International Journal of Computational Economics and Econometrics

ISSN online: 1757-1189 - ISSN print: 1757-1170
https://www.inderscience.com/ijcee

A new approach for independent component analysis and its application for clustering the economic data

Fatemeh Asadi, Hamzeh Torabi, Hossein Nadeb

DOI: 10.1504/IJCEE.2025.10069874

Article History:

Received:	30 January 2024
Last revised:	25 June 2024
Accepted:	26 June 2024
Published online:	17 March 2025

A new approach for independent component analysis and its application for clustering the economic data

Fatemeh Asadi, Hamzeh Torabi* and Hossein Nadeb

Department of Statistics,
Yazd University,
Yazd, Iran
Email: Asadi.f.2005@gmail.com
Email: htorabi@yazd.ac.ir
Email: honadeb@yahoo.com
*Corresponding author

Abstract: In conventional independent component analysis (ICA) algorithms, the definition of the objective function is typically based on specific dependency criteria. The choice of these criteria significantly influences the performance of the algorithm. This article introduces a general class of dependency criteria, which is based on the cumulative distribution function, to characterise the independence of two variables. Furthermore, an applicable ICA algorithm, grounded in this class and utilising a non-parametric estimator, is proposed. The performance of the proposed algorithm is evaluated and compared with several well-known traditional algorithms, using Amari error estimation calculation as a benchmark. The proposed algorithms have been applied to a real-time series data, serving as a pre-processing clustering method.

Keywords: Amari error; clustering; dependence criteria; independent components analysis.

Reference to this paper should be made as follows: Asadi, F., Torabi, H. and Nadeb, H. (2025) 'A new approach for independent component analysis and its application for clustering the economic data', *Int. J. Computational Economics and Econometrics*, Vol. 15, Nos. 1/2, pp.147–171.

Biographical notes: Fatemeh Asadi is a PhD student in Statistics at Yazd University, Yazd, Iran. She has received her BSc in Statistics from the Isfahan University of Technology in 2008, and her MSc in Mathematical Statistics from the University of Isfahan in 2010. She is currently a data analyst and researcher. Her expertise lies in applied statistics, data science, data mining, and data analysis algorithms, particularly independent component analysis (ICA).

Hamzeh Torabi is currently a Full Professor of Statistics at the Yazd University, Yazd, Iran. He obtained his BSc in Statistics from the Shiraz University in 1994 and his MSc from the Kharazmi University (Tarbiat Moallem-e Tehran), Tehran, Iran in 1996. Later, he received his PhD in Statistics from the Department of Statistics, Shiraz University, Shiraz, Iran in 2005. His research interests are data science, distribution theory, reliability theory, survival analysis for complete and censored data, new methods for estimating and hypotheses testing, static and dynamic simulation, goodness-of-fit tests, quality control, computational statistics, and fuzzy probability and statistics.

Hossein Nadeb is a member of R&D unit in Iran Central Iron Ore Company and an invited member of Department of Statistics, Yazd University, Iran. He has a BSc in Statistics from the Yazd University in 2012, an MSc in Mathematical Statistics from the Yazd University in 2014 and a PhD in Statistics from the Yazd University in 2020. His research interest includes statistical inference based on incomplete data, goodness of fit testing, reliability theory and stochastic orderings.

1 Introduction

Multivariate data, due to its high dimensionality and complex interdependencies among observed values, presents significant challenges for analysis. Therefore, it becomes essential to segregate the data into different aspects to discern patterns effectively. Independent component analysis (ICA) has been extensively utilised as an unsupervised tool for blind source separation in multivariate statistics, particularly for analysing mixtures of independent source signals (Pfister et al., 2019). ICA was first introduced by Hérault in the 1980s, with further developments made by Hyvärinen et al. (2001) and Comon (1994). The application of ICA on signal data has been explored by Comon and Jutten (2010) and Nordhausen and Oja (2018).

ICA is a matrix factorisation technique where the signals captured by each individual matrix factor are optimised to be as mutually independent as possible. This method has been highlighted in various works for its applications in dimensionality reduction, deconvolution, data pre-processing, meta-analysis, and other areas across different data types (Sompairac et al., 2019). The primary objective of ICA is to extract significant components from a dataset, which can encompass various forms such as sounds, stock markets, or images.

ICA has been utilised across multiple fields for the extraction of independent components. These include network analysis for traffic incident detection and estimation (Sheikh and Regan, 2022), optimal portfolio diversification (Lassance et al., 2022), identification of structural autoregressive models in macroeconomics (Moneta and Pallante, 2022), and magnetotelluric sounding data analysis (Zhou et al., 2022). Furthermore, it has been employed for clustering in wireless sensor networks (Shahina and Kumar, 2022), image feature extraction (Shang et al., 2022), and various medical applications, such as brain activity analysis (Tabanfar et al., 2022; Meng et al., 2022; Jayabal et al., 2022), electroencephalogram (EEG) analysis (Lyu and Fu, 2022; Antony et al., 2022), individual molecular characterisation (Rincourt et al., 2022), and the dissociation of biologically single-layer networks (Lipshutz et al., 2022), among others.

ICA has found applications in diverse scientific fields, including medical signals (Xie et al., 2017; Pontifex et al., 2017), audio signals for noise elimination (Du and Swamy, 2014), robust automatic speech discernment model projection (Cho and

Park, 2016), biological assays (Aziz et al., 2016), space image reception (Naik and Kumar, 2011), climate data analysis (Funaro et al., 2001), financial data (Moneta et al., 2010), and socio-economic data, work environment, and employment rate (Tan and Zhang, 2012). It is also used as a pre-processing step in clustering time series data (Rahmanishamsi et al., 2018; Nascimento et al., 2017).

ICA is a computational technique utilised for decomposing a multivariate signal into constituent, additive sub-components. It aims to identify independent components by maximising the statistical independence among the estimated elements. The formulation of the ICA algorithm hinges upon the selection of a metric for independence, with two primary paradigms being the minimisation of mutual information and the maximisation of non-Gaussianity.

This study introduces a novel set of dependency criteria founded on the distribution function to characterise the independence of two random variables and examine their properties. Additionally, an ICA algorithm is proposed based on an approach derived from these novel criteria. To assess the efficacy of the proposed algorithm, a comparative analysis is conducted between the new algorithm and existing methods. Finally, the algorithm is employed as a pre-processing step for ICA in clustering a batch of time series examples.

In the upcoming section, we will delve into the concept of ICA. Following this, an exploration into a dependency criterion based on distribution functions will be introduced and substantiated. Section 4 will unveil several new algorithms designed for ICA. Their comparative performance against existing ICA algorithms will be evaluated in terms of the average of Amari errors via a Monte Carlo simulation survey, presented in Section 5. Section 6 will focus on the application of these algorithms in time series data clustering. Furthermore, this study delves into the application of data clustering methodologies to a set of real-time series data, specifically the gross domestic product (GDP) per capita index, extracted from the statistics of 26 countries spanning from 1975 to 2020. The conclusion of the research is drawn by summarising the results and analysing the structures extracted, as detailed in Section 7.

2 Independent component analysis

Given the complexity and high dimensionality inherent in multivariate data analysis, it often presents challenges in its application. Therefore, it becomes essential to segregate different facets of the data for effective utilisation, particularly in the context of identifying specific patterns in such scenarios (Tharwat, 2021).

In this study, ICA has been employed for signal analysis. Consider each signal as a time-varying vector, denoted as $\boldsymbol{s_i} = (s_{1i}, s_{2i}, \ldots, s_{ni})^T, i = 1, 2, ..., d$ represent a signal where n is the number of time steps, s_{ij} is time j of the signal $\boldsymbol{s_i}$, and d is the number of source signals. Here, (d) also represents independent source signals, and a matrix $S = (\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_d})$ is defined, where $S \in \mathbb{R}^{n\times d}$ is matrix of the source signals. These source signals may be mixed, and each source signal influencing the output signals in different ways. Therefore, d mixtures can be represented as $X =: SA$, where $X \in \mathbb{R}^{n\times d}$ is the matrix of the mixed signals, n is the number of mixes, and $A \in \mathbb{R}^{d\times d}$ is a mixing coefficients matrix. S into a mixed signal in $X = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_d})$ space as $X = SA$. The objective of the model is defined as the extraction of source signals by finding the unmixing coefficient matrix (W). This matrix is used to convert the mixed signals into a set of independent signals as as $X \rightarrow Y : Y = XW$, where $W \in \mathbb{R}^{d \times d}$ is the matrix of unmixing coefficients, and it is inverse of the matrix A.

Based on the estimation principles of ICA, the stages of centralisation and whitening are commonly employed. Data whitening involves transforming mixed signals into uncorrelated signals and scaling them by their respective variances. This whitening process is a step that is essential for preparing the data for the ICA algorithm. Notably, the principal component analysis (PCA) method can be used to whiten the output signals in ICA. The PCA has been used by Ismail et al. (2009), Anbumalar et al. (2011), Kaur and Vashisht (2011), Sivaramakrishna (2011) and Rahmanr et al. (2012).

In various research studies, different types of ICA procedures and their interpretations have been explored. Most ICA algorithms aim to minimise a contrast function that measures the degree of dependency between components. The effectiveness of these algorithms is contingent upon the choice of the contrast function and the algorithm used to implement the optimisation problem. The estimation of the unmixing matrix (W matrix) can be achieved through several main approaches to independence, leading to the creation of unmixed matrices with slight variations. In all these approaches, an unmixed matrix is obtained, and the whitened data is projected onto this matrix to extract the independent signals. For further details, the works of Tharwat (2021), Pfister et al. (2019) and Stone (2004) are recommended for reference.

