Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

AN ALGORITHM FOR INDEPENDENT COMPONENT ANALYSIS USING A GENERAL CLASS OF COPULA-BASED DEPENDENCE CRITERIA

F. Asadi, H. Torabi ✉, and H. Nadeb

Article type: Research Article
(Received: 28 February 2024, Received in revised form 05 September 2024)
(Accepted: 13 December 2024, Published Online: 18 December 2024)

Abstract. The efficiency of Independent Component Analysis (ICA) algorithms relies heavily on the choice of objective function and optimization algorithms. The design of objective functions for ICA algorithms necessitate a foundation built upon specific dependence criteria. This paper will investigate a general class of dependency criteria based on the copula density function. One of the aims of this study is to characterize the independence between two random variables and investigate their properties. Additionally, this paper introduces a novel algorithm for ICA based on estimators derived from the proposed criteria. To compare the performance of the proposed algorithm against existing methods, a Monte Carlo simulation-based approach was employed. The results of this simulation revealed significant improvements in the algorithm's outputs. Finally, the algorithm was tested on a batch of time series data related to the international tourism receipts index. It served as a pre-processing procedure within a hybrid clustering algorithm alongside `PAM`. The obtained results demonstrated that the utilization of this algorithm led to improved performance in clustering countries based on their international tourism receipts index.

Keywords: Amari error, Clustering, Copula, Dependence criteria, Independent components analysis, Mutual information.
2020 MSC: Primary 62H20, 62H25.

1. Introduction

The extraction of efficacious components from a dataset is a primary objective when employing Independent Component Analysis (ICA). This technique proves advantageous across a variety of datasets, including those related to sound, stock markets, and images. ICA is utilized to identify independent components comprised of diverse elements. According to Sheikh and Regan [33], This approach is applied in network analysis for the detection and prediction of traffic incidents. Based on Lassance et al. [17], it also has been used in finance for optimal portfolio diversification.

✉ htorabi@yazd.ac.ir, ORCID: 0000-0002-4200-9770
https://doi.org/10.22103/jmmr.2024.23031.1591
Publisher: Shahid Bahonar University of Kerman
How to cite: F. Asadi, H. Torabi, H. Nadeb, *An algorithm for independent component analysis using a general class of copula-based dependence criteria*, J. Mahani Math. Res. 2025; 14(1): 527-550.
© the Author(s)

In the realm of macroeconomics, ICA has been instrumental in identifying structural autoregressive models, as evidenced by Moneta and Pallante [23]. It has also been applied in the analysis of magnetotelluric sounding data, as shown by Zhou et al. [39], and in the clustering of wireless sensor network data, as demonstrated by Shahina and Pradeep Kumar [31].

In the field of image processing, ICA has been used for feature extraction, as shown by Shang et al. [32]. It has also found extensive application in various medical fields, including the analysis of brain activities, as demonstrated by Tabanfar et al. [35], Meng et al. [22], and Jayabal et al. [14]. In the study of Electroencephalogram (EEG) data, ICA has been employed by Lyu and Fu [21] and Antony et al. [3]. It has also been used in the analysis of individual molecular characterization, as shown by Rincourt et al. [29], and in the dissociation of biologically single-layer networks, as demonstrated by Lipshutz et al. [20]. In summary, ICA is a versatile tool that has been applied across a wide range of fields, demonstrating its utility in extracting independent components from diverse datasets.

In this study, a novel class of dependency criteria based on a generalized density-based dependency measure and its copula-based variant has been proposed. The proposed criteria are argued to characterize the independence of two random variables effectively. Furthermore, an ICA algorithm has been developed, which is grounded on an estimator derived from the proposed dependency criteria. To assess the efficacy of the introduced methods, they have been benchmarked against several well-established algorithms. Additionally, the proposed method has been applied to a batch of time series data as a preprocessing step in clustering, to demonstrate its practical utility in real-world scenarios.

This study is organized into nine sections to comprehensively explore its subject matter. Section 2 delves into the conceptual framework of ICA. Section 3 critically examines various ICA approaches centered on the minimization of mutual information (MI). The subsequent section (Section 4) introduces a generalized density-based dependency criterion alongside its copula-based iteration. Section 5 focuses on the exploration of the ICA estimator within two specific scenarios. In Section 6, a novel ICA algorithm is presented, proposing an innovative approach to the subject. Section 7 then evaluates the performance of this suggested methodology through the utilization of Monte Carlo simulations, employing the average of Amari errors as an evaluative metric. Section 8 utilizes data encompassing international tourism receipts collected between 2000 and 2020 from diverse countries to illustrate the methodology's application in clustering analysis. Finally, Section 9 concludes this investigation, offering pertinent insights and conclusions derived from the study's findings.

2. Independent Component Analysis

In this investigation, we delve into the concept of ICA. For this study, consider that $\boldsymbol{x_i} = (x_{1i}, x_{2i}, \ldots, x_{ni})^T, i = 1, 2, ..., d$ is an n-dimensional vector. Additionally, a multivariate dataset $X = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_d})$ is represented as an $n \times d$ matrix, which is a mixture of unknown independent components. Here, n denotes the number of mixtures, and d signifies the number of dependent components. Each component is time-varying and is defined as $\boldsymbol{s_i} = (s_{1i}, s_{2i}, \ldots, s_{ni})^T, i = 1, 2, ..., d$, where n is the number of time steps, and s_{ij} is the time j of components $\boldsymbol{s_i}$. Thus, an $n \times d$ matrix with independent components is defined as $S = (\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_d})$.

Based on these definitions, the ICA model can be mathematically represented as $X = SA + \varepsilon$, where A is a $d \times d$ mixing coefficients matrix and ε is a $n \times d$ residual matrix with $E[\varepsilon] = 0$. Note that in the ICA model, both matrices S and A are unknown and the goal of ICA is to estimate A to get the estimation of S. Let $\hat{A}$ and $\hat{S}$ be the estimations of A and S, respectively. Since $E[\varepsilon] = 0$, thus we should have $X \simeq \hat{S}\hat{A}$. Therefore, the goal of ICA is to find a separating matrix $W = \hat{A}^{-1}$, such that the reconstructed matrix is defined as $\hat{S} = XW$.

In the process of modeling, the steps of centralization and whitening should be applied before ICA. This investigation reveals that when centralized data undergoes whitening, it can be defined by a whitening matrix denoted as $Z = QX_c$ computed with $Q = \Lambda^{-1}D'$, where Λ represents a diagonal matrix and D denotes an eigenvector matrix calculated from the covariance of X_c. So, after the whitening phase, the resulting transformation is defined as $Z = QX_c = QSA$.

The estimation of independent components is achieved through various methods, as expounded in the scientific literature. One noteworthy approach, investigated by Hyvärinen and Oja [13], involves maximizing non-Gaussianity through negative entropy. Additionally, an alternative method entails estimating the W matrix using the maximum likelihood approach, offering diverse avenues for the extraction of independent components. Moreover, the minimization through Mutual Information (MI) criterion, as proposed by Hyvärinen [14], has emerged as one of the most widely adopted methods in this domain. Consequently, numerous algorithms designed for ICA are rooted in the principle of minimizing MI to facilitate independent components estimation, as emphasized by Langlois et al. [16].

Some of the most useful and applicable algorithms include the FastICA algorithm based on MI minimization (Hyvärinen [14]), the Infomax algorithm relying on the maximum likelihood method (Lee et al. [19]), the JADE algorithm founded on maximizing kurtosis (Cardoso [6]), RADICAL algorithm based on the Kullback-Leibler criterion (Learned-Miller and Iii [18]), HICA algorithm utilizing the copula function of Hoeffding's criterion (Rahmanishamsi

et al. [27]), and RLICA algorithm employing the copula function of the squared loss MI criterion (Rahmanishamsi and Dolati [26]).

3. Some ICA Approaches Based on Minimizing MI

In this section, various ICA approaches centered on the minimization of Mutual Information (MI) are explored. It is deduced that when a dependency criterion characterizes the independence of random variables, it proves to be highly fruitful. To illustrate this, consider X_1 and X_2 as two random variables with the joint distribution function F, joint density function f, and marginal density functions f_1 and f_2, respectively. The MI criterion between X_1 and X_2 is defined by the following equation:

$$\mathrm{MI}(X_1, X_2) = \iint \log\left(\frac{f(x_1, x_2)}{f_1(x_1)\, f_2(x_2)}\right) \mathrm{d}F(x_1, x_2).$$

As the Mutual Information (MI) value approaches zero under conditions of variable independence, in ICA, it is observed that the matrix W tends to be close to zero (Hyvärinen [14]).

In the ICA approach, where reliance on the MI criterion is pivotal, achieving precise MI estimation is of utmost importance. However, this is a challenging and time-consuming process. Moreover, it has been observed that this intricate estimation procedure can diminish the accuracy of independent component estimation. Recognizing these challenges, Suzuki and Sugiyama [34] have proposed the LICA algorithm, which is founded on a squared-loss variant of MI denoted as $\mathrm{I_s}$. The formulation of $\mathrm{I_s}$ is represented by the following equation, designed to address some of the aforementioned challenges:

$$\mathrm{I_s}(X_1, X_2) = \frac{1}{2} \iint \left(\frac{f(x_1, x_2)}{f_1(x_1)\, f_2(x_2)} - 1\right)^2 f_1(x_1) f_2(x_2) \mathrm{d}x_1 \mathrm{d}x_2,$$

where the estimation of the ratio $\frac{f(x_1,x_2)}{f_1(x_1)f_2(x_2)}$, denoted as the estimation of $\mathrm{I_s}$, serves to alleviate the challenging task of density estimation.

Furthermore, Chen et al. [9] delved into the COPICA method for exploring independent components. Copulae were employed in the p-dimensional data model, and independent components were identified by optimizing the copula parameters. Additionally, Abayomi et al. [1] proposed an alternative copula approach with the objective of orthogonalizing a measure of multivariate dispersion, resulting in an orthogonal basis for a multivariate data set. This study demonstrated that the flexibility of the copula approach permits parameterizations of non-Gaussian, non-monotone dependence, as defined below:

$$\mathrm{MI}_C(X_1, X_2) = \iint \log(c(u, v)) \mathrm{d}C(u, v),$$

where C denotes the copula corresponding to the joint distribution function F and $c(u, v) = \frac{\partial C(u,v)}{\partial u \partial v}$ represents its density.

Moreover, Keziou et al. [15] introduced a novel blind source separation approach based on the modified Kullback-Leibler divergence between copula densities, applicable to both independent and dependent source component signals. This method presents a significant advantage in its inherent adaptability to the separation of mixtures involving dependent source components.

Additionally, Rahmanishamsi and Dolati [26] introduced a dependency criterion called squared-loss Mutual Information (SMI), denoted by SMI, by a combination of ideas from MI and $\mathrm{I_s}$, which is defined by the equation below:

$$\mathrm{SMI}(X_1, X_2) = \iint \left(\frac{f(x_1, x_2)}{f_1(x_1)\, f_2(x_2)} - 1 \right)^2 f_1(x_1) f_2(x_2) \mathrm{d}x_1 \mathrm{d}x_2,$$

with its copula-based version, denoted by SMI^C defined as below:

$$\mathrm{SMI}^C(X_1, X_2) = \int_0^1 \int_0^1 \left(c(u, v) - 1\right)^2 \mathrm{d}u\mathrm{d}v. \tag{1}$$

In these cases, researchers no longer encounter the challenge of estimating the marginal and joint density functions; instead, it is only necessary to estimate the copula density function.

