 32, pp. 881–885, 2025. doi: 10.1109/LSP.2025.3540370

[8] F. Totir, G. Vasile, M. Gay, L. Anton, and G.-A. Ilie, "ICA-based information extraction method for PolSAR images," in Proc. 8th Int. Conf. Commun. (COMM), Bucharest, Romania, Jun. 2010, pp. 149–152.

[9] M. H. Moradi, A. Nouri, Z. Ghanbari, and M. R. Aslani, "A new approach to feature extraction in MI-based BCI systems," in Artificial Intelligence-Based Brain-Computer Interface, Elsevier, 2022, pp. 75-98, doi: 10.1016/B978-0-323-91197-9.00002-3

[10] S. N. Jain, "Blind source separation and ICA techniques: A review," Technical Report, Vidyanagar, India, 2012.

[11] A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis, Wiley-Interscience, 2001.

[12] B. Ans, J. Hérault, and C. Jutten, "Architectures neuromimétiques adaptatives : Détection de primitives," in Cognitiva 85, vol. 2, pp. 593–597, Paris, 1985.

[13] Y. Deville, "Blind source separation and blind mixture identification methods," Wiley Encyclopedia of Electrical and Electronics Engineering, Wiley, 2016. doi: 10.1002/047134608x.w8300

[14] A. R. Alnassar, "Measures of Shape: Skewness and Kurtosis," 2020, College of Science, Atmospheric Science Dept., Mustansiriyah University.

[15] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," Neural Networks, vol. 13, no. 4-5, pp. 411–430, 2000. doi: 10.1016/S0893-6080(00)00026-5.

[16] L. Tuță, G. Roșu, C. Popovici, and I. Nicolaescu, “Real-Time EEG Data Processing Using Independent Component Analysis (ICA),” in Proc. IEEE 14th Int. Conf. Commun. (COMM), Bucharest, Romania, Jul. 2022, doi: 10.1109/COMM54429.2022.9817209