To estimate the independent components, several methods have investigated in previous re- searches. Based on Hyvärinen et al. (2001), maximising non-Gaussianity by negative entropy and estimating the W matrix by the maximum likelihood. Minimising the mutual information (MI) criterion (Hyvärinen, 1999) has represented one of the most popular ICA estimation perspectives. There are several algorithms constructed base on minimising MI to estimate the independent components (Langlois et al., 2010).

Some of the most applicable algorithm in these perspectives are FastICA algorithm based on MI minimisation (Hyvärinen, 1999), Infomax algorithm based on the maximum likelihood method (Lee et al., 1999), JADE algorithm based on maximising kurtosis (Cardoso, 1999), RADICAL algorithm based on the kullback leibler criterion (Learned-Miller and Fisher, 2003), HICA algorithm based on the copula function of Hoeffding's criterion (Rahmanishamsi et al., 2018) and RLICA algorithm based on the copula function of the squared loss MI criterion (Rahmanishamsi and Dolati, 2018). In the next section, some ICA approaches based on minimising MI will be investigated.

The estimation of independent components in ICA involves various methods and criteria. One approach, as proposed by Bingham and Hyvärinen (2000), is based on non-Gaussianity, measured by kurtosis and negative entropy (negentropy), with the aim of maximising non-Gaussianity to find the independent components (Shimizu et al., 2006). Additionally, the maximum likelihood (ML) method is utilised in ICA to estimate the matrix W, providing the best fit for the extracted Y signals – see: Gaeta (1990), Pham et al. (1992) and Pearlmutter and Parra (1996). Due to the mutual information (MI) criterion measures the independence between two random variables, many algorithms for ICA are constructed based on minimising MI to estimate the initial signals (Langlois et al., 2010). Therefore, independent components can be obtained by minimising MI between different components; see Pfister et al. (2019) and Tharwat (2021). In this setting, the matrix W is obtained such that the MI criterion to be close to zero.

3 Proposed dependency criterion

The formulation of the objective function in ICA traditionally relies on density or copula density functions. However, accurately estimating these functions poses significant challenges due to inherent errors in the estimation process. A potential resolution to this issue lies in the utilisation of cumulative distribution functions (CDFs) within ICA methodologies. Unlike non-parametric-based density function estimators, empirical distribution functions (EDFs) offer a notably simpler and faster approach in statistical inference. EDFs exhibit a convergence towards theoretical cumulative distribution functions, enhancing their efficacy in approximating dependencies between variables. The use of empirical distribution functions, which converge to theoretical CDFs, holds promise in evaluating the independence between two random variables. This approach proves particularly valuable when assessing a dependency criterion for characterising independence between variables.

Drawing from these interpretations, an investigation centred on a criterion employing cumulative distribution functions has been undertaken. This criterion aims to characterise the independence between two random variables by evaluating their equality to zero.

To explore this function, consider two random variables, X_1 and X_2, with a joint distribution function F, and individual marginal distribution functions F_1 and F_2, respectively. Let $g : \mathbb{R}^+ \to \mathbb{R}^+$ be a decreasing function for $x < 1$, increasing for $x \geq 1$, and $g(1) = 0$. A generalised dependency criterion denoted as GDC_g, based on cumulative distribution functions, is defined as follows:

$$GDC_g(X_1, X_2) = \sup_{(x_1,x_2)\in\mathbb{R}^2} \left[g\left(\frac{F(x_1, x_2)}{F_1(x_1)F_2(x_2)} \right) \right].$$

Notice that these divergences will vanish if and only if the random variables X_1 and X_2 are independent.

Let $X_{n\times 2}$ be the matrix of observation from random vector (X_1, X_2) with the joint distribution function F and the marginal cumulative distribution functions F_1 and F_2, respectively. Consider $x_{ij} = [X]_{ij}$, where $i = 1, 2, ..., n$, $j = 1, 2$. The functions denoted as $\hat{F}_1(x_1) = \frac{\sum_{i=1}^{n} I(x_{i1} \leq x_1)}{n}$ and $\hat{F}_2(x_2) = \frac{\sum_{i=1}^{n} I(x_{i2} \leq x_2)}{n}$ correspond to the marginal distribution functions, while $\hat{F}(x_1, x_2) = \frac{\sum_{i=1}^{n} I(x_{i1} \leq x_1, x_{i2} \leq x_2)}{n}$ represents the joint empirical distribution function. Consequently, the GDC_g can be estimated as:

$$\widehat{GDC}_g(X_1, X_2) = \sup_{(x_1,x_2)\in\mathbb{R}^2} \left[g\left(\frac{\hat{F}(x_1, x_2)}{\hat{F}_1(x_1)\hat{F}_2(x_2)} \right) \right]. \tag{1}$$

4 An ICA algorithm based on $\widehat{GDC}_g$ (GDCICA)

In this section, the estimator denoted as $\widehat{GDC}_g$ for GDC_g in equation (1) has been utilised in the development of several new algorithms for ICA. Let $\boldsymbol{X}$ be a d-dimensional random vector. ICA aims to find a matrix denoted as $W \in \mathbb{R}^{d\times d}$ such that the components $Y_1, Y_2, \ldots, Y_d$ of the new random vector $\mathbf{Y} = \mathbf{X}W$ exhibit minimal statistical dependency. A contrast function within an ICA algorithm is typically based on a measure of dependency, represented as f, and is defined as $\varphi_X(.)$, satisfying $\varphi_X(W) = f(XW)$. Consequently, the core challenge in ICA involves determining $\widehat{W} \in \mathbb{R}^{d\times d}$ that minimises the contrast function $\varphi_X(W)$.

For statistical independence in ICA problem, pairwise independence refers to a sufficient measure (Comon, 1994). Thus a d-dimensional ICA problem solution defines as a generalised successively solution for a 2-dimensional ICA problems. In other words, a d-dimensional linear transformation described by a $d \times d$ orthogonal matrix R defines as equivalent to a composition of 2-dimensional rotations. Furthermore, the idea for searching a rotation angle such that the corresponding demixed dataset has its GICA$_g$ minimised, is similar to the RADICAL algorithm (Learned-Miller and Fisher, 2003).

Based on previous definitions, the general process of the algorithm for the d-dimensional case is summarised as follows:

Algorithm 4.1:

Input: A matrix $X_{n\times d}$ where rows are mixed signals (centred).

Procedure for calculate GDCICA**:**

1 Whiten the matrix X of the form $Y = X \times Q'$, where Q is a whitening matrix;

2 Define the matrices $\widehat{W}$ and $\widehat{S}$ with all entries zero, as the initial matrices for unmixing and source signals, respectively.

3 Set $R_{ij}(\theta)$, a rotation matrix of elements (i, j), of the form

$$R_{ij}(\theta) = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos(\theta) & 0 & \cdots & 0 & -\sin(\theta) & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & \sin(\theta) & 0 & \cdots & 0 & \cos(\theta) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix}, \quad (2)$$

For all $1 \leqslant i < j \leqslant d$, repeat the following steps:

4 Define $\widehat{GDC}_g(S_i(\theta), S_j(\theta))$ as a function of θ, where it is assumed that $S^{(i,j)}(\theta)$ which denotes the $(i, j)^{\text{th}}$ columns of $Y \times R_{i,j}(\theta)$, is a sample from $(S_i(\theta), S_j(\theta))$.

5 Minimise the function $\widehat{GDC}_g(S_i(\theta), S_j(\theta))$ over $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and set

$$\theta_0^{(i,j)} = \underset{\theta}{\operatorname{argmin}}\, \widehat{GDDC}_g(S_i(\theta), S_j(\theta)).$$

6 Update $(i,j)^{\text{th}}$ columns of $\widehat{W}$ as $R^{'}_{i,j}(\theta_0^{(i,j)}) \times Q$, and $(i,j)^{\text{th}}$ columns of $\widehat{S}$ as $Y \times R(\theta_0^{(i,j)})$.

Output: The unmixing matrix $\widehat{W}$ and the matrix of source signal estimates $\widehat{S}$.

5 Simulation study

To conduct a simulation study, the Monte Carlo simulation method has employed to measure the performance of GDCICA using various algorithms such as FastICA (Hyvärinen, 1999), Infomax (Lee et al., 1999), JADE (Cardoso, 1999), RADICAL (Learned-Miller and Fisher, 2003), HICA (Rahmanishamsi et al., 2018) and RLICA (Rahmanishamsi and Dolati, 2018) within the `R` software environment.