4. **Proposed Dependency Criterion**

This section introduces a broad class of dependence criteria. Notably, when the criteria equal zero, they signify the independence of two random variables. To illustrate this, let's consider two random variables denoted by X_1 and X_2, with a joint distribution function denoted by F. The marginal distribution functions are defined as F_1 and F_2. The following condition is assumed:

- A1 : $g : \mathbb{R}^+ \to \mathbb{R}^+$, is decreasing for $x < 1$, increasing for $x \geq 1$, and $g(1) = 0$.
- A2 : $H(x_1, x_2)$ is an arbitrary absolutely continuous joint distribution function.

Under this framework, a generalized density-based dependency criterion, represented by $GDDC_{g,H}$, is derived from the following equations:

$$GDDC_{g,H}(X_1, X_2) = \iint g\left(\frac{f(x_1, x_2)}{f_1(x) f_2(x)} \right) \mathrm{d}H(x_1, x_2).$$

It is evident that when the random variables X_1 and X_2 are independent, these divergences disappear. Let's denote C as a copula function corresponding to the joint distribution function F with a copula density function c, and C^* represent an arbitrary absolutely continuous copula. Therefore, the copula-based version of $GDDC_{g,H}$, denoted by $GDDC_{g,C^*}$, is proposed as

$$GDDC_{g,C^*}(X_1, X_2) = \int_0^1 \int_0^1 g\left(c(u, v)\right) \mathrm{d}C^*(u, v). \tag{2}$$

In this context, two specific cases for C^* can be considered, namely $C^*(u,v) = C(u,v)$ and the independent copula $C^*(u,v) = \Pi(u,v) = uv$. As demonstrated in Equations (1) and (2), it is clear that if $g(x) = (x-1)^2$, then $GDDC_{g,\Pi}(X_1, X_2) = \mathrm{SMI}^C(X_1, X_2)$.

Theorem 4.1. *Let X_1 and X_2 be two continuous random variables where are linked through a copula C with the corresponding density c, and C^* be an arbitrary absolutely continuous copula.*

(i) *$GDDC_{g,C^*}(X_1, X_2) = 0$, if and only if X_1 and X_2 are independent.*
(ii) *$GDDC_{g,\Pi}(X_1, X_2) = GDDC_{g,\Pi}(X_2, X_1)$.*
(iii) *$GDDC_{g,C}$ and $GDDC_{g,\Pi}$ are invariant under monotone transformations.*

Proof. (i) Let X_1 and X_2 be independent. Thus, the copula function C corresponding to (X_1, X_2) is of the form $C(u,v) = uv$ with the density copula function $c(u,v) = \frac{\partial^2 C(u,v)}{\partial u \partial v} = 1$ for all $(u,v) \in [0,1]^2$. Therefore, Condition A1 implies that $GDDC_{g,C^*}(X_1, X_2) = 0$. Conversely, suppose that $GDDC_{g,C^*}(X_1, X_2) = 0$ and $A \subseteq [0,1]^2$ be a set such that $\lambda^2(A) = 1$, where λ^2 denotes a two-dimensional Lebesgue measure on $[0,1]^2$. By the nature of copula, we know that $C^*(u,v)$ is a bivariate continuous distribution function on $[0,1]^2$. On the other hand $g(x) \geq 0$ for all $x \in \mathbb{R}^+$. Thus, from $\int_0^1 \int_0^1 g\left(c(u,v)\right) \mathrm{d}C^*(u,v) = 0$, we conclude that $g\left(c(u,v)\right) = 0$ for all $(u,v) \in A$. Therefore, it implies that $c(u,v) = 1$ for all $(u,v) \in A$. Hence, the proof is completed.

(ii) The desired result clearly holds.

(iii) To prove we use Theorem 2.4.4 of Nelsen [25]. Let $\psi_1(X_1)$ and $\psi_2(X_2)$ be two transformations on X_1 and X_2. If ψ_1 and ψ_2 are both increasing real-valued functions, then $C_{\psi_1(X_1),\psi_2(X_2)}(u,v) = C(u,v)$, which implies $c_{\psi_1(X_1),\psi_2(X_2)}(u,v) = c(u,v)$. If ψ_1 and ψ_2 are both decreasing real-valued functions, then $C_{\psi_1(X_1),\psi_2(X_2)}(u,v) = u+v-1+C(1-u,1-v)$, which implies $c_{\psi_1(X_1),\psi_2(X_2)}(u,v) = c(1-u,1-v)$. If ψ_1 is increasing and ψ_2 is decreasing, then $C_{\psi_1(X_1),\psi_2(X_2)}(u,v) = v - C(u,1-v)$, which implies $c_{\psi_1(X_1),\psi_2(X_2)}(u,v) = c(u,1-v)$. Also, if ψ_1 is decreasing and ψ_2 is increasing, then $C_{\psi_1(X_1),\psi_2(X_2)}(u,v) = u - C(1-u,v)$, which implies $c_{\psi_1(X_1),\psi_2(X_2)}(u,v) = c(1-u,v)$. By substituting in (2), we have

$$GDDC_{g,C}(\psi_1(X_1), \psi_2(X_2)) = GDDC_{g,C}(X_1, X_2),$$

$$GDDC_{g,\Pi}(\psi_1(X_1), \psi_2(X_2)) = GDDC_{g,\Pi}(X_1, X_2),$$

where complete the proof.

□

5. Estimation of the Proposed Dependency Criterion

To propose a dependency criterion estimation, we need a random sample. So, let $(X_{11}, X_{21}), \ldots, (X_{1n}, X_{2n})$ denote a random sample of size n from a bivariate distribution function F with the associated copula C and the corresponding copula density c. For $i = 1, 2$, $j = 1, \ldots, n$, let $R_{ij} = \sum_{k=1}^{n} \mathbb{I}_{[Xik \leq X_{ij}]}$ be the rank of X_{ij}, where $\mathbb{I}_{[.]}$ is the indicator function. Then $U_{ij} = \frac{R_{ij}}{n+1}$ is defined based on this argument. In this arrangement, we obtain the random sample $(U_{11}, U_{21}), \ldots, (U_{1n}, U_{2n})$. It is noteworthy that using $n + 1$ in the denominator of U_{ij} ensures that U_{ij} is in the range of 0 to 1. Moreover, based on Wand and Jones [36], the kernel estimator of the copula density c is given by the following equation with bandwidth $h > 0$, where k is a given bivariate kernel:

$$\widehat{c}(u, v) = \frac{1}{nh^2} \sum_{j=1}^{n} k\left(\frac{u - U_{1j}}{h}, \frac{v - U_{2j}}{h}\right), \quad (u, v) \in [0, 1]^2.$$

Chen [8] introduced the univariate beta kernel density estimator to estimate a density function with compact support. Charpentier et al. [7] used this idea to estimate a bivariate copula density as the product of the beta kernels as follows:

$$\begin{aligned} \widehat{c}(u, v) &= \frac{1}{nh^2} \sum_{j=1}^{n} \kappa\left(U_{1j}; \frac{u}{h} + 1, \frac{1 - u}{h} + 1\right) \\ &\quad \times \kappa\left(U_{2j}; \frac{v}{h} + 1, \frac{1 - v}{h} + 1\right), \quad (u, v) \in [0, 1]^2, \end{aligned} \tag{3}$$

where $\kappa(u; \alpha, \beta)$ is defined as the density function of the beta distribution function with parameters α and β at point u.

According to Nagler [24], Proposition 3.2, if $n \to \infty$, for all $(u, v) \in (0, 1)$, then $Bias(\widehat{c}(u, v)) \to 0$ and $Var(\widehat{c}(u, v)) \to 0$. That is, $\widehat{c}(u, v) \xrightarrow{p} c(u, v)$.

Based on this, the beta kernel has two leading advantages. One is that it matches the compact support of the object to be estimated. The other is that it has flexibility in form and changes the smoothness naturally as we move away from the boundaries. As a result, beta kernel estimators are naturally free of boundary bias and they produce estimates with a smaller variance (Charpentier et al. [7]).

Bouezmarni and Rolin [5] proved that even under the condition that the density is unbounded at the boundaries, the beta kernel density estimator is consistent. This property may also hold in the copula density estimation problem. For instance, the bivariate Gaussian copula density is unbounded at the corners (0,0) and (1,1). Thus, the beta kernels are suitable candidates to construct well-behaved nonparametric estimators for a copula density.

Furthermore, the performances of the beta kernel estimators are very similar to those of the reflection estimator. They do not show boundary effects only when the density satisfies some conditions. Therefore, the beta kernel estimator is free of the boundary bias problem (Zhang and Karunamuni [38]).

Finally, we can obtain the estimators for $GDDC_{g,C}$ and $GDDC_{g,\Pi}$ as

$$\begin{aligned} \widehat{GDDC}_{g,C}(X_1, X_2) &= \int_0^1 \int_0^1 g\left(\widehat{c}(u,v)\right) \widehat{c}(u,v) \mathrm{d}u\mathrm{d}v, \\ \widehat{GDDC}_{g,\Pi}(X_1, X_2) &= \int_0^1 \int_0^1 g\left(\widehat{c}(u,v)\right) \mathrm{d}u\mathrm{d}v, \end{aligned} \tag{4}$$

where $\widehat{c}(u,v)$ was presented in (3).

6. The Proposed ICA Algorithm Based on $\widehat{GDDC}_{g,C}$ and $\widehat{GDDC}_{g,\Pi}$

In this section, we delve into the fundamental processes of several novel ICA algorithms. These algorithms denoted as $\widehat{GDDC}_{g,C}$ and $\widehat{GDDC}_{g,\Pi}$ leverage distinct estimators, namely GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ for $GDDC_{g,C}$ and $GDDC_{g,\Pi}$.

Let us begin by considering a d-dimensions random vector, denoted as $\boldsymbol{X}$, and a matrix W, representing the transformation matrix. The objective of ICA algorithms is to determine the optimal W matrix, such that the components $Y_1, Y_2, \ldots, Y_d$ of a new random vector, $\boldsymbol{Y}$, can be expressed as $\boldsymbol{Y} = \boldsymbol{X}W$, while minimizing the statistical dependency. This optimization is achieved through the formulation of a contrast function, denoted as f, which quantifies the degree of dependency among the components. Consequently, the ICA problem can be defined as the minimization of this contrast function.

In the context of statistical independence within the ICA framework, pairwise independence serves as a sufficient measure (Comon [10]). Hence, solving a d-dimensional ICA problem entails successively addressing 2-dimensional ICA subproblems. In practice, a d-dimensional linear transformation can be described by a $d \times d$ orthogonal matrix, denoted as R, which can be decomposed into a series of 2-dimensional rotations. Utilizing this insight, the transformation matrix can be formulated as $R = \prod_{1 \leqslant i \leqslant j \leqslant d} R_{ij}(\theta) = [r_{ij}(\theta)]_{d \times d}$, where for $i \neq j$, $r_{ii}(\theta) = r_{jj}(\theta) = \cos(\theta)$ and $r_{ij}(\theta) = -r_{ji}(\theta) = \sin(\theta)$, and for $k \neq i, j$, $r_{kk} = 1$, with the other entries 0.

According to this equation, each matrix $R_{ij}(\theta)$ defines a rotation matrix for a specific pair of dimensions i and j. The concept of searching for the optimal rotation angle shares similarities with the RADICA Algorithms proposed by Learned-Miller and Iii [18], wherein the corresponding demixed dataset aims to minimize its Generalized Independent Component Analysis (GICA$_g$).

The general process of the algorithm for the 2-dimensional case (i.e., $d = 2$) is summarized as follows:

Algorithm 6.1.

Input: A matrix $X \in \mathbb{R}^{n \times 2}$ where the rows are mixed centered components.

Procedure for calculating GDDCICA$_{g,C}$ **and** GDDCICA$_{g,\Pi}$ **:**

(1) Make the matrix X white, so that $Y = X \times Z'$, where Z is a whitening matrix.

(2) Define $\widehat{GDDC}_{g,C}(S(\theta))$ or $\widehat{GDDC}_{g,\Pi}(S(\theta))$ as function of θ, where $S(\theta) = Y \times R(\theta)$, such that $R(\theta)$ is an orthogonal rotation matrix as

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

(3) Minimize the function $GDDC_{g,C^*}(X_1, X_2)$ over $\theta \in [-\pi/2, \pi/2]$ and set
$\theta_0 = \underset{\theta}{\operatorname{argmin}}\ \widehat{GDDC}_{g,C}(S(\theta))$ or $\theta_0 = \underset{\theta}{\operatorname{argmin}}\ \widehat{GDDC}_{g,\Pi}(S(\theta))$.

Output: Unmixing $\widehat{W} = R^{'}(\theta_0) \times Z$, and matrix of source signal estimates $\hat{S} = Y \times R(\theta_0)$.

7. Simulation Study

To compare the performance between GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ rather to FastICA, Infomax, JADE, RADICAL, HICA and RLICA, the Monte-Carlo simulation were conducted in `R` software package. The study employed the set of 18 distinct one-dimensional densities as provided by Bach and Jordan [4] include student-t, uniform, exponential, and a mixture of two Laplace densities, as well as symmetric and nonsymmetric Gaussian mixtures. 1 illustrates the density plots of the various distributions mentioned above. All ICA algorithms share the objective of extracting an unmixing matrix denoted as W, which is applied to the matrix X to recover estimations of independent components. The Amari error, introduced by Amari et al. [2], was employed to compare the performances of these algorithms. In this context, the mixing matrix is denoted by A. Therefore, the true unmixing matrix, with the estimator, is defined as $W = A^{-1}$. The calculation of the Amari error is derived through

$$\text{Amari error} = \frac{1}{2d(d-1)} \sum_{i,j=1}^{d} \left(\frac{|a_{ij}|}{\max_i |a_{ij}|} + \frac{|a_{ij}|}{\max_j |a_{ij}|} \right) - \frac{1}{d-1},$$

where $a_{ij} = [\widehat{W}A]_{ij}$ denoted as an estimated unmixing matrix $\widehat{W}$. It is notable that this metric always takes values between 0 and $d-1$. It would be equal to zero if the estimated and true unmixing matrices, denoted as $\widehat{W}$ and W respectively, indicate similar components. Additionally, it is invariant to the permutation and scaling of the columns of A and $\widehat{W}$.

For the simulation, 2-dimensional data with sizes $n = 1000$ and $n = 1500$ were utilized along with the mixing matrix $A = \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix}$, where $U_i, i = 1, 2, 3, 4$ are independently follow a uniform distribution over $(0, 2)$. In the process the average of the Amari errors of the methods FastICA, Infomax, JADE, RADICAL, HICA, RLICA, GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ by 120

replications of their corresponding algorithms were obtained. In GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ the following functions as some special cases of the function g in (4) were considered:

$$g_1(x) = \frac{(x^{c_1}-1)^2}{x^{2c_1}+1}, \qquad g_2(x) = e^{c_2(x^{c_2^*}-1)} - c_2(x^{c_2^*}-1) - 1,$$
$$g_3(x) = -\ln(x^{c_3}) + x^{c_3} - 1, \; g_4(x) = x^{c_4}\ln(x^{c_4}) - x^{c_4} + 1,$$

where c_1, c_2, c_2^*, c_3 and c_4 have represented some real values.

So, the $\varphi(x)$ denotes the generator of an Archimedean copula. Definition 4.1.1 of Nelsen [25] implies that it is strictly decreasing over $[0,\infty)$ and $\varphi(1)=0$. Therefore, the generators of some of the well-known Archimedean copulas, including Gumbel-Hougaard denoted by g_{GH}, Gumbel-Barnett denoted by g_{GB}, Clayton denoted by g_{C}, and Joe denoted by g_{J} is occured.

$$g_{\mathrm{GH}_i}(x) = \left((-\ln(x))^{c_5}\right)^2, c_5 \in [1,\infty),$$
$$g_{\mathrm{GB}_i}(x) = \left(\ln\left(1 - c_6\ln(x)\right)\right)^2, c_6 \in (0,1],$$
$$g_{\mathrm{C}_i}(x) = \left(\frac{x^{-c_7}-1}{c_7}\right)^2, c_7 \in [-1,\infty),$$
$$g_{\mathrm{J}}(x) = \left(-\ln\left(1-(1-x)^{c_8}\right)\right)^2, c_8 \in [1,\infty),$$

where $i = 1,2,3$.

The results investigate that GDDCICA$_{g_1,C}$ with $c_1 = 2$, GDDCICA$_{g_2,C}$ with $(c_2, c_2^*) = (-0.5, 1)$, GDDCICA$_{g_3,\Pi}$ with $c_3 = -0.5$, and GDDCICA$_{g_4,\Pi}$ with $c_4 = -0.5$ represent good performances. Also GDDCICA$_{g_{\mathrm{GH}_1},C}$ and GDDCICA$_{g_{\mathrm{GH}_1},\Pi}$, with $c_5 = 1$, GDDCICA$_{g_{\mathrm{GH}_2},C}$ and GDDCICA$_{g_{\mathrm{GH}_2},\Pi}$, with $c_5 = 2$, GDDCICA$_{g_{\mathrm{GH}_3},C}$ and GDDCICA$_{g_{\mathrm{GH}_3},\Pi}$, with $c_5 = 3$, GDDCICA$_{g_{\mathrm{GB}_1},C}$ and GDDCICA$_{g_{\mathrm{GB}_1},\Pi}$, with $c_6 = 0.5$, GDDCICA$_{g_{\mathrm{GB}_2},C}$ and GDDCICA$_{g_{\mathrm{GB}_2},\Pi}$, with $c_6 = 0.7$, GDDCICA$_{g_{\mathrm{GB}_3},C}$ and GDDCICA$_{g_{\mathrm{GB}_3},\Pi}$, with $c_6 = 0.9$, GDDCICA$_{g_{\mathrm{C}_1},C}$ and GDDCICA$_{g_{\mathrm{C}_1},\Pi}$, with $c_7 = -0.5$, GDDCICA$_{g_{\mathrm{C}_2},C}$ and GDDCICA$_{g_{\mathrm{C}_2},\Pi}$, with $c_7 = -1.5$, GDDCICA$_{g_{\mathrm{C}_3},C}$ and GDDCICA$_{g_{\mathrm{C}_3},\Pi}$, with $c_7 = 0.5$, and GDDCICA$_{g_{\mathrm{J}},C}$ and GDDCICA$_{g_{\mathrm{J}},\Pi}$, with $c_8 = 1$ represents fruitful outputs.

The Amari errors averages for $n = 1000$ and $n = 1500$ samples are reported in Tables 1 and 2 respectively. As a results, for $n = 1000$, each various types of GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ outperform their competitors in 13 out of cases. Also for $n = 1500$ various types of GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ in 11 out of 18, the results were better. For $n = 1000$, the JADE and RADICAL methods outperform in 2 out of 18 cases. Also for $n = 1500$, the JADE method outperforms in 3 cases, the RADICAL method outperforms in 2 out of 18 cases, the results have shown better performances than others.

The average of all Amari errors on all distributions (a) to (r) reported at last rows in Tables 1 and 2. It could be concluded that for $n = 1000$ the GDDCICA$_{g_1,C}$, GDDCICA$_{g_2,C}$, GDDCICA$_{g_3,\Pi}$, GDDCICA$_{g_4,\Pi}$, GDDCICA$_{g_{\mathrm{C}_3},\Pi}$,

$\text{GDDCICA}_{g_{\text{GH}_1},\Pi}$, $\text{GDDCICA}_{g_{\text{GH}_2},\Pi}$, and $\text{GDDCICA}_{g_{\text{GH}_3},\Pi}$ are the best techniques in terms of the average of the Amari errors in all distributions. Also, we see that for $n = 1500$ the $\text{GDDCICA}_{g_{\text{C}_2},C}$ is the best technique in terms of the average of the Amari errors in all distributions.

Indeed, two random selected distributions average of Amari errors calculations which reported in the last line of Tables 1 and 2 showed that for $n = 1000$ and $n = 1500$, the Infomax and JADE methods perform well.

Also to more insights, performances evaluation for all methods in different classes of distributions was done by average of Amari errors in each class. The classes on unimodal distributions (a, b, d, e, i, l, o, r), multimodal distributions (f, g, j, m, p), and transitional distributions (c, h, k, n, q) have considered as an elementary perspective. But in the progressive perspective, the class of symmetric distributions (a, b, c, d, f, g, h, i, m, n, o), and nonsymmetric distributions (e, j, k, l, p, q, r) have considered in the second view and also the class of positive kurtosis distributions (a, b, d, e, f), and negatively kurtosis distributions (c, g, h, i, j, k, l, m, n, o, p, q, r) were considered in the third view. In each step, the classes are a partition for the set including all distributions. The results are summarized in Table 3.

It is illustrated that, for $n = 1000$ in the symmetric, multimodal, and negatively kurtosis classes, the $\text{GDDCICA}_{g_{\text{C}_2},C}$ is the best and the $\text{GDDCICA}_{g_{\text{GH}_3},\Pi}$, in the symmetric, unimodal, and negatively kurtosis classes, has the best performances. For $n = 1500$ in all classes except for the transitional class, the $\text{GDDCICA}_{g_1,C}$ and $\text{GDDCICA}_{g_{\text{GH}_3},\Pi}$ are the best. Also, the $\text{GDDCICA}_{g_{\text{C}_3},\Pi}$ is the best in all classes except for the positive kurtosis classes. Generally, $\text{GDDCICA}_{g,C}$ and $\text{GDDCICA}_{g,\Pi}$, in terms of the average of the Amari errors, have the best performances in symmetric and negative kurtosis classes.