The data generation process followed these steps:

- Generate a matrix $S_{n\times d}$, where n denoted the number of observations and d denoted the number of variables ($n \geq d$), also each column comprises a random sample of size n drawn from a specific distribution. Those incorporated 18 distinct one-dimensional densities recommended by Bach and Jordan (2002), encompassing distributions such as student-t, uniform, exponential, a mixture of two Laplace densities, symmetric and non-symmetric Gaussian mixtures. Figure 1 illustrates the density plots of these 18 distributions.
- Generate a random matrix

  $$A_{d\times d} = \begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ U_{d1} & U_{d2} & \cdots & U_{dd} \end{pmatrix},$$

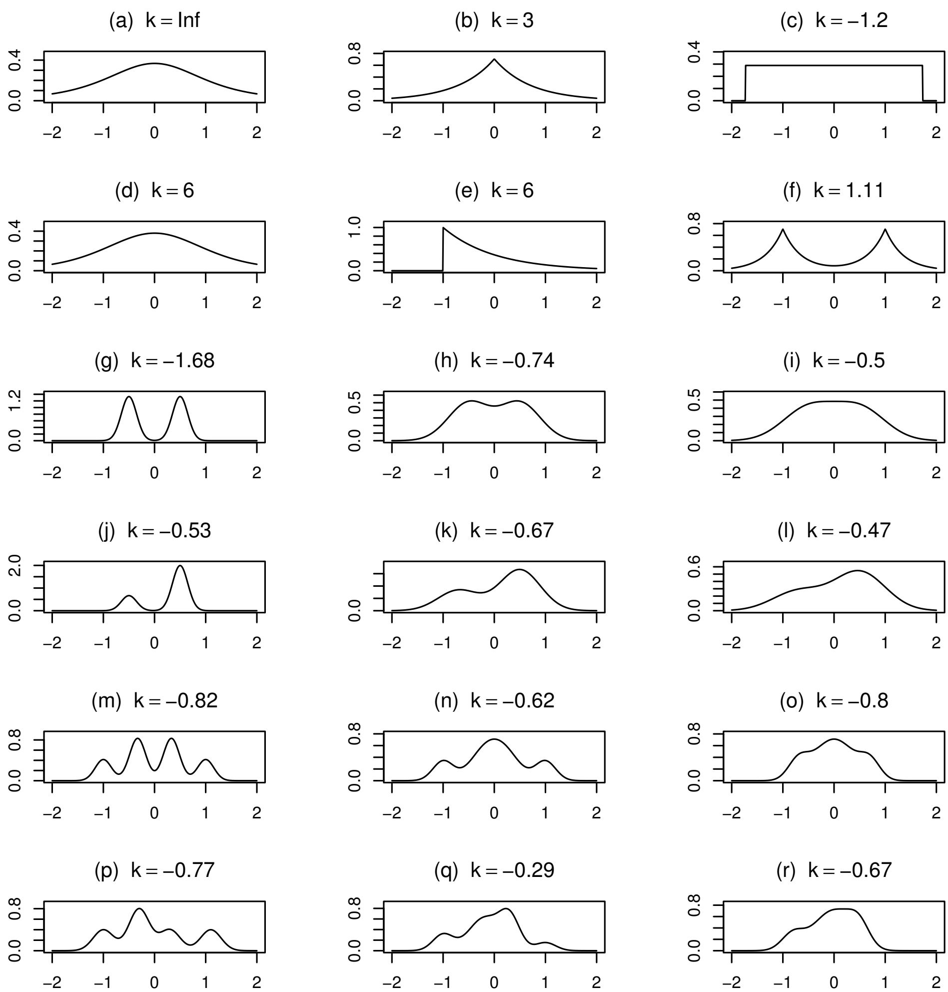
  where $U_{ij}, i, j = 1, 2, \cdots, d$, independently follows a arbitrary random distribution.
- Construct a data matrix $X_{n\times d}$ using the formula $X = SA$.
- Whiten the matrix X of the form $Y = X \times Q'$. It can be whitened by multiplying X by the inverse of the square root of the sample covariance matrix, resulting in a matrix of whitened data $X_{n\times d}$. This whitened data matrix (Q) is the input for the GDCICA algorithms.

Subsequently, all ICA algorithms generated an unmixing matrix W, which was applied to matrix X to recover estimations of the independent components. The Amari error, initially proposed by Amari et al. (1996), serves as a robust method for error assessment in this context. Let A represent the mixing matrix, while the unmixing matrix is defined through the equation $W = A^{-1}$ using $\widehat{W}$ estimator. The Amari error computation is based on the following expression:

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

Here, the symbol $a_{ij} = [\widehat{W}A]_{ij}$ represents an estimated unmixing matrix $\widehat{W}$. Under these assumptions, the following conditions hold true within this expression: the metric $[0, d-1]$ consistently resides within the range of Amari error, also equating to zero solely if $\widehat{W}$ and W denote similar components. Furthermore, this metric remains invariant to the permutation and scaling of columns within A and $\widehat{W}$.

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) non-symmetric mixtures of two Gaussians, multimodal, transitional and unimodal (m, n, o) symmetric mixtures of four Gaussians: multimodal, transitional and unimodal (p, q, r) non-symmetric mixtures of four Gaussians: multimodal, transitional and unimodal



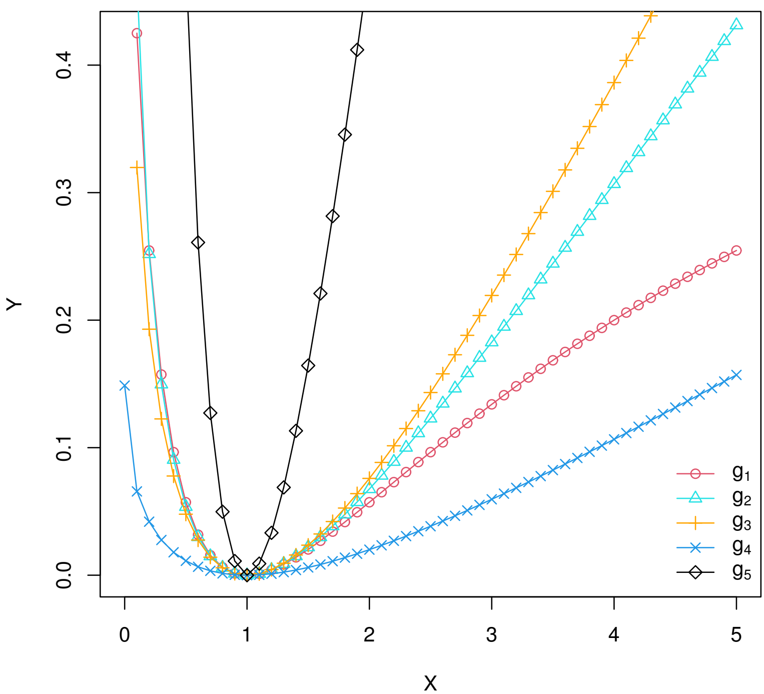
Within GDCICA, certain functions are regarded as special cases of the function g outlined in equations below:

$$g_1(x) = \frac{(x^{c_1} - 1)^2}{x^{2c_1} + 1}, \qquad g_2(x) = -\ln(x^{c_2}) + x^{c_2} - 1,$$
$$g_3(x) = x^{c_3} \ln(x^{c_3}) - x^{c_3} + 1, \qquad g_4(x) = e^{c_4(x^{c_4^*} - 1)} - c_4(x^{c_4^*} - 1) - 1,$$
$$g_5(x) = \ln(x)^{2c_5},$$

where c_1, c_2, c_3, c_4, c_4^* and c_5 are some real values.

In GDCICA, a comprehensive exploration of various constants c_1, c_2, c_3, c_4, c_4^* and c_5 has been conducted. The findings reveal that employing specific functions g_1, g_2, g_3, g_4, g_5 with corresponding $c_1 = -0.5$, $c_2 = 0.5$, $c3 = 0.5$, $c_4 = -0.5$, $c_4^* = 0.5$, and $c_5 = 1$ values yields commendable performance. Specifically, denoting GDCICA$_g$ with $c_1 = -0.5$ is denoted by GDCICA$_{g_1}$, with $c_2 = 0.5$ is denoted by GDCICA$_{g_2}$, with $c_3 = 0.5$ is denoted by GDCICA$_{g_3}$, with $(c_4, c_4^*) = (-0.5, 0.5)$ is denoted by GDCICA$_{g_4}$, with $c_5 = 1$ is denoted by GDCICA$_{g_5}$. The selected functions are visually presented in Figure 2.

Figure 2 Plots of five different functions in this research (see online version for colours)



In generating 2-dimensional and 4-dimensional datasets of size $n = 250, n = 500, n =$ 1,000, and $n =$ 1,500, governed by the mixing matrix

$$A_{2\times2} = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} \quad \text{and} \quad A_{4\times4} = \begin{pmatrix} U_{11} & U_{12} & U_{13} & U_{14} \\ U_{21} & U_{22} & U_{23} & U_{24} \\ U_{31} & U_{32} & U_{33} & U_{34} \\ U_{41} & U_{42} & U_{43} & U_{44} \end{pmatrix},$$

where $U_{ij}, i = 1, 2, 3, 4$ independently follows a uniform distribution over $(0, 2)$, a simulation was conducted. Subsequently, the average Amari errors of methods such as FastICA, Infomax, JADE, RADICAL, HICA, RLICA and GDCICA$_g$, and others were computed over 120 replications of their respective algorithms.

The outcomes are presented in Tables 1 and 4, illustrating the average errors across various samples. The findings indicate that across 18 cases, each distinct type of GDCICA$_g$ method consistently outperforms its competitors in seven instances, collectively excelling in distributions c, f and j in both 2-dimensional and 4-dimensional situations.