8. **Illustrative Data Analyses**

Due to the identification of homogeneous structures within the data and the objective of maximizing within-group similarities while maximizing between-group differences, clustering, particularly time series clustering, emerges as an effective and appealing method for generating valuable insights across various fields. Furthermore, one of the primary aims in employing all ICA algorithms, despite their inherent complexity and unknown details, is the extraction of independent components from a set of mixed components. Consequently, the algorithm $\text{GDDCICA}_{g,C}$ and algorithm $\text{GDDCICA}_{g,\Pi}$ can be utilized as a pre-processing stage for time series data in clustering applications, facilitating the identification of the block dependencies between time series. Indeed, the utilization of ICA not only enables more suitable clustering of time series data but also provides valuable insights into common features shared among the time series. This pre-processing technique creates new components from sources, which contain sufficient information about the trend of the time series. So, the

TABLE 1. Averages of Amari errors for $d = 2$ and $n = 1000$ samples with 120 replications for each distribution (a) to (r). The best (smallest) entry of our functions in each row is boldfaced.

distribution	FastICA	Infomax	JADE	RADICAL	HICA	RLICA	g_1, C	g_2, C	g_3, Π	g_4, Π
a	0.5022	0.5022	0.4971	0.5069	0.4936	0.4962	0.4959	0.4999	**0.4913**	0.4963
b	0.5125	0.5125	0.5020	0.5163	0.4993	0.5072	**0.4980**	0.5131	0.5013	0.5102
c	0.5291	0.5291	0.5299	0.5301	0.5308	0.5352	0.5337	0.5326	**0.5258**	0.5303
d	0.5231	0.5232	0.5047	0.5047	0.5154	0.5009	**0.4947**	0.5093	**0.4999**	**0.4998**
e	0.5385	0.5366	0.5374	0.5383	0.5451	0.5441	0.5433	**0.5274**	**0.5360**	0.5427
f	0.5346	0.5345	0.5384	0.5379	0.5398	0.5351	**0.5306**	0.5371	0.5372	0.5355
g	0.5326	0.5326	0.5341	0.5326	0.5332	0.5323	0.5345	0.5327	**0.5312**	**0.5312**
h	0.5259	0.5259	0.5316	0.5251	0.5254	0.5379	0.5257	0.5396	0.5357	0.5295
i	0.5311	0.5315	0.5305	0.5185	0.5375	0.5220	0.5207	**0.5159**	0.5216	**0.5180**
j	0.5383	0.5232	0.5433	0.5324	0.5371	0.5311	0.5356	0.5365	0.5300	0.5266
k	0.5241	0.5241	0.5182	0.5186	0.5252	0.5218	0.5184	0.5281	0.5247	0.5247
l	0.5186	0.5136	0.5183	0.5162	0.5162	0.5142	**0.5114**	**0.5113**	0.5195	0.5139
m	0.5215	0.5214	0.5203	0.5140	0.5007	0.5060	**0.5006**	0.5039	0.5142	0.5131
n	0.4779	0.4789	0.4656	0.4654	0.5303	0.4705	0.4692	0.4783	0.4675	0.4693
o	0.5020	0.5020	0.4962	0.5048	0.5081	0.5062	0.4971	0.5007	0.5030	0.5044
p	0.5101	0.5118	0.5124	0.5092	0.5093	0.5137	0.5155	0.5130	0.5118	0.5135
q	0.5022	0.5022	0.4971	0.5069	0.4936	0.4962	0.4959	0.4999	**0.4913**	0.4963
r	0.5237	0.5236	0.5231	0.5329	0.5083	0.5128	**0.5000**	**0.5020**	0.5216	0.5170
mean	0.5193	0.5183	0.5167	0.5173	0.5194	0.5157	**0.5123**	**0.5156**	**0.5146**	**0.5151**
rand	0.4814	**0.4768**	0.4812	0.4935	0.4883	0.4828	0.4866	0.4933	0.4900	0.4917

distribution	g_{C_1}, C	g_{C_2}, C	g_{C_3}, C	g_{C_1}, Π	g_{C_2}, Π	g_{C_3}, Π	g_{GH_1}, C	g_{GH_2}, C	g_{GH_3}, C	g_{GH_1}, Π
a	0.5006	0.5045	0.4955	0.5017	0.5002	**0.4861**	0.4977	0.4970	0.4991	0.4968
b	0.5109	0.5058	0.5095	0.5092	0.5102	0.5057	0.5135	0.5050	0.5091	0.5053
c	0.5379	0.5461	0.5393	0.5332	0.5360	0.5332	0.5382	0.5313	**0.5286**	0.5306
d	0.5109	0.5111	0.5099	0.5016	0.5066	0.5018	0.5094	**0.4926**	0.5020	0.5045
e	0.5391	**0.5310**	0.5459	0.5431	0.5406	0.5471	0.5380	**0.5318**	0.5442	**0.5360**
f	0.5366	**0.5342**	0.5382	0.5352	0.5355	**0.5327**	0.5364	0.5392	0.5353	**0.5334**
g	**0.5264**	**0.5313**	**0.5284**	**0.5308**	**0.5278**	**0.5286**	0.5335	0.5381	**0.5305**	0.5330
h	0.5310	0.5288	0.5356	0.5429	0.5315	0.5268	0.5336	0.5324	0.5316	0.5320
i	0.5250	0.5210	0.5332	0.5292	0.5206	0.5301	**0.5078**	**0.5260**	**0.5170**	**0.5160**
j	0.5297	0.5344	0.5413	0.5294	0.5327	0.5285	0.5348	0.5299	0.5397	0.5298
k	0.5236	0.5264	0.5238	0.5200	0.5216	0.5217	0.5239	0.5336	0.5242	0.5235
l	**0.5112**	**0.5089**	0.5164	0.5170	0.5158	0.5182	0.5174	0.5236	0.5177	0.5152
m	**0.4931**	0.5153	0.5049	0.5107	0.5053	0.5008	0.5105	0.5013	**0.4987**	0.5150
n	**0.4931**	0.5153	0.5049	0.5107	0.5053	0.5008	0.5105	0.5013	**0.4987**	0.5150
o	0.5041	0.5110	0.4970	0.5054	0.5077	0.5010	0.4965	0.5216	0.5077	0.5050
p	0.5159	0.5161	0.5137	0.5195	0.5150	0.5129	0.5145	0.5114	0.5168	0.5127
q	0.5006	0.5045	0.4955	0.5017	0.5002	**0.4861**	0.4977	0.5139	0.5043	0.4968
r	0.5006	0.5045	0.4955	0.5017	0.5002	**0.4861**	0.4977	0.5139	0.5043	0.4968
mean	0.5158	0.5172	0.5175	0.5173	0.5162	**0.5132**	0.5163	0.5167	0.5169	**0.5153**
rand	0.4871	0.4789	0.4923	0.4900	0.4872	0.4895	0.5004	0.4955	0.5037	0.4943

distribution	g_{GH_2}, Π	g_{GH_3}, Π	g_{GB_1}, C	g_{GB_2}, C	g_{GB_3}, C	g_{GB_1}, Π	g_{GB_2}, Π	g_{GB_3}, Π	g_J, C	g_J, Π
a	**0.4934**	**0.4912**	0.5081	0.5088	0.4957	**0.4872**	0.4936	0.4954	0.4982	**0.4888**
b	0.5079	0.5057	0.5116	0.5111	0.5132	0.5095	0.5088	0.5131	0.5160	0.5104
c	0.5298	0.5299	0.5361	0.5382	0.5304	0.5384	0.5312	0.5501	0.5306	0.5370
d	**0.4933**	**0.4962**	**0.5005**	0.5050	0.5064	0.5076	0.5040	0.5024	**0.5002**	0.5059
e	0.5450	0.5372	**0.5328**	0.5641	0.5583	0.5429	0.5640	0.5574	0.5397	0.5470
f	**0.5315**	0.5361	**0.5330**	**0.5341**	**0.5328**	0.5371	**0.5338**	**0.5308**	**0.5331**	0.5347
g	0.5368	0.5391	**0.5276**	0.5337	**0.5305**	**0.5312**	0.5325	0.5327	0.5339	**0.5303**
h	0.5286	0.5269	0.5389	0.5317	0.5341	0.5261	0.5256	0.5322	0.5345	0.5315
i	**0.5127**	**0.5168**	0.5198	0.5244	0.5200	0.5274	0.5203	**0.5184**	0.5233	**0.5179**
j	0.5265	0.5280	0.5334	0.5324	0.5486	0.5302	0.5305	0.5474	0.5261	0.5291
k	0.5196	0.5236	0.5275	0.5267	0.5271	0.5275	0.5238	0.5263	0.5256	0.5239
l	0.5209	0.5161	**0.5127**	**0.5125**	**0.5100**	0.5163	0.5187	0.5181	0.5196	0.5140
m	0.5087	**0.4906**	**0.4983**	0.5094	0.5083	0.5147	0.5088	0.5051	0.5067	0.5066
n	0.4720	0.4655	0.4722	0.4657	0.4701	0.4720	0.4745	0.4730	0.4768	0.4790
o	0.5093	0.5124	0.5128	0.5100	0.5063	0.5069	0.5044	0.5092	0.5159	0.4986
p	0.5146	0.5127	0.5154	0.5174	0.5155	0.5149	0.5149	0.5134	0.5098	0.5133
q	**0.4934**	0.5063	0.5081	0.5088	0.4957	**0.4872**	**0.4936**	0.4954	0.4982	**0.4888**
r	0.5169	**0.5039**	**0.5056**	**0.5057**	0.5111	0.5163	0.5129	0.5159	**0.5077**	0.5213
mean	**0.5145**	**0.5132**	0.5164	0.5189	0.5175	0.5163	0.5164	0.5187	0.5164	0.5155
rand	0.4983	0.5079	0.4859	0.4896	0.4849	0.4884	0.4909	0.4900	0.4798	0.4897

TABLE 2. Averages of Amari errors for $d = 2$ and $n = 1500$ samples with 120 replications for each distribution (a) to (r). The best (smallest) entry of our functions in each row is boldfaced.

distribution	FastICA	Infomax	JADE	RADICAL	HICA	RLICA	g_1, C	g_2, C	g_3, Π	g_4, Π
a	0.5251	0.5250	0.5220	0.5311	0.5352	0.5240	0.5272	**0.5219**	0.5289	**0.5191**
b	0.4913	0.4915	0.4997	0.5015	0.4988	0.4993	0.5024	0.4998	0.5051	0.5017
c	0.5180	0.5182	0.5182	0.5197	0.5147	0.5193	0.5197	**0.5097**	0.5190	**0.5123**
d	0.5078	0.5079	0.5092	0.5149	0.5118	0.5102	0.5179	0.5090	0.5147	0.5149
e	0.4988	0.4988	0.4998	0.5040	0.5091	0.5103	0.5096	0.5090	0.5033	**0.4944**
f	0.4737	0.4736	0.4727	0.4780	0.4732	0.4779	0.4814	0.4737	0.4855	0.4878
g	0.5351	0.5352	0.5355	0.5326	0.5402	0.5413	**0.5303**	0.5350	**0.5324**	0.5370
h	0.5057	0.5059	0.5154	0.5026	0.5071	0.5087	0.5067	0.5109	0.5079	0.5100
i	0.5316	0.5350	0.5314	0.5264	0.5202	0.5203	0.5232	0.5280	0.5237	0.5285
j	0.5005	0.4994	0.5051	0.5244	0.5041	0.5062	0.5149	0.5026	0.5034	0.5008
k	0.4749	0.4765	0.4704	0.4731	0.4814	0.4799	0.4805	0.4824	0.4807	0.4814
l	0.5133	0.5109	0.5108	0.5261	0.5146	0.5105	**0.5016**	**0.5029**	**0.5083**	0.5131
m	0.5181	0.5182	0.5249	0.5245	0.5135	0.5167	0.5212	0.5157	0.5144	0.5248
n	0.5117	0.5120	0.5112	0.5213	0.5227	0.5142	0.5207	0.5345	0.5211	0.5127
o	0.5217	0.5205	0.5195	0.5122	0.5143	0.5218	0.5203	0.5187	0.5207	0.5198
p	0.5276	0.5232	0.5284	0.5216	0.5108	0.5167	0.5254	0.5329	0.5206	0.5212
q	0.5048	0.5048	0.5056	0.5049	0.5085	0.5048	0.5072	0.5077	0.5058	0.5054
r	0.5398	0.5391	0.5810	0.5459	0.5415	0.5383	**0.5351**	0.5406	0.5427	0.5432
mean	0.5111	0.5109	0.5145	0.5147	0.5123	0.5122	0.5136	0.5131	0.5132	0.5127
rand	0.5103	0.5048	**0.4983**	0.5080	0.5065	0.5071	0.5100	0.5082	0.5042	0.5095

distribution	g_{C_1}, C	g_{C_2}, C	g_{C_3}, C	g_{C_1}, Π	g_{C_2}, Π	g_{C_3}, Π	g_{GH_1}, C	g_{GH_2}, C	g_{GH_3}, C	g_{GH_1}, Π
a	0.5213	0.5201	0.5290	0.5250	0.5274	0.5271	0.5274	**0.5173**	**0.5052**	0.5256
b	0.4958	0.4984	0.5031	0.5077	**0.4907**	0.5057	0.4945	0.5068	0.5033	0.5026
c	0.5242	0.5195	0.5251	0.5288	0.5269	0.5249	0.5192	0.5008	0.5079	0.5285
d	0.5109	0.5097	0.5223	0.5151	0.5109	0.5155	0.5113	0.5587	0.5506	0.5139
e	0.5220	0.5205	0.5021	0.5139	0.5211	**0.4982**	0.5162	0.5245	0.5022	**0.4980**
f	0.4744	0.4746	0.4783	0.4799	0.4751	0.4822	0.4843	0.5117	0.5102	0.4772
g	**0.5304**	**0.5295**	**0.5251**	0.5323	0.5348	**0.5310**	0.5380	0.4904	0.5028	0.5364
h	0.5121	0.5116	0.5074	0.5062	0.5081	0.5092	0.5108	0.5222	0.5167	0.5069
i	**0.5166**	**0.5201**	0.5288	0.5209	0.5244	0.5253	0.5297	0.4990	0.5078	**0.5201**
j	0.5007	0.5004	0.5065	0.5015	0.5018	0.5028	0.5054	0.5167	0.5124	0.5020
k	0.4813	0.4725	0.4842	0.4828	0.4813	0.4822	0.4829	0.5034	0.5130	0.4791
l	0.5188	0.5236	**0.5103**	0.5149	0.5135	0.5150	0.5181	0.5162	0.5052	0.5131
m	0.5209	**0.5107**	0.5185	0.5204	0.5179	0.5167	0.5218	0.5226	0.5368	0.5190
n	0.5187	0.5138	0.5249	0.5298	0.5213	0.5190	0.5117	0.5153	0.5228	0.5126
o	0.5199	0.5129	0.5266	0.5183	0.5220	0.5209	0.5167	0.5160	0.5157	0.5207
p	0.5303	0.5119	0.5280	0.5240	0.5264	0.5261	0.5264	0.5303	0.5437	0.5246
q	0.5079	**0.5047**	0.5056	0.5052	0.5070	**0.5035**	0.5072	0.5131	0.5091	**0.5045**
r	**0.5349**	**0.5343**	0.5435	0.5380	0.5378	0.5442	**0.5361**	0.5463	0.5209	0.5401
mean	0.5134	**0.5105**	0.5150	0.5147	0.5138	0.5139	0.5143	0.5173	0.5159	0.5125
rand	0.5190	0.5137	0.5176	0.5185	0.5168	0.5180	0.5152	0.5091	0.5026	0.5040

distribution	g_{GH_2}, Π	g_{GH_3}, Π	g_{GB_1}, C	g_{GB_2}, C	g_{GB_3}, C	g_{GB_1}, Π	g_{GB_2}, Π	g_{GB_3}, Π	g_J, C	g_J, Π
a	**0.5142**	**0.5125**	0.5301	0.5291	0.5328	0.5356	0.5343	0.5317	0.5247	0.5230
b	0.5099	0.4990	0.5023	0.4966	**0.4911**	0.5077	0.5060	0.4983	0.5058	0.5088
c	0.5220	0.5079	0.5182	0.5214	0.5291	0.5245	0.5185	0.5281	0.5302	0.5311
d	0.5038	0.5115	0.5186	0.5096	**0.5053**	**0.5072**	0.5205	0.5128	0.5085	0.5094
e	0.4989	0.5099	0.5081	0.5515	0.5374	0.5092	0.5453	0.5428	0.5028	0.5109
f	0.4865	0.5132	0.4799	0.4752	0.4745	0.4774	0.4831	0.4759	0.4763	0.4850
g	0.5314	0.4853	**0.5300**	0.5384	0.5396	**0.5307**	0.5397	0.5403	0.5370	0.5334
h	0.5107	0.5106	0.5034	0.5059	0.5087	0.5082	0.5092	0.5065	0.5041	0.5043
i	0.5236	0.4992	0.5232	0.5233	0.5435	0.5235	**0.5186**	0.5471	0.5235	0.5272
j	0.5032	0.5212	0.5020	0.5018	0.5030	0.5031	0.5059	0.5063	0.5009	0.5061
k	0.4846	0.5048	0.4729	0.4804	0.4833	0.4822	0.4803	0.4800	0.4815	0.4792
l	0.5092	0.5020	0.5157	0.5093	0.5164	0.5136	0.5159	0.5133	0.5135	0.5111
m	0.5165	0.5288	0.5167	0.5195	0.5156	0.5182	0.5200	0.5171	0.5164	0.5283
n	0.5146	0.5162	0.5227	0.5280	0.5371	0.5208	0.5169	0.5172	0.5170	0.5209
o	0.5690	0.5180	0.5159	0.5160	0.5222	0.5215	0.5205	0.5211	0.5173	0.5222
p	0.5331	0.5201	0.5291	0.5281	0.5318	0.5346	0.5333	0.5307	0.5237	0.5220
q	0.5070	0.4952	**0.5035**	0.5058	0.5076	0.5056	0.5070	0.5058	0.5051	**0.5038**
r	0.5537	0.5365	0.5383	0.5400	0.5429	0.5440	0.5401	**0.5351**	**0.5368**	**0.5343**
mean	0.5162	0.5107	0.5128	0.5156	0.5179	0.5149	0.5175	0.5172	0.5125	0.5145
rand	0.5134	0.5149	0.5167	0.5188	0.5138	0.5189	0.5145	0.5174	0.5175	0.5176

TABLE 3. Averages of Amari errors for $d = 2$, $n = 1000$, and $n = 1500$ samples in each class of distributions. The best (smallest) entry of our functions in each row is boldfaced.

n=1000										
class	FastICA	Infomax	JADE	RADICAL	HICA	RLICA	g_1, C	g_2, C	g_3, Π	g_4, Π
symmetric	0.5175	0.5176	0.5137	0.5142	0.5195	0.5136	**0.5092**	0.5148	**0.5117**	**0.5125**
nonsymmetric	0.5222	0.5193	0.5214	0.5221	0.5193	0.5191	**0.5172**	**0.5169**	0.5193	0.5192
kurtoses+	0.5211	0.5207	0.5142	0.5193	0.5168	0.5167	**0.5131**	0.5165	**0.5109**	0.5159
kurtoses-	0.5187	0.5173	0.5176	0.5165	0.5204	0.5154	**0.5119**	**0.5153**	0.5161	**0.5148**
unimodal	0.5190	0.5182	0.5137	0.5173	0.5154	0.5130	**0.5076**	**0.5100**	**0.5118**	**0.5128**
multimodal	0.5274	0.5247	0.5297	0.5252	0.5240	0.5236	**0.5234**	0.5246	0.5249	0.5240
transitional	0.5118	0.5120	0.5085	0.5092	0.5211	0.5123	0.5086	0.5157	0.5090	0.5100
class	g_{C_1}, C	g_{C_2}, C	g_{C_3}, C	g_{C_1}, Π	g_{C_2}, Π	g_{C_3}, Π	g_{GH_1}, C	g_{GH_2}, C	g_{GH_3}, C	g_{GH_1}, Π
symmetric	0.5142	0.5164	0.5146	0.5150	0.5138	**0.5101**	**0.5134**	**0.5136**	**0.5124**	**0.5126**
nonsymmetric	**0.5185**	**0.5184**	0.5221	0.5208	0.5199	**0.5182**	0.5208	0.5217	0.5240	0.5194
kurtoses+	0.5199	0.5197	0.5200	0.5178	0.5187	0.5148	0.5194	**0.5115**	0.5166	0.5146
Kurtoses-	**0.5143**	0.5162	0.5165	0.5171	**0.5152**	**0.5126**	**0.5151**	0.5187	0.5170	**0.5155**
unimodal	0.5139	**0.5126**	0.5157	0.5153	0.5144	**0.5129**	**0.5125**	0.5131	0.5147	**0.5126**
multimodal	**0.5203**	0.5263	0.5253	0.5251	**0.5233**	**0.5207**	0.5259	0.5240	0.5242	0.5248
transitional	0.5145	0.5155	0.5126	0.5127	0.5119	**0.5063**	0.5128	0.5152	0.5130	0.5101
class	g_{GH_2}, Π	g_{GH_3}, Π	g_{GB_1}, C	g_{GB_2}, C	g_{GB_3}, C	g_{GB_1}, Π	g_{GB_2}, Π	g_{GB_3}, Π	g_J, C	g_J, Π
symmetric	**0.5113**	**0.5100**	0.5144	0.5156	**0.5134**	0.5144	**0.5125**	0.5148	0.5154	**0.5128**
nonsymmetric	0.5196	**0.5183**	0.5194	0.5239	0.5238	0.5193	0.5226	0.5248	**0.5181**	0.5196
kurtoses+	**0.5139**	**0.5120**	0.5178	0.5254	0.5208	0.5171	0.5203	0.5237	0.5169	0.5178
Kurtoses-	**0.5147**	**0.5137**	0.5158	0.5163	0.5162	0.5160	**0.5149**	0.5168	0.5162	**0.5145**
unimodal	**0.5124**	**0.5099**	**0.5130**	0.5177	0.5151	0.5143	0.5158	0.5162	0.5151	**0.5130**
multimodal	**0.5236**	**0.5213**	**0.5215**	0.5254	0.5271	0.5256	0.5241	0.5259	0.5219	**0.5228**
transitional	0.5087	0.5104	0.5166	0.5142	0.5115	0.5102	0.5097	0.5154	0.5131	0.5120