Further elucidation on this investigation is provided below:

- When $n = 250, d = 2$, the GDCICA$_{g_2}$, GDCICA$_{g_3}$ and GDCICA$_{g_4}$ methods exhibited the lowest Amari errors, that this methods applied to data cases with distributions labelled as a, c, e, f, g, j and m have yielded the best performance, also for $n = 250, d = 4$, the GDCICA$_{g_2}$, GDCICA$_{g_3}$ and GDCICA$_{g_4}$ methods exhibited the lowest Amari errors, that this methods applied to data cases with distributions labelled as a, c, e, f, j, m and n have yielded the best performance.
- When $n = 500, d = 2$, the GDCICA$_{g_2}$ and GDCICA$_{g_4}$ demonstrated the most favourable Amari errors, that this methods applied to data cases with distributions labelled as a, c, f, g, j and m have yielded the best performance, also for $n = 500, d = 4$, the GDCICA$_{g_2}$ and GDCICA$_{g_4}$ demonstrated the most favourable Amari errors, that this methods applied to data cases with distributions labelled as a, c, e, f, j and n have yielded the best performance.
- When $n = 1{,}000, d = 2$, the GDCICA$_{g_2}$ and GDCICA$_{g_3}$ showcased the best Amari errors, that this methods applied to data cases with distributions labelled as a, c, e, f, g, j and m have yielded the best performance, also for $n = 1{,}000, d = 4$, the GDCICA$_{g_2}$ and GDCICA$_{g_3}$ showcased the best Amari errors, that this methods applied to data cases with distributions labelled as a, c, e, f, g, j, n and r have yielded the best performance.
- When $n = 1{,}500, d = 2$, the GDCICA$_{g_2}$, GDCICA$_{g_3}$ and GDCICA$_{g_4}$ again emerged with the most optimal Amari errors, that this methods applied to data cases with distributions labelled as c, f, h, i, j and n have yielded the best performance, also for $n = 1{,}500, d = 4$, the GDCICA$_{g_2}$, GDCICA$_{g_3}$ and GDCICA$_{g_4}$ again emerged with the most optimal Amari errors, that this methods applied to data cases with distributions labelled as c, f, i, j, m and n have yielded the best performance.

Further, two distributions were selected from among all distributions labelled (a) to (r). The average Amari errors regarding these selections were computed and discussed in the final row of Tables 1 and 4. The results have revealed that when a stochastic distribution is chosen, the Amari error for the GDCICA$_{g_2}$ and GDCICA$_{g_4}$ methods yields better performance than other methods.

Table 1 Averages of Amari errors for $d = 2, d = 4$, and $n = 250$ sample with 120 replications for each distribution (a) to (r), the smallest (best) entry of our functions in each row is boldfaced

Distribution	*d*	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
a	2	0.5033	0.5072	0.5000	0.5307	0.5297	0.5052	0.5117	0.5024	**0.4895**	**0.4961**	0.5053
	4	0.4472	0.4471	0.4525	0.4334	0.4540	0.4446	**0.4328**	0.4394	**0.4326**	**0.4328**	0.4344
b	2	0.4701	0.4705	0.4871	0.4652	0.4740	0.4824	0.4825	0.5084	0.4982	0.4977	0.4854
	4	0.4318	0.4319	0.4822	0.4505	0.4466	0.4535	0.4427	0.4435	0.4365	0.4392	0.4391
c	2	0.5052	0.5049	0.5067	0.4947	0.5042	0.5049	**0.4879**	0.5044	**0.4732**	**0.4851**	**0.4790**
	4	0.4455	0.4450	0.4456	0.4342	0.4488	0.4522	0.4391	**0.4340**	**0.4340**	**0.4326**	0.4354
d	2	0.4766	0.4886	0.5015	0.5131	0.5059	0.5125	0.5117	0.5345	0.5164	0.5226	0.5198
	4	0.4507	0.4503	0.4644	0.4209	0.4482	0.4526	0.4374	0.4396	0.4356	0.4365	0.4345
e	2	0.5018	0.5019	0.5254	0.5109	0.5208	0.5201	**0.5015**	**0.4851**	**0.4847**	**0.4686**	**0.5017**
	4	0.4344	0.4339	0.4458	0.4434	0.4506	0.4516	0.4358	**0.4327**	**0.4332**	**0.4338**	0.4358
f	2	0.5571	0.5570	0.5532	0.5647	0.5611	0.5620	**0.5155**	**0.5094**	**0.5166**	**0.5197**	**0.5150**
	4	0.4427	0.4428	0.4426	0.4430	0.4493	0.4475	**0.4417**	**0.4392**	**0.4362**	**0.4411**	0.4468
g	2	0.5345	0.5345	0.5347	0.5308	0.5408	0.5335	**0.5176**	**0.5123**	**0.5152**	**0.5139**	**0.5150**
	4	0.4445	0.4345	0.4451	0.4349	0.4531	0.4537	0.4384	0.4400	0.4352	0.4418	0.4419
h	2	0.5178	0.5313	0.5049	0.5270	0.4977	0.5071	0.5301	0.5330	0.5340	0.5335	0.5292
	4	0.4368	0.4372	0.4621	0.4084	0.4496	0.4512	0.4393	0.4367	0.4388	0.4376	0.4390
i	2	0.5205	0.5161	0.5385	0.5234	0.5238	0.5033	0.5582	0.5439	0.5622	0.5475	0.5549
	4	0.4390	0.4355	0.4617	0.4420	0.4530	0.4527	0.4425	0.4365	0.4402	0.4388	0.4438
j	2	0.4896	0.4803	0.5271	0.4893	0.4808	0.4814	**0.4600**	**0.4603**	**0.4745**	**0.4685**	**0.4701**
	4	0.4451	0.4402	0.4475	0.4433	0.4475	0.4456	0.4429	**0.4370**	**0.4373**	**0.4334**	**0.4312**
k	2	0.5111	0.5107	0.5289	0.5187	0.5305	0.5438	0.5217	**0.5095**	0.5218	0.5112	0.5358
	4	0.4347	0.4347	0.4474	0.4034	0.4542	0.4500	0.4383	0.4405	0.4371	0.4395	0.4414
l	2	0.4769	0.4925	0.5061	0.5355	0.5466	0.5187	0.5648	0.5527	0.5593	0.5498	0.5568
	4	0.4647	0.4595	0.4433	0.4386	0.4488	0.4580	0.4456	0.4414	0.4399	0.4431	0.4478
m	2	0.4374	0.4372	0.4468	0.4631	0.4642	0.4560	**0.4313**	**0.4238**	**0.4268**	**0.4203**	**0.4321**
	4	0.4435	0.4405	0.4366	0.4344	0.4426	0.4492	0.4354	**0.4339**	**0.4327**	**0.4332**	0.4389
n	2	0.4635	0.4577	0.4774	0.4905	0.5285	0.5074	0.5136	0.5156	0.5124	0.5070	0.5104
	4	0.4391	0.4351	0.4325	0.4325	0.4524	0.4532	0.4362	**0.4313**	**0.4307**	0.4393	**0.4316**
o	2	0.5202	0.5259	0.5231	0.5375	0.5395	0.5328	0.5319	0.5269	0.5260	0.5272	0.5255
	4	0.4202	0.4205	0.4313	0.4331	0.4500	0.4501	0.4413	0.4327	0.4347	0.4327	0.4354
p	2	0.5295	0.5165	0.5360	0.4798	0.5046	0.5016	0.4957	0.5141	0.5064	0.4959	0.4916
	4	0.4523	0.4525	0.4501	0.4319	0.4525	0.4529	0.4351	0.4423	0.4371	0.4406	0.4383
q	2	0.5295	0.5165	0.5360	0.4798	0.5046	0.5016	0.4957	0.5141	0.5064	0.4959	0.4916
	4	0.4398	0.4002	0.4332	0.4240	0.4531	0.4448	0.4334	0.4417	0.4391	0.4373	0.4423
r	2	0.5056	0.5010	0.4919	0.5100	0.5061	0.5070	0.5604	0.5388	0.5584	0.5269	0.5596
	4	0.4286	0.4287	0.4315	0.4378	0.4503	0.4512	0.4459	0.4388	0.4372	0.4379	0.4389
rand	2	0.5119	0.5257	0.5175	0.5074	0.5066	0.5152	0.5077	**0.4975**	**0.5014**	**0.4990**	0.5075
	4	0.4314	0.4371	0.4315	0.4394	0.4482	0.4539	0.4369	**0.4302**	**0.4304**	**0.4307**	0.4374

Table 2 Averages of Amari errors for $d = 2, d = 4$, and $n = 500$ sample with 120 replications for each distribution (a) to (r), the smallest (best) entry of our functions in each row is boldfaced