n=1500										
class	FastICA	Infomax	JADE	RADICAL	HICA	RLICA	g_1, C	g_2, C	g_3, Π	g_4, Π
symmetric	0.5127	0.5130	0.5145	0.5150	0.5138	0.5140	0.5155	0.5143	0.5158	0.5153
nonsymmetric	0.5085	**0.5075**	0.5144	0.5143	0.5100	0.5095	0.5106	0.5112	0.5093	0.5085
kurtoses+	**0.4993**	0.4994	0.5007	0.5059	0.5056	0.5043	0.5077	0.5027	0.5075	0.5036
Kurtoses-	0.5156	0.5153	0.5198	0.5181	0.5149	0.5153	0.5159	0.5170	0.5154	0.5162
unimodal	0.5162	0.5161	0.5217	0.5203	0.5182	0.5168	0.5172	0.5162	0.5184	0.5168
multimodal	0.5110	0.5099	0.5133	0.5162	0.5084	0.5118	0.5146	0.5120	0.5113	0.5143
transitional	**0.5030**	0.5035	0.5042	0.5043	0.5069	0.5054	0.5070	0.5090	0.5069	0.5044
class	g_{C_1}, C	g_{C_2}, C	g_{C_3}, C	g_{C_1}, Π	g_{C_2}, Π	g_{C_3}, Π	g_{GH_1}, C	g_{GH_2}, C	g_{GH_3}, C	g_{GH_1}, Π
symmetric	0.5132	**0.5110**	0.5172	0.5168	0.5145	0.5161	0.5150	0.5146	0.5163	0.5149
nonsymmetric	0.5137	0.5097	0.5115	0.5115	0.5127	0.5103	0.5132	0.5215	0.5152	0.5088
kurtoses+	0.5049	0.5047	0.5070	0.5083	0.5050	0.5057	0.5067	0.5238	0.5143	0.5035
Kurtoses-	0.5167	**0.5127**	0.5180	0.5172	0.5172	0.5170	0.5172	**0.5148**	0.5165	0.5160
unimodal	0.5175	0.5175	0.5207	0.5192	0.5185	0.5190	0.5188	0.5231	0.5139	0.5168
multimodal	0.5113	**0.5054**	0.5113	0.5116	0.5112	0.5118	0.5152	0.5143	0.5212	0.5118
transitional	0.5088	**0.5044**	0.5094	0.5106	0.5089	0.5078	0.5064	0.5110	0.5139	0.5063
class	g_{GH_2}, Π	g_{GH_3}, Π	g_{GB_1}, C	g_{GB_2}, C	g_{GB_3}, C	g_{GB_1}, Π	g_{GB_2}, Π	g_{GB_3}, Π	g_J, C	g_J, Π
symmetric	0.5184	**0.5093**	0.5146	0.5148	0.5181	0.5159	0.5170	0.5178	0.5146	0.5176
nonsymmetric	0.5128	0.5128	0.5099	0.5167	0.5175	0.5132	0.5183	0.5163	0.5092	0.5096
kurtoses+	0.5027	0.5092	0.5078	0.5124	0.5082	0.5074	0.5178	0.5123	0.5036	0.5074
Kurtoses-	0.5214	**0.5112**	**0.5147**	0.5168	0.5216	0.5177	0.5174	0.5191	0.5159	0.5172
unimodal	0.5228	**0.5111**	0.5190	0.5219	0.5240	0.5203	0.5252	0.5253	0.5166	0.5184
multimodal	0.5141	0.5137	0.5115	0.5126	0.5129	0.5128	0.5164	0.5141	0.5109	0.5150
transitional	0.5078	0.5069	0.5041	0.5083	0.5132	0.5083	0.5064	0.5075	0.5076	0.5079

mixing matrix coefficients obtained by GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ can be used as an input variable.

Let X be a time series matrix and $\hat{S}$ and $\hat{A}$ be estimates of the source signal matrix and the mixing coefficients matrix obtained by an ICA algorithm on X, respectively. Thus, time series matrix X can be predicted by $\hat{S}\hat{A}$; i.e., $X \approx \hat{S}\hat{A}$, which it means that matrix X is approximately equal to $\hat{S}\hat{A}$. We know that the columns of $\hat{S}$ including the independent time series and their

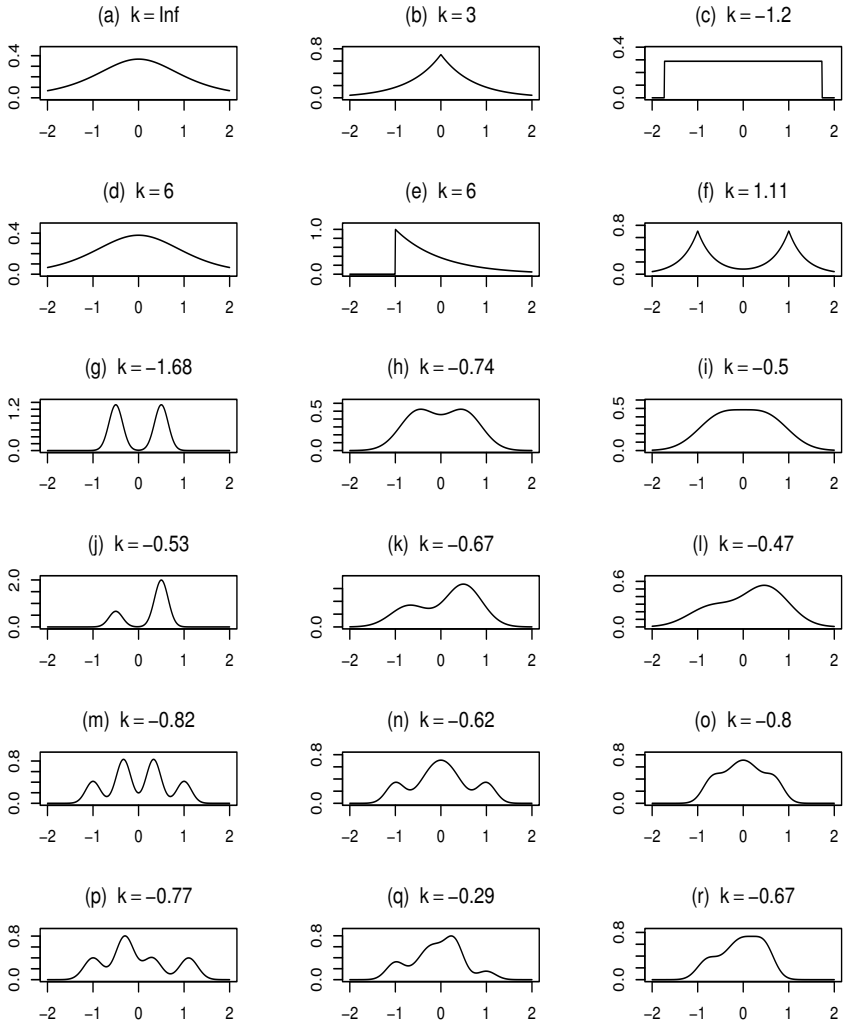

FIGURE 1. Density plots of 18 different distribution of sources: (a) Student with 3 degrees of freedom; (b) double exponential; (c) uniform; (d) Student with 5 degrees of freedom; (e) exponential; (f) mixture of two double exponentials; (g)-(h)-(i) symmetric mixtures of two Gaussians: multimodal, transitional and unimodal; (j)-(k)-(l) nonsymmetric mixtures of two Gaussians, multimodal, transitional and unimodal; (m)-(n)-(o) symmetric mixtures of four Gaussians: multimodal, transitional and unimodal; (p)-(q)-(r) nonsymmetric mixtures of four Gaussians: multimodal, transitional and unimodal.

dependencies are converted to the weight matrix $\hat{A}$. On the other hand, i-th column of X is constructed as a multiplication of matrix $\hat{S}$ and the i-th column of $\hat{A}$. Thus, the same columns in $\hat{A}$ construct the same trends in the original time series. Therefore, detecting the similar columns of $\hat{A}$, imply to detect the similar columns of X. Hence, for clustering the time series matrix X, we can apply a clustering algorithm on $\hat{A}$.

Based on Guo et al. [11], Zanghaei et al. [37] and Rahmanishamsi et al. [27], instead of applying clustering algorithms on data, it is better that a suitable transformation being used before clustering process to strengthen clustering results. In this article, the ICA algorithm was first implemented on the data and then clustering was done.

So, application of the suggested algorithm as a pre-processing in the time series clustering can induce its application. The Partitioning Around Medoids (`PAM`) algorithm was introduced by Rdusseeun and Kaufman [28] can be used as a complementary algorithm. Regard to this, the `PAM` clustering algorithm can employing to unmixing matrix extracted from resulted mixing matrix estimation using all algorithms GDDCICA$_{g,C}$ and GDDCICA$_{g,\Pi}$ to determine the suitable number of clusters and to select the best clustering in terms of the Silhouette criterion (Rousseeuw [30]) using `NbClust R` package.

Here, we consider a dataset including international tourism receipts (% of total exports) from 2000 to 2020 for 19 various countries. The dataset was selected from the organization of the World Bank (`https://data.worldbank.org`) and including the countries Argentina, Australia, Brazil, Bulgaria, Croatia, Cyprus, France, Germany, Greece, Hungary, Japan, Malaysia, Mexico, Norway, Poland, Portugal, Switzerland, Thailand, and the United States.

The goal of this analysis is clustering of these countries based on their time series over the past 21 years. In this purpose, we use the standardised data. After applying the GDDCICA$_{g,C}$, GDDCICA$_{g,\Pi}$ and some existing ICA algorithms on the data, we clustered the countries. To evaluate the accuracy of clustering, we used the common criterion, the average Silhouette score. The higher value of the average Silhouette score implies the batter clustering. The results are reported in Table 4. In this table, it is observed that the algorithm GDDCICA$_{g_2,C}$ exhibiting the highest average Silhouette score among all of algorithms, and thus it is selected as the best-performing result.

Therefore, since the GDDCICA$_{g_2,C}$ algorithm implied the highest Silhouette in clustering, the result of its clustering is presented in Table 5.

We also directly applied the PAM algorithm on the standardized data matrix. In this case, the PAM algorithm suggest 5 clusters. The result of this clustering is presented in Table 6.

Time series plots in 7 clusters obtained by GDDCICA$_{g_2,C}$ and then using the PAM, are drawn in Figure 2 and their plots in 5 clusters obtained by the direct using of the PAM, are drawn in Figure 3 . Comparing the trends plotted in all clusters in the both clusters, it is concluded that the pre-processing

TABLE 4. Number of clusters and declare the best clustering method based on the Silhouette criterion for international tourism receipts time series.

Method	Suggested number of clusters	Average Silhoutte Score	Method	Suggested number of clusters	Average Silhoutte Score
FastICA	5	0.029	$\text{GDDCICA}_{g_{\text{C}_3},\Pi}$	4	0.030
Infomax	3	0.037	$\text{GDDCICA}_{g_{\text{GH}_1},C}$	3	0.036
JADE	5	0.035	$\text{GDDCICA}_{g_{\text{GH}_2},C}$	8	0.039
RADICAL	6	0.024	$\text{GDDCICA}_{g_{\text{GH}_3},C}$	5	0.033
HICA	3	0.021	$\text{GDDCICA}_{g_{\text{J}},\Pi}$	5	0.038
RLICA	4	0.022	$\text{GDDCICA}_{g_{\text{GH}_1},\Pi}$	6	0.031
$\text{GDDCICA}_{g_1,C}$	4	0.032	$\text{GDDCICA}_{g_{\text{GH}_2},\Pi}$	4	0.029
$\text{GDDCICA}_{g_2,C}$	7	**0.041**	$\text{GDDCICA}_{g_{\text{GH}_3},\Pi}$	4	0.036
$\text{GDDCICA}_{g_3,\Pi}$	8	0.039	$\text{GDDCICA}_{g_{\text{GB}_1},C}$	5	0.029
$\text{GDDCICA}_{g_4,\Pi}$	10	0.038	$\text{GDDCICA}_{g_{\text{GB}_2},C}$	6	0.039
$\text{GDDCICA}_{g_{\text{C}_1},C}$	7	0.025	$\text{GDDCICA}_{g_{\text{GB}_3},C}$	7	0.038
$\text{GDDCICA}_{g_{\text{C}_2},C}$	6	0.028	$\text{GDDCICA}_{g_{\text{GB}_1},\Pi}$	8	0.037
$\text{GDDCICA}_{g_{\text{C}_3},C}$	8	0.035	$\text{GDDCICA}_{g_{\text{GB}_2},\Pi}$	9	0.033
$\text{GDDCICA}_{g_{\text{C}_1},\Pi}$	6	0.033	$\text{GDDCICA}_{g_{\text{GB}_3},\Pi}$	5	0.040
$\text{GDDCICA}_{g_{\text{C}_2},\Pi}$	4	0.030	$\text{GDDCICA}_{g_{\text{J}},\text{C}}$	4	0.037

TABLE 5. Clustering of standardized international tourism receipts time series using the PAM algorithm after using the $\text{GDDCICA}_{g_2,C}$ algorithm.

	Cluster members
Cluster 1	Argentina, Australia, Brazil, United States
Cluster 2	Bulgaria, Malaysia, Thailand
Cluster 3	Croatia, Cyprus, Germany, Hungary, Norway
Cluster 4	France, Mexico
Cluster 5	Japan, Greece
Cluster 6	Portugal, Poland
Cluster 7	Switzerland

technique by the proposed algorithm for clustering countries works very well, since it provided more homogeneity in the clusters. It is induced from extracted similar trends in standardized international tourism receipts in countries in the same identical clusters.

9. Conclusion

Keziou et al. [15] introduced blind source separation for independent component estimation, proposing a criterion based on mutual information derived from copulas. Building upon this work, our paper extends their concept by

TABLE 6. Clustering of standardized international tourism receipts time series by using the PAM algorithm on the data matrix directly.

	Cluster members
Cluster 1	Argentina, France, Hungary, Malaysia, United States
Cluster 2	Australia, Bulgaria, Portugal, Thailand
Cluster 3	Brazil, Germany, Japan, Mexico, Norway, Poland, Switzerlandy
Cluster 4	Cyprus, Greece
Cluster 5	Croatia

proposing a criterion based on a general class of density-based dependence criteria. Additionally, we present a copula-based version denoted as $GDDC_{g,C^*}$. Our results demonstrate that criteria from this class remain invariant under monotone transformations of random variables, and they vanish when the random variables are independent. Furthermore, we introduce two estimators, denoted as estimator $\widehat{GDDC}_{g,C}$ and estimator $\widehat{GDDC}_{g,\Pi}$, for the criterion based on copulas. Subsequently, new algorithms, denoted as algorithm $\text{GDDCICA}_{g,C}$ and algorithm $\text{GDDCICA}_{g,\Pi}$, for ICAs are developed based on these estimators.

The performance of the suggested algorithms is evaluated and compared to existing algorithms through Monte Carlo simulation studies, where the independent components follow the 18 different distributions provided by Bach and Jordan [4]. Our results indicate that in most cases, the proposed algorithms demonstrate good performance in estimating the unmixing matrix, outperforming traditional ICA algorithms across various classes of distributions in terms of the average of the Amari errors.

To demonstrate the practical application of the suggested algorithms, we employ a batch of time series data comprising international tourism receipts (% of total exports) from 2000 to 2020 for 19 different countries, using `R` software for analysis. The algorithms are applied to these samples, along with the `PAM` algorithm, to cluster countries based on their international tourism receipts time series data samples. The algorithm $\text{GDDCICA}_{g_2,C}$ is employed as a pre-processing step, followed by principal component analysis. The resulting coefficients of the mixing matrix are then used as inputs to the `PAM` clustering algorithm. Our findings from this hybrid approach illustrate that the pre-processing technique by the proposed algorithm can effectively facilitate suitable clustering of different data following diverse distributions.

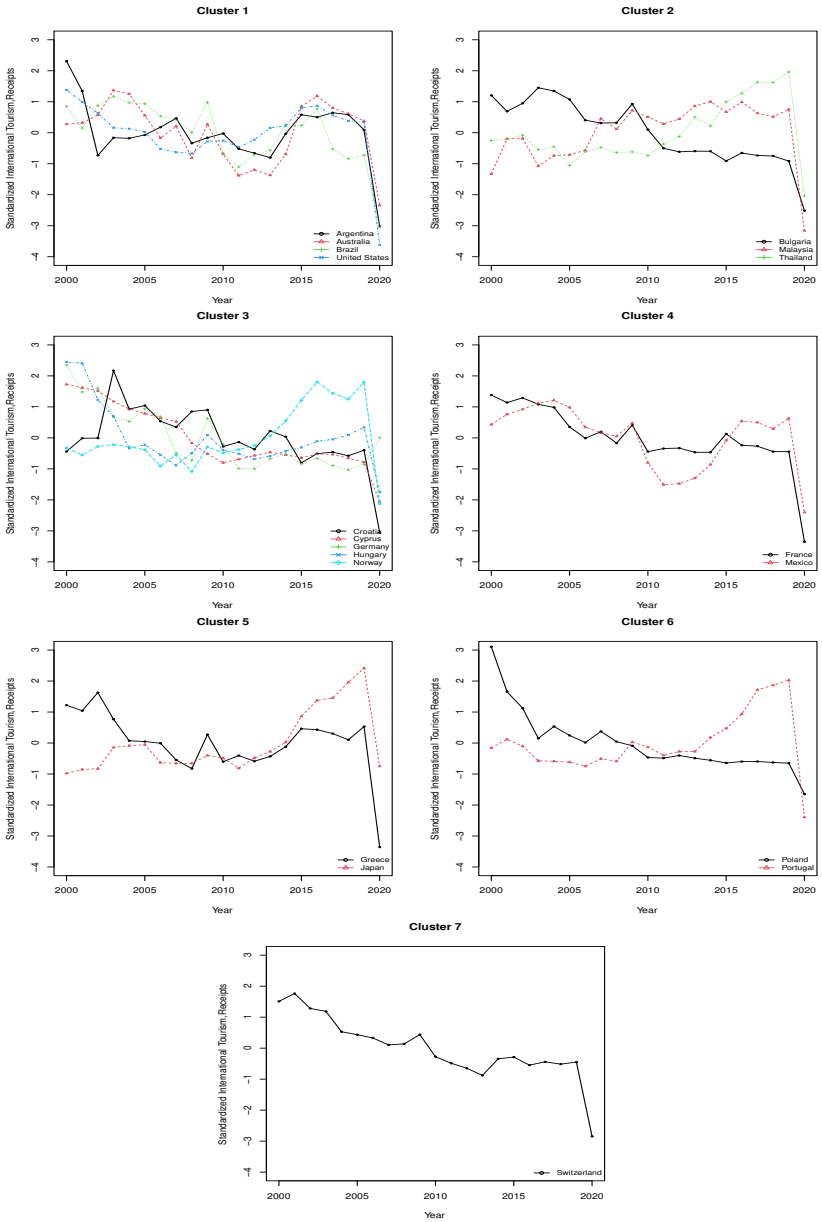

FIGURE 2. Trend plots of standardized international tourism receipts time series in 7 clusters obtained by GDDCICA$_{g_2,C}$ and then using the PAM.

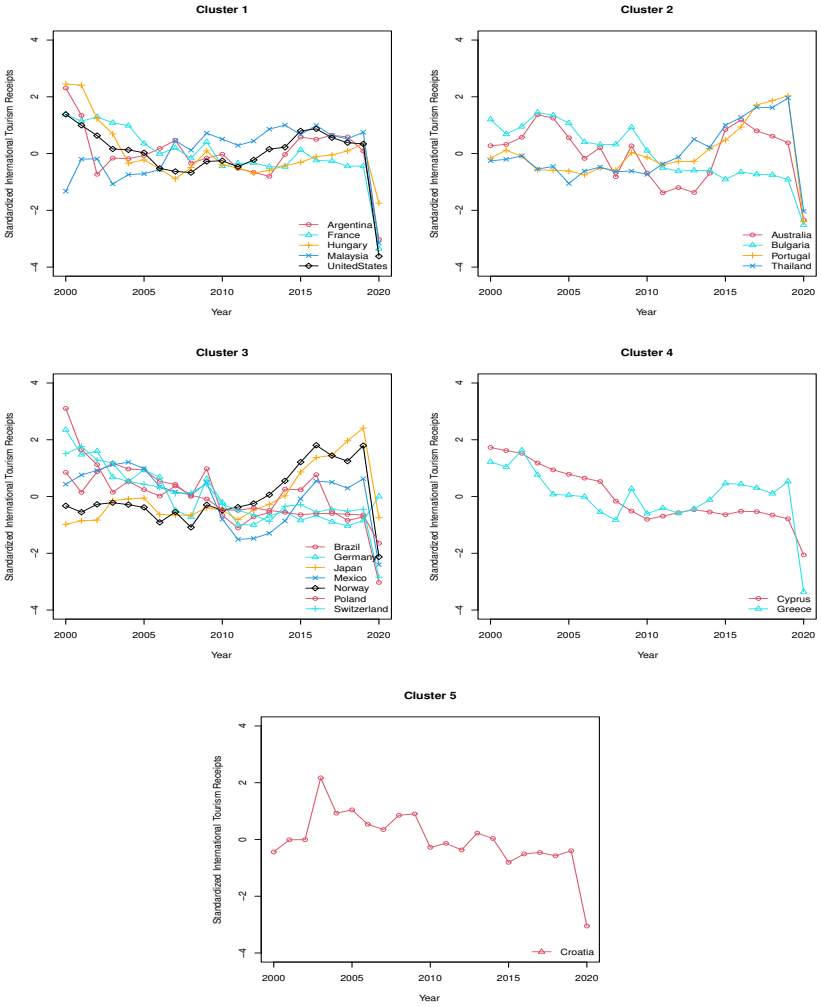

FIGURE 3. Trend plots of standardized international tourism receipts time series in 5 clusters obtained by the direct using of the PAM on the data matrix.