Distribution	*d*	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
a	2	0.5668	0.5668	0.5447	0.5755	0.5538	0.5743	**0.5136**	**0.5232**	**0.5069**	**0.5183**	**0.5191**
	4	0.4422	0.4422	0.4756	0.4643	0.4465	0.4494	0.4406	**0.4393**	0.4450	**0.4378**	0.4455
b	2	0.4927	0.4925	0.4993	0.4923	0.4698	0.4846	0.4914	0.4977	0.4840	0.4876	0.4880
	4	0.4354	0.4354	0.4543	0.4517	0.4538	0.4596	0.4377	0.4393	0.4438	0.4386	0.4412
c	2	0.4929	0.4929	0.4925	0.5050	0.4978	0.4935	**0.4670**	**0.4646**	**0.4710**	**0.4712**	**0.4692**
	4	0.4605	0.4605	0.4850	0.4432	0.4486	0.4505	0.4459	**0.4394**	**0.4385**	**0.4399**	0.4485
d	2	0.5210	0.5244	0.5335	0.5221	0.5599	0.5382	0.5397	0.5329	0.5286	0.5314	0.5374
	4	0.4287	0.4283	0.4254	0.4183	0.4577	0.4503	0.4484	0.4403	0.4404	0.4401	0.4429
e	2	0.5100	0.5105	0.5040	0.5188	0.5305	0.5260	0.5186	0.5448	0.5362	**0.5104**	0.5354
	4	0.4552	0.4563	0.4642	0.4500	0.4454	0.4507	**0.4388**	**0.4396**	**0.4419**	**0.4412**	0.4588
f	2	0.5087	0.5086	0.5163	0.5136	0.5227	0.5123	**0.4984**	**0.4879**	**0.5063**	**0.5044**	**0.5031**
	4	0.4381	0.4381	0.4487	0.4433	0.4542	0.4521	**0.4380**	**0.4314**	**0.4378**	0.4475	**0.4345**
g	2	0.5251	0.5251	0.5250	0.5252	0.5257	0.5133	**0.5051**	**0.5018**	**0.4968**	**0.5102**	**0.5126**
	4	0.4324	0.4320	0.4324	0.4420	0.4536	0.4543	0.4429	0.4405	0.4448	0.4426	0.4401
h	2	0.5004	0.5004	0.4835	0.5194	0.5185	0.4854	0.5676	0.5657	0.5771	0.5827	0.5577
	4	0.4592	0.4586	0.4646	0.4363	0.4498	0.4522	0.4386	0.4427	0.4487	0.4416	0.4497
i	2	0.4773	0.4713	0.5191	0.5196	0.5341	0.5122	0.5416	0.5293	0.5247	0.5378	0.5443
	4	0.4556	0.4556	0.4578	0.4167	0.4530	0.4460	0.4454	0.4471	0.4422	0.4445	0.4476
j	2	0.5337	0.5072	0.5226	0.5134	0.5106	0.5097	**0.4980**	**0.4916**	**0.5023**	**0.5003**	**0.5045**
	4	0.4432	0.4424	0.4422	0.4432	0.4547	0.4524	0.4438	**0.4344**	**0.4330**	**0.4369**	**0.4327**
k	2	0.5703	0.5629	0.5690	0.5540	0.5517	0.5452	0.5668	0.5717	0.5633	0.5645	0.5557
	4	0.4812	0.4795	0.4851	0.4264	0.4489	0.4489	0.4449	0.4387	0.4377	0.4402	0.4470
l	2	0.4825	0.4754	0.4811	0.5128	0.4825	0.4874	0.5080	0.5159	0.5086	0.5369	0.5144
	4	0.4300	0.4291	0.4348	0.4394	0.4528	0.4518	0.4379	0.4396	0.4497	0.4485	0.4465
m	2	0.4925	0.4923	0.4880	0.4905	0.5526	0.4964	0.4886	0.4885	0.4948	**0.4860**	0.4917
	4	0.4476	0.4374	0.4324	0.4418	0.4459	0.4561	0.4445	0.4391	0.4418	0.4457	0.4397
n	2	0.4346	0.4222	0.4393	0.4672	0.5105	0.4746	0.5149	0.5110	0.5196	0.5111	0.5014
	4	0.4325	0.4324	0.4309	0.4318	0.4459	0.4561	0.4335	**0.4308**	**0.4306**	**0.4305**	0.4328
o	2	0.5460	0.5461	0.5565	0.5657	0.5672	0.5746	0.5681	0.5679	0.5677	0.5665	0.5771
	4	0.4641	0.4631	0.4507	0.4088	0.4517	0.4500	0.4428	0.4490	0.4410	0.4474	0.4391
p	2	0.5378	0.5368	0.5464	0.5317	0.5109	0.5087	0.5311	0.5273	0.5181	0.5207	0.5536
	4	0.4097	0.4082	0.4350	0.4535	0.4473	0.4445	0.4398	0.4475	0.4492	0.4464	0.4491
q	2	0.5402	0.5409	0.5373	0.5445	0.5394	0.5231	0.5413	0.5560	0.5391	0.5501	0.5570
	4	0.4050	0.4006	0.4358	0.4104	0.4515	0.4538	0.4418	0.4466	0.4409	0.4417	0.4483
r	2	0.4955	0.4955	0.4893	0.4838	0.5061	0.5034	0.5017	0.4997	0.4984	0.4969	0.4912
	4	0.4467	0.4488	0.4529	0.4325	0.4508	0.4552	0.4426	0.4475	0.4403	0.4412	0.4405
rand	2	0.5107	0.5311	0.4944	0.5027	0.4932	0.4968	0.5121	**0.4860**	0.4942	**0.4918**	0.4974
	4	0.4332	0.4332	0.4412	0.4382	0.4684	0.4671	0.4454	**0.4323**	0.4347	**0.4325**	0.4456

Table 3 Averages of Amari errors for $d = 2, d = 4$, and $n = 1{,}000$ sample with 120 replications for each distribution (a) to (r), the smallest (best) entry of our functions in each row is boldfaced

Distribution	*d*	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
a	2	0.4878	0.4879	0.4972	0.4941	0.5009	0.5027	**0.4737**	**0.4799**	**0.4814**	**0.4735**	**0.4780**
	4	0.4439	0.4439	0.4383	0.4354	0.4476	0.4441	0.4473	**0.4323**	**0.4305**	0.4365	0.4511
b	2	0.5090	0.5090	0.5038	0.5216	0.5066	0.5223	0.5575	0.5347	0.5485	0.5449	0.5374
	4	0.4400	0.4400	0.4178	0.4371	0.4515	0.4527	0.4539	0.4414	0.4366	0.4301	0.4548
c	2	0.5195	0.5195	0.5116	0.5151	0.5128	0.5053	**0.4986**	**0.4851**	**0.4938**	**0.4962**	**0.4938**
	4	0.4482	0.4482	0.4461	0.4396	0.4490	0.4483	0.4496	**0.4389**	**0.4367**	**0.4302**	0.4471
d	2	0.4639	0.4639	0.4655	0.4674	0.4656	0.4671	0.4699	0.4964	0.4726	0.4783	0.4897
	4	0.4558	0.4558	0.4442	0.4344	0.4534	0.4547	0.4475	0.4423	0.4430	0.4460	0.4488
e	2	0.5325	0.5446	0.5482	0.5476	0.5418	0.5408	0.5370	**0.5240**	0.5338	0.5355	0.5404
	4	0.4592	0.4592	0.4566	0.4372	0.4527	0.4482	0.4492	**0.4414**	**0.4370**	0.4449	0.4427
f	2	0.5261	0.5260	0.5268	0.5281	0.5280	0.5265	0.5535	0.5602	0.5614	**0.5200**	0.5679
	4	0.4556	0.4556	0.4407	0.4373	0.4493	0.4554	0.4502	**0.4367**	**0.4395**	0.4478	0.4475
g	2	0.5577	0.5577	0.5546	0.5508	0.5614	0.5619	**0.5495**	0.5555	0.5390	0.5520	**0.5492**
	4	0.5260	0.5224	0.5283	0.5063	0.4562	0.4504	**0.4500**	**0.4432**	**0.4455**	0.4551	**0.4494**
h	2	0.4639	0.4639	0.4616	0.4550	0.4594	0.4707	0.4770	0.4823	0.4771	0.4798	0.4704
	4	0.4384	0.4384	0.4270	0.4327	0.4536	0.4478	0.4523	0.4438	0.4399	0.4516	0.4462
i	2	0.4936	0.5028	0.5013	0.4981	0.5145	0.5313	0.5308	0.5246	0.5281	0.5069	0.5352
	4	0.5019	0.4986	0.4839	0.4321	0.4497	0.4491	0.4503	0.4372	0.4427	0.4510	0.4475
j	2	0.6040	0.5740	0.5317	0.5262	0.5279	0.5255	0.5319	0.5446	**0.5195**	**0.5108**	**0.5245**
	4	0.4634	0.4633	0.4482	0.4498	0.4607	0.4533	0.4495	**0.4443**	**0.4380**	0.4485	**0.4458**
k	2	0.5062	0.5062	0.5082	0.5127	0.4984	0.4952	0.5280	0.5336	0.5255	0.5263	0.5153
	4	0.4217	0.4248	0.4757	0.4269	0.4424	0.4404	0.4436	0.4342	0.4346	0.4439	0.4517
l	2	0.5033	0.5039	0.5169	0.5303	0.5387	0.5254	0.5525	0.5509	0.5529	0.5525	0.5455
	4	0.5096	0.5090	0.4950	0.4050	0.4503	0.4523	0.4537	0.4420	0.4386	0.4518	0.4476
m	2	0.4803	0.4803	0.4739	0.4844	0.5299	0.4723	0.4951	0.4767	0.4975	0.4880	**0.4713**
	4	0.4609	0.4609	0.4292	0.4468	0.4558	0.4482	0.4457	0.4385	0.4342	0.4468	0.4488
n	2	0.5230	0.5231	0.5019	0.4728	0.5292	0.4692	0.5085	0.4978	0.5025	0.4988	0.5072
	4	0.4821	0.4841	0.4441	0.4425	0.4465	0.4500	0.4499	**0.4368**	**0.4424**	0.4562	0.4523
o	2	0.4892	0.4892	0.4877	0.4858	0.5024	0.4714	0.4812	0.4744	0.4940	0.4845	0.4948
	4	0.4578	0.4578	0.4325	0.4363	0.4479	0.4489	0.4538	0.4488	0.4386	0.4513	0.4557
p	2	0.4982	0.4982	0.5147	0.4993	0.5194	0.4931	0.5330	0.5292	0.5255	0.5239	0.5278
	4	0.4747	0.4742	0.4304	0.4327	0.4523	0.4547	0.4522	0.4393	0.4391	0.4472	0.4481
q	2	0.5603	0.5450	0.5197	0.4844	0.4983	0.5190	0.5627	0.5642	0.5419	0.5553	0.5532
	4	0.4337	0.4337	0.4139	0.4328	0.4493	0.4558	0.4506	0.4423	0.4389	0.4515	0.4489
r	2	0.4978	0.4978	0.4965	0.4741	0.4868	0.4918	0.5075	0.4949	0.5098	0.5118	0.4932
	4	0.4695	0.4694	0.4447	0.4352	0.4491	0.4527	0.4502	**0.4371**	**0.4369**	0.4529	0.4462
rand	2	0.5091	0.5185	0.5290	0.5354	0.5170	0.5351	0.5147	**0.5061**	**0.5058**	0.5163	0.5104
	4	0.4494	0.4522	0.4329	0.4328	0.4643	0.4616	0.4463	**0.4325**	0.4334	0.4425	0.4408