10. **Author Contributions**

Fatemeh Asadi: original draft preparation, conceptualization, methodology, and resources. Hamzeh Torabi: original draft preparation, methodology, review and editing. Hossein Nadeb: Programming and data curation.

11. Data Availability Statement

We considered a dataset including international tourism receipts (% of total exports) from 2000 to 2020 for 19 various countries. The dataset was selected from the organization of the World Bank (`https://data.worldbank.org`) and including the countries Argentina, Australia, Brazil, Bulgaria, Croatia, Cyprus, France, Germany, Greece, Hungary, Japan, Malaysia, Mexico, Norway, Poland, Portugal, Switzerland, Thailand, and United States.

12. Aknowledgement

The authors would like to thank the editor-in-chief, associate editor and two anonymous reviewers for their helpful comments and suggestions, which led to the improved presentation of this article significantly.

13. Ethical considerations

The authors avoided from data fabrication and falsification.

14. Funding

No funding received.

15. Conflict of interest

The authors declare no conflict of interest.

References

[1] Abayomi, K., Lall, U., & de la Pena, V. (2009). Copula based independent component analysis. SSRN. http://ssrn.com/abstract=1028822

[2] Amari, S., Cichocki, A., & Yang, H. (1996). A new learning algorithm for blind source separation. Advances in Neural Information Processing Systems, 8, 757-763. https://proceedings.neurips.cc/paper/1995/hash/e19347e1c3ca0c0b97de5fb3b690855a-Abstract.html

[3] Antony, M.J., Sankaralingam, B.P., Mahendran, R.K., Gardezi, A.A., Shafiq, M., Choi, J.G., & Hamam, H. (2022). Classification of EEG using adaptive SVM classifier with CSP and online recursive independent component analysis. Sensors (Basel, Switzerland), 22(19), 7596. https://doi.org/10.3390/s22197596

[4] Bach, F.R., & Jordan, M.I. (2002). Kernel independent component analysis. Journal of Machine Learning Research, 3, 1-48. https://www.jmlr.org/papers/volume3/bach02a/bach02a.pdf

[5] Bouezmarni, T., & Rolin, J.M. (2003). Consistency of the beta kernel density function estimator. The Canadian Journal of Statistics, 31(1), 89-98. https://doi.org/10.2307/3315905

[6] Cardoso, J.F. (1999). High-order contrasts for independent component analysis. Neural Computation, 11(1), 157-192. https://doi.org/10.1162/089976699300016863

[7] Charpentier, A., Fermanian, J.D., & Scaillet, O. (2007). The estimation of copulas: Theory and practice. In: Jörn Rank Copulas: From Theory to Application in Finance, 35-64. Risk Books, London.

[8] Chen, S.X. (1999). Beta kernel estimators for density functions. Computational Statistics & Data Analysis, 31(2), 131-145. https://doi.org/10.1016/S0167-9473(99)00010-9

[9] Chen, R.B., Guo, M., Härdle, W.K., & Huang, S.F. (2007). Independent component analysis via copula techniques. SFB 649 Discussion Paper 2008-004, SSRN. https://ssrn.com/abstract=2894312

[10] Comon, P. (1994). Independent component analysis, a new concept?. Signal Processing, 36(3), 287-314. https://doi.org/10.1016/0165-1684(94)90029-9

[11] Guo, C., Jia, H., & Zhang, N. (2008). Time series clustering based on ICA for stock data analysis. In 2008 4th international conference on wireless communications, networking and mobile computing, IEEE, 1-4. https://doi.org/10.1109/WiCom.2008.2534

[14] Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks, 10(3), 626-634. https://doi.org/10.1109/72.761722

[13] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. Neural Networks, 13(4-5), 411-430. https://doi.org/10.1016/S0893-6080(00)00026-5

[14] Jayabal, V., Badier, J. M., Pizzo, F., Villalon, S.M., Papageorgakis, C., López-Madrona, V., ... & Bénar, C.G. (2022). Virtual MEG sensors based on beamformer and independent component analysis can reconstruct epileptic activity as measured on simultaneous intracerebral recordings. Neuro Image, 264, 1-17. https://doi.org/10.1016/j.neuroimage.2022.119681

[15] Keziou, A., Fenniri, H., Ghazdali, A., & Moreau, E. (2014). New blind source separation method of independent/dependent sources. Signal Processing, 104, 319-324. https://doi.org/10.1016/j.sigpro.2014.04.017

[16] Langlois, D., Chartier, S., & Gosselin, D. (2010). An introduction to independent component analysis: Infomax and FASTICA algorithms. Tutorials in Quantitative Methods for Psychology, 6(1), 31-38. https://doi.org/10.20982/tqmp.06.1.p031

[17] Lassance, N., DeMiguel, V., & Vrins, F. (2022). Optimal portfolio diversification via independent component analysis. Operations Research, 70(1), 55-72. https://doi.org/10.1287/opre.2021.2140

[18] Learned-Miller, E.G., & Iii, J.W.F. (2003). ICA using spacings estimates of entropy. Journal of Machine Learning Research, 4, 1271-1295. https://www.jmlr.org/papers/volume4/learned-miller03a/learned-miller03a.pdf

[19] Lee, T.W., Girolami, M., & Sejnowski, T.J. (1999). Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. Neural Computation, 11(2), 417-441. https://doi.org/10.1162/089976699300016719

[20] Lipshutz, D., Pehlevan, C., & Chklovskii, D.B. (2022). Biologically plausible single-layer networks for nonnegative independent component analysis. Biological Cybernetics, 116(5), 557-568. https://doi.org/10.1007/s00422-022-00943-8

[21] Lyu, Q., & Fu, X. (2022). On finite-sample identifiability of contrastive learning-based nonlinear independent component analysis. In: Proceedings of the 39^{th} International Conference on Machine Learning, 162, 14582-14600. Baltimore, Maryland, USA, PMLR. https://proceedings.mlr.press/v162/lyu22a.html

[22] Meng, X., Iraji, A., Fu, Z., Kochunov, P., Belger, A., Ford, J., ... & Calhoun, V.D. (2022). Multimodel order independent component analysis: a data-driven method for evaluating brain functional network connectivity within and between multiple spatial scales. Brain Connectivity, 12(7), 617-628. https://doi.org/10.1089/brain.2021.0079

[23] Moneta, A., & Pallante, G. (2022). Identification of structural var models via independent component analysis: a performance evaluation study. Journal of Economic Dynamics and Control, 144, 1-37. https://doi.org/10.1016/j.jedc.2022.104530

[24] Nagler, T. (2014). Kernel methods for Vine copula estimation. Master's Thesis, Technische Universität München. https://mediatum.ub.tum.de/doc/1231221/1231221.pdf

[25] Nelsen, R.B. (2007). An introduction to copulas, Springer Science & Business Media, New York.
[26] Rahmanishamsi, J., & Dolati, A. (2018). Rank based least-squares independent component analysis. Journal of Statisical Research of Iran, 14(2), 247-266. https://doi.org/10.29252/jsri.14.2.247
[27] Rahmanishamsi, J., Dolati, A., & Aghabozorgi, M.R. (2018). A copula based ICA algorithm and its application to time series clustering. Journal of Classification, 35(2), 230-249. https://doi.org/10.1007/s00357-018-9258-x
[28] Rdusseeun, L.K.P.J., & Kaufman, P. (1987). Clustering by means of medoids. In: Proceedings of the Statistical Data Analysis Based on the L1 Norm Conference, 31, 405-416. Neuchatel, Switzerland.
[29] Rincourt, S.L., Michiels, S., & Drubay, D. (2022). Complex disease individual molecular characterization using infinite sparse graphical independent component analysis. Cancer Informatics, 21, 1-16. https://doi.org/10.1177/11769351221105776
[30] Rousseeuw, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7
[31] Shahina, K., & Pradeep Kumar, T.S. (2022). Similarity-based clustering and data aggregation with independent component analysis in wireless sensor networks. In: Transactions on Emerging Telecommunications Technologies, 33(7), 1-15. https://doi.org/10.1002/ett.4462
[32] Shang, L., Zhang, Y., & Sun, Z.L. (2022). Palmprint feature extraction utilizing WTA-ICA in contourlet domain. In: International Conference on Intelligent Computing, 464-471. Xi'an, China. https://doi.org/10.1007/978-3-031-13870-6_39
[33] Sheikh, M.S., & Regan, A. (2022). A complex network analysis approach for estimation and detection of traffic incidents based on independent component analysis. Physica A: Statistical Mechanics and its Applications, 586, 126504. https://doi.org/10.1016/j.physa.2021.126504
[34] Suzuki, T., & Sugiyama, M. (2011). Least-squares independent component analysis. Neural Computation, 23(1), 284-301. https://doi.org/10.1162/NECO_a_00062
[35] Tabanfar, Z., Ghassemi, F., & Moradi, M.H. (2022). Estimating brain periodic sources activities in steady-state visual evoked potential using local fourier independent component analysis. Biomedical Signal Processing and Control, 71, 103162. https://doi.org/10.1016/j.bspc.2021.103162
[36] Wand, M.P., & Jones, M.C. (1995) Kernel smoothing. Chapman & Hall, Boca Raton.
[37] Zanghaei, A., Abolhassani, M., Ahmadian, A., Ay, M.R., & Saberi, H. (2013). A New nethod to enhance the clustering algorithm. International Journal of Computer and Electrical Engineering, 5(1), 120-122. https://doi.org/10.7763/IJCEE.2013.V5.677
[38] Zhang, S., & Karunamuni, R.J. (2010). Boundary performance of the beta kernel estimators. Journal of Nonparametric Statistics, 22(1), 81-104. https://doi.org/10.1080/10485250903124984
[39] Zhou, R., Han, J., Li, T., & Guo, Z. (2022). Fast independent component analysis denoising for magnetotelluric data based on a correlation coefficient and fast iterative shrinkage threshold algorithm. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-15. https://doi.org/10.1109/TGRS.2022.3182504

Fatemeh Asadi
Orcid number: 0009-0003-1534-9256
Department of Statistics
Yazd University
Yazd, Iran
Email address: asadi.f.2005@gmail.com

Hamzeh Torabi
Orcid number: 0000-0002-4200-9770
Department of Statistics
Yazd University
Yazd, Iran
Email address: htorabi@yazd.ac.ir

Hossein Nadeb
Orcid number: 0000-0003-4066-5309
Department of Statistics
Yazd University
Yazd, Iran
Email address: honadeb@yahoo.com