Table 4 Averages of Amari errors for $d = 2, d = 4$, and $n = 1{,}500$ sample with 120 replications for each distribution (a) to (r), the smallest (best) entry of our functions in each row is boldfaced

Distribution	*d*	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
a	2	0.5204	0.5203	0.5266	0.5155	0.5285	0.5242	0.5586	0.5472	0.5488	0.5487	0.5376
	4	0.4436	0.4437	0.4491	0.4552	0.4530	0.4555	0.4454	0.4489	**0.4432**	0.4455	0.4465
b	2	0.5606	0.5606	0.5708	0.5533	0.5689	0.5590	0.5413	0.5428	0.5359	0.5487	0.5297
	4	0.4405	0.4436	0.4352	0.4452	0.4458	0.4456	0.4408	0.4420	0.4430	0.4393	0.4439
c	2	0.5247	0.5247	0.5244	0.5018	0.5231	0.5258	**0.5212**	**0.5327**	**0.5195**	**0.5163**	**0.5204**
	4	0.4438	0.4437	0.4519	0.4524	0.4532	0.4525	0.4482	**0.4412**	**0.4404**	**0.4405**	**0.4396**
d	2	0.5492	0.5491	0.5554	0.5674	0.5614	0.5606	0.5557	0.5599	0.5570	0.5676	0.5495
	4	0.4444	0.4569	0.4538	0.4482	0.4589	0.4571	0.4515	0.4501	0.4519	0.4520	0.4540
e	2	0.5025	0.5025	0.5020	0.5090	0.5092	0.5071	0.5050	0.5174	0.5012	0.4911	0.5126
	4	0.4423	0.4506	0.4459	0.4525	0.4563	0.4578	0.4533	0.4517	0.4547	0.4531	0.4498
f	2	0.5004	0.5004	0.5001	0.4964	0.5008	0.5017	0.5037	0.5097	**0.5060**	**0.5208**	0.5140
	4	0.4528	0.4534	0.4489	0.4554	0.4536	0.4587	**0.4447**	**0.4473**	**0.4452**	0.4497	**0.4488**
g	2	0.5678	0.5678	0.5682	0.5662	0.5662	0.5632	0.5373	0.5523	0.5475	0.5686	0.5394
	4	0.4345	0.4344	0.4534	0.4589	0.4559	0.4456	0.4570	0.4467	0.4502	0.4463	0.4511
h	2	0.5424	0.5424	0.5415	0.5380	0.5368	0.5279	**0.5274**	**0.5225**	**0.5092**	0.5254	**0.5436**
	4	0.5396	0.4284	0.4547	0.4548	0.4502	0.4491	0.4494	0.4469	0.4487	0.4507	0.4491
i	2	0.5056	0.5055	0.5134	0.4889	0.4891	0.5041	**0.4824**	**0.4820**	**0.4879**	**0.4939**	0.4758
	4	0.4470	0.4496	0.4580	0.4563	0.4560	0.4557	0.4485	**0.4459**	**0.4440**	**0.4453**	**0.4435**
j	2	0.5594	0.5524	0.5459	0.5457	0.5492	0.5454	**0.5497**	**0.5455**	**0.5522**	0.5387	**0.5536**
	4	0.4539	0.4538	0.4513	0.4566	0.4568	0.4480	0.4496	**0.4463**	**0.4476**	**0.4437**	**0.4417**
k	2	0.5080	0.5081	0.5026	0.5129	0.5039	0.5127	0.5344	**0.5345**	0.5309	**0.5181**	0.5351
	4	0.4399	0.4387	0.4551	0.4589	0.4523	0.4451	0.4519	0.4497	0.4481	0.4526	0.4470
l	2	0.5337	0.5338	0.5268	0.5216	0.5443	0.5404	0.5312	0.5411	0.5375	0.5394	0.5347
	4	0.4451	0.4348	0.4572	0.4514	0.4527	0.4525	0.4525	0.4505	0.4539	0.4572	0.4506
m	2	0.5272	0.5273	0.5143	0.5189	0.5430	0.5181	0.5220	0.5171	0.5181	0.5139	0.5236
	4	0.4440	0.4439	0.4543	0.4570	0.4558	0.4526	0.4549	**0.4419**	**0.4435**	**0.4425**	0.4451
n	2	0.5060	0.5066	0.5193	0.4954	0.5404	0.5200	0.5257	0.5265	0.5306	**0.5176**	**0.5233**
	4	0.4451	0.4427	0.4536	0.4517	0.4498	0.4545	0.4504	**0.4407**	**0.4426**	**0.4384**	**0.4359**
o	2	0.4441	0.4440	0.4408	0.4561	0.4409	0.4358	0.4629	0.4507	0.4495	0.4508	0.4623
	4	0.4540	0.4549	0.4468	0.4523	0.4600	0.4539	0.4515	0.4468	0.4531	0.4544	0.4522
p	2	0.4999	0.4999	0.5069	0.5257	0.5088	0.5061	0.5059	0.5034	0.5052	0.5062	0.5047
	4	0.4403	0.4492	0.4598	0.4542	0.4535	0.4589	0.4435	0.4404	0.4486	0.4434	0.4560
q	2	0.4880	0.4881	0.4826	0.4929	0.4839	0.4927	0.5144	0.5145	0.5109	0.4981	0.5151
	4	0.4439	0.4436	0.4534	0.4553	0.4512	0.4548	0.4532	0.4566	0.4573	0.4489	0.4533
r	2	0.5360	0.5352	0.5545	0.5510	0.5557	0.5524	0.5305	0.5058	0.5335	0.5380	0.5201
	4	0.4389	0.4348	0.4538	0.4577	0.4554	0.4488	0.4475	0.4479	0.4463	0.4514	0.4518
rand	2	0.5360	0.5352	0.5545	0.5510	0.5557	0.5524	0.5355	**0.5058**	**0.5335**	**0.5201**	0.5380
	4	0.4455	0.4435	0.4440	0.4556	0.4468	0.4439	0.4437	**0.4425**	**0.4429**	**0.4429**	0.4444

Table 5 Averages of Amari errors for $d = 2$, $d = 4$ and $n = 250$, $n = 500$, $n = 1{,}000$, $n = 1{,}500$ samples in each class of distributions, the smallest entry of our functions in each row is boldfaced

Class	*d*	*n*	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
symmetric	2	250	0.5006	0.5028	0.5067	0.5128	0.5154	0.5097	0.5084	0.5104	0.5064	0.5064	0.5065
		500	0.5053	0.5039	0.5089	0.5178	0.5284	0.5145	0.5178	0.5155	0.5161	0.5188	0.5183
		1000	0.5013	0.5021	0.4987	0.4976	0.5101	0.5001	0.5087	0.5061	0.5087	0.5021	0.5086
		1500	0.5226	0.5226	0.5250	0.5180	0.5272	0.5219	0.5217	0.5221	0.5191	0.5248	0.5199
	4	250	0.4401	0.4382	0.4506	0.4334	0.4498	0.4510	0.4388	0.4370	0.4352	0.4368	0.4381
		500	0.4451	0.4440	0.4507	0.4362	0.4510	0.4524	0.4417	0.4399	0.4413	0.4415	0.4420
		1000	0.4646	0.4642	0.4484	0.4437	0.4510	0.4500	0.4500	**0.4400**	0.4439	0.4457	0.4499
		1500	0.4536	0.4545	0.4508	0.4534	0.4538	0.4528	0.4493	0.4453	0.4460	0.4459	0.4463
non-symmetric	2	250	0.5055	0.5039	0.5204	0.5109	0.5106	0.5051	**0.5031**	**0.5009**	0.5150	0.5018	0.5119
		500	0.5166	0.5111	0.5135	0.5175	0.5133	0.5097	0.5165	0.5226	0.5171	0.5192	0.5260
		1000	0.5327	0.5273	0.5213	0.5103	0.5188	0.5159	0.5374	0.5346	0.5306	0.5316	0.5308
		1500	0.5199	0.5187	0.5198	0.5243	0.5252	0.5240	**0.5177**	0.5213	0.5234	**0.5186**	0.5235
	4	250	0.4428	0.4436	0.4427	0.4432	0.4510	0.4506	**0.4395**	**0.4392**	0.4437	**0.4379**	**0.4393**
		500	0.4439	0.4438	0.4500	0.4436	0.4502	0.4510	**0.4414**	**0.4420**	0.4481	**0.4423**	0.4461
		1000	0.4617	0.4619	0.4521	0.4431	0.4510	0.4511	**0.4430**	**0.4401**	0.4438	0.4487	0.4473
		1500	0.4543	0.4544	0.4537	0.4552	0.4540	0.4502	0.4502	**0.4490**	0.4509	**0.4500**	0.4503

Table 5 Averages of Amari errors for $d = 2$, $d = 4$ and $n = 250$, $n = 500$, $n = 1{,}000$, $n = 1{,}500$ samples in each class of distributions, the smallest entry of our functions in each row is boldfaced (continued)

Class	d	n	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
unimodal	2	250	0.4969	0.5005	0.5092	0.5158	0.5183	0.5103	0.5278	0.5241	0.5243	0.5171	0.5261
		500	0.5115	0.5103	0.5159	0.5238	0.5255	0.5251	0.5228	0.5264	0.5194	0.5232	0.5259
		1000	0.4971	0.4999	0.5021	0.5024	0.5072	0.5066	0.5138	0.5100	0.5151	0.5110	0.5143
		1500	0.5190	0.5189	0.5238	0.5204	0.5248	0.5230	0.5210	0.5184	0.5189	0.5223	0.5153
	4	250	0.4396	0.4384	0.4516	0.4375	0.4502	0.4518	0.4405	0.4380	**0.4362**	0.4399	0.4387
		500	0.4447	0.4449	0.4520	0.4352	0.4515	0.4516	0.4418	0.4427	0.4430	0.4424	0.4453
		1000	0.4672	0.4667	0.4516	0.4316	0.4503	0.4503	0.4507	0.4403	0.4380	0.4456	0.4493
		1500	0.4445	0.4461	0.4499	0.4523	0.4547	0.4533	0.4489	0.4480	0.4488	0.4498	0.4490
multimodal	2	250	0.5054	0.5042	0.5108	0.5021	0.5131	0.5130	0.5098	0.5153	0.5096	0.5065	0.5092
		500	0.5077	0.5039	0.5043	0.5180	0.5236	0.5044	0.5315	0.5338	0.5340	0.5359	0.5282
		1000	0.5146	0.5115	0.5006	0.4880	0.4996	0.4919	0.5150	0.5126	0.5082	0.5113	0.5080
		1500	0.5138	0.5140	0.5141	0.5082	0.5176	0.5158	0.5246	0.5261	0.5202	0.5151	0.5275
	4	250	0.4456	0.4421	0.4444	0.4375	0.4490	0.4498	0.4387	0.4384	0.4396	0.4380	0.4394
		500	0.4342	0.4316	0.4381	0.4448	0.4511	0.4519	0.4418	0.4439	0.4413	0.4438	0.4439
		1000	0.4561	0.4553	0.4354	0.4346	0.4349	0.4324	0.4495	0.4404	0.4439	0.4491	0.4479
		1500	0.4451	0.4470	0.4535	0.4564	0.4551	0.4527	0.4499	0.4455	0.4470	0.4453	0.4485
transitional	2	250	0.5096	0.5051	0.5196	0.5055	0.5103	0.5069	**0.4840**	**0.4840**	**0.4879**	**0.4837**	**0.4848**
		500	0.5196	0.5140	0.5197	0.5149	0.5245	0.5081	**0.5042**	**0.4994**	**0.5037**	**0.5043**	**0.5131**
		1000	0.5333	0.5272	0.5203	0.5178	0.5333	0.5159	0.5326	0.5332	0.5286	0.5189	0.5281
		1500	0.5309	0.5296	0.5271	0.5306	0.5336	0.5269	**0.5237**	**0.5256**	**0.5258**	0.5296	0.5271
	4	250	0.4439	0.4439	0.4442	0.4392	0.4516	0.4503	**0.4372**	**0.4368**	**0.4359**	**0.4372**	**0.4379**
		500	0.4477	0.4463	0.4603	0.4393	0.4489	0.4523	0.4409	0.4396	**0.4390**	**0.4388**	0.4453
		1000	0.4448	0.4458	0.4414	0.4349	0.4482	0.4548	0.4492	0.4392	**0.4343**	**0.4347**	0.4492
		1500	0.4562	0.4544	0.4537	0.4546	0.4513	0.4512	**0.4506**	**0.4470**	**0.4474**	**0.4462**	**0.4450**

Table 5 Averages of Amari errors for $d = 2$, $d = 4$ and $n = 250$, $n = 500$, $n = 1{,}000$, $n = 1{,}500$ samples in each class of distributions, the smallest entry of our functions in each row is boldfaced (continued)

Class	d	n	*FastICA*	*Infomax*	*JADE*	*RADICAL*	*HICA*	*RLICA*	$GDCICA_{g_1}$	$GDCICA_{g_2}$	$GDCICA_{g_3}$	$GDCICA_{g_4}$	$GDCICA_{g_5}$
kurtoses +	2	250	0.5018	0.5050	0.5134	0.5169	0.5183	0.5164	**0.5009**	0.5046	0.5080	**0.5011**	0.5054
		500	0.5198	0.5206	0.5196	0.5245	0.5273	0.5271	**0.5104**	**0.5123**	**0.5173**	**0.5124**	**0.5166**
		1000	0.5039	0.5063	0.5083	0.5118	0.5086	0.5119	0.5104	0.5183	0.5190	0.5195	0.5227
		1500	0.5299	0.5301	0.5310	0.5283	0.5338	0.5305	0.5354	0.5329	0.5354	**0.8279**	**0.5281**
	4	250	0.4413	0.4412	0.4575	0.4438	0.4497	0.4500	**0.4380**	0.4388	**0.4348**	**0.4366**	**0.4378**
		500	0.4440	0.4401	0.4536	0.4455	0.4515	0.4524	0.4407	**0.4380**	**0.4354**	0.4410	0.4446
		1000	0.4509	0.4509	0.4440	0.4436	0.4509	0.4510	0.4496	**0.4388**	**0.4373**	0.4411	0.4490
		1500	0.4480	0.4496	0.4477	0.4513	0.4535	0.4549	**0.4471**	0.4480	**0.4476**	**0.4472**	0.4486
kurtoses −	2	250	0.5032	0.5019	0.5122	0.5062	0.5132	0.5076	0.5064	0.5130	0.5115	0.5136	0.5117
		500	0.5099	0.5053	0.5115	0.5179	0.5237	0.5098	0.5258	0.5231	0.5224	0.5217	0.5254
		1000	0.5152	0.5124	0.5062	0.4992	0.5138	0.5025	0.5144	0.5197	0.5164	0.5159	0.5140
		1500	0.5187	0.5181	0.5186	0.5165	0.5219	0.5188	0.5173	0.5188	0.5176	0.5179	0.5194
	4	250	0.4411	0.4357	0.4437	0.4306	0.4505	0.4511	0.4394	0.4374	0.4364	0.4375	0.4389
		500	0.4437	0.4422	0.4492	0.4328	0.4503	0.4517	0.4419	0.4418	0.4414	0.4421	0.4432
		1000	0.4683	0.4681	0.4538	0.4399	0.4510	0.4501	0.4501	0.4405	**0.4389**	0.4491	0.4489
		1500	0.4515	0.4425	0.4541	0.4552	0.4540	0.4517	0.4506	0.4463	0.4480	0.4473	0.4475

Certainly, the average Amari errors were computed for performance evaluation across all methods in various distributions, categorised into different classes. Initially, the classifications were divided into 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) for the first analysis. Subsequently, the perspective shifted to symmetric distributions (a, b, c, d, f, g, h, i, m, n, o) and non-symmetric distributions (e, j, k, l, p, q, r). Finally, the third perspective involved positively kurtosis distributions (a, b, d, e, f) and negatively kurtosis distributions (c, g, h, i, j, k, l, m, n, o, p, q, r). Each of these classes formed a partition within the set comprising all distributions. The detailed results of this analysis are presented in Table 5.

According to Table 5, the optimal performance in terms of Amari errors was observed in two specific scenarios:

- For $d = 2$, the GDCICA$_{g_3}$ and GDCICA$_{g_5}$ methods have demonstrated superior results in the transitional distributions and positively kurtosis class. Also for $d = 4$, the GDCICA$_{g_3}$ method has demonstrated superior results in the transitional distributions and positively kurtosis class.
- For $d = 2, 4$, the GDCICA$_{g_1}$, GDCICA$_{g_2}$ and GDCICA$_{g_4}$ methods have demonstrated superior results in the non-symmetric distributions, transitional distributions and positively kurtosis class.

Overall, there was a consistent trend of achieving commendable results in terms of Amari errors across transitional distributions and the positively kurtosis class, indicating a notably favourable performance level.

6 Real data

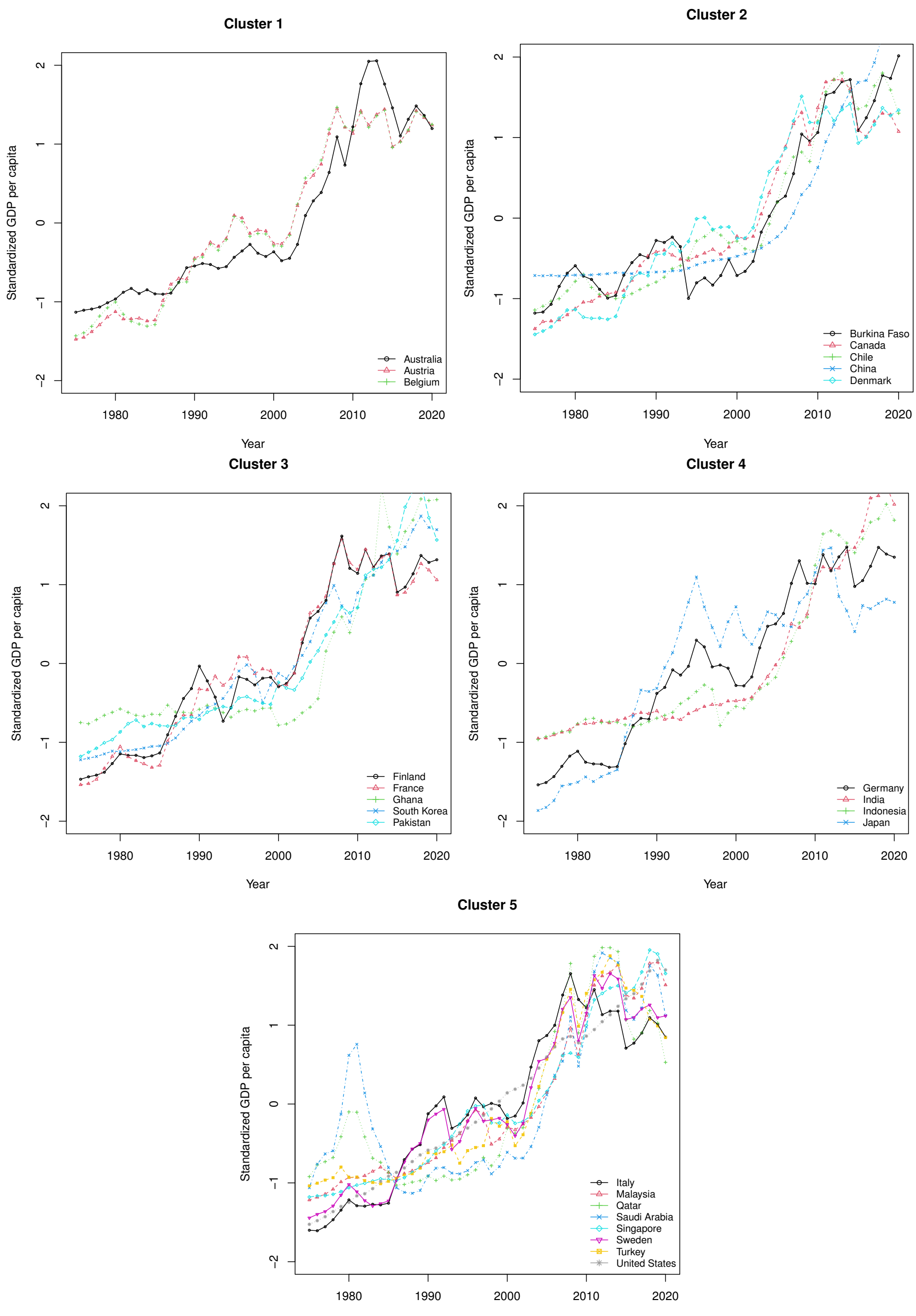
The application of ICA within statistics involves transforming raw time series data into a set of independent variable values. Its utility extends to cluster analysis, aiding in the identification of datasets exhibiting similar temporal patterns. Moreover, ICA serves as an effective data pre-processing step, enhancing temporal data clustering.

ICA's significance lies in its ability to facilitate clustering for small, distribution-free data series while considering dependencies between successive time points (Nascimento et al., 2017). By inherently addressing temporal dependencies during variable decomposition, ICA swiftly generates independent components. These components are subsequently grouped based on the temporal relationships among them (Calhoun and Adali, 2012). This inherent capacity to account for temporal dependency within its model distinguishes ICA's effectiveness in swiftly generating and clustering independent components, bolstering its relevance in diverse statistical applications.

Based on Guo et al. (2008), Zanghaei et al. (2013) and Rahmanishamsi et al. (2018), 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.

Absolutely, GDCICA algorithms offer substantial value in the realm of time series data pre-processing. Their efficacy lies in identifying homogeneous structures within the data, thereby maximising within-group similarities while emphasising distinctions between different groups. This approach enhances the effectiveness of time series clustering, rendering it an attractive and potent method for extracting valuable insights across diverse domains.

Figure 3 Trend plots of standardised GDP per capita time series in five clusters (see online version for colours)



ICA's applications extend beyond time series clustering; it provides valuable insights into common characteristics of datasets. This method involves a multi-stage process utilising the GDCICA algorithm as a pre-processing step in time series clustering when applied to real data. Statistically independent components can be obtained through GDCICA, leveraging dependency implications to reduce data dimensionality. This method effectively detects dependencies within time series data. By generating new components from sources containing ample information about time series trends, this pre-processing technique enhances data representation. Consequently, the coefficients of the mixing matrix derived from GDCICA can serve as input variables in clustering methods.

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 the their dependencies are converted to the weight matrix $\hat{A}$. On the other hand, i-th column of X is constructed as 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 $\hat{A}$.

Table 6 Clustering of standardised GDP per capita time series.

Method	*Suggested number of clusters*	*Average silhouette*	*Method*	*Suggested number of clusters*	*Average silhouette*
GDCICA$_{g_1}$	5	0.092	JADE	7	0.069
GDCICA$_{g_2}$	3	0.118	Infomax	6	0.049
GDCICA$_{g_3}$	5	0.129	FastICA	8	0.059
GDCICA$_{g_4}$	3	0.104	RADICAL	6	0.055
GDCICA$_{g_5}$	3	0.110	HICA	7	0.080
			RLICA	4	0.090

Then, the GDP per capita time series data for 25 countries: Australia, Austria, Belgium, Burkina Faso, Canada, Chile, China, Denmark, Finland, France, Germany, Ghana, India, Indonesia, Italy, Japan, South Korea, Malaysia, Pakistan, Qatar, Saudi Arabia, Singapore, Sweden, Turkey, and the USA, was sourced from the World Bank's repository (`https://data.worldbank.org`) spanning the period 1975 to 2020.

In this study, time series clustering employed suggested algorithms as a pre-processing step. Subsequently, the `PAM` algorithm (k-medoids), introduced by Rdusseeun and Kaufman (1987), was applied for final clustering.

Furthermore, after estimating the mixing matrix using multiple algorithms, including GDCICA with FastICA, Infomax, JADE, RADICAL, HICA and RLICA, the `PAM` clustering algorithm was employed on the unmixing matrix. This step aimed to determine the appropriate number of clusters and declare the best clustering method based on the silhouette criterion, utilising the `NbClust R` package (Rdusseeun and Kaufman, 1987). The detailed results are reported in Table 6. Based on the results, the function exhibiting the highest average silhouette score has been selected as the best-performing result.

This dataset was subjected to clustering using the GDCICA_{g_3} function. The primary objective of this experiment was to cluster countries based on the patterns observed in their GDP per capita time series spanning the past 46 years. To the study objective satisfaction, the primary GDP per capita data has standardised. Then, the GDCICA algorithm has applied on the principal components and the coefficients of the mixing matrix in GDP per capita. The extracted data has utilised as input to PAM clustering algorithm. The clustering results illustrated in Table 7.

Table 7 Clustering of standardised GDP per capita time series.

Cluster 1	Australia, Austria, Belgium.
Cluster 2	Burkina Faso, Canada, Chile, China, Denmark.
Cluster 3	Finland, France, Ghana, South Korea, Pakistan.
Cluster 4	Germany, India, Indonesia, Japan.
Cluster 5	Italy, Malaysia, Qatar, Saudi Arabia, Singapore, Sweden, Turkey, USA.

Although these time series data adhere to various distributions, the proposed generalised density contrast independent component analysis (GDCICA) algorithms are capable of effectively extracting the sources necessary for clustering. To evaluate the method's efficiency, time series plots within five distinct clusters have been generated, as depicted in Figure 3. The trend plots derived from all clusters, based on the standardised gross domestic product (GDP) per capita, indicate that the preprocessing technique employed by the proposed algorithm for clustering countries is highly effective. This is evidenced by the clear observation that countries within the same cluster exhibit similar trends in their standardised GDP per capita.

7 Conclusions

The objective function of many common ICA algorithms is defined based on density functions, which estimate marginal and joint density functions. This can lead to longer and time-consuming algorithm execution. As a result, this is a challenging problem. In this study, a class of dependence criteria of two continues random variables X_1 and X_2 based on the cumulative distribution function, called GDC has proposed to solve the challenging problem. The criterion of this class remains unchanged despite increasing transformations of random variables, and it only disappears when the random variables X_1 and X_2 are independent.

Then, some algorithms for ICA problem based on this dependence criterion has presented called GDCICA. To suggested algorithms performance evaluation, these has compared to some exiting algorithms using by the Monte-Carlo simulation studies, when the source signals follow the 18 different distributions. The simulation results argued that the suggested algorithms estimate the unmixing matrix very well. So, it obtained better performance than the usual ICA algorithms in various classes of distributions bas on average of the Amari errors. Usually, various types of GDCICA_g have good performance in term of the average of the Amari errors in the transitional distributions and positively kurtosis class.

So, one batch of GDP per capita for 25 countries during 1975 to 2020 has utilised as time series samples to clustering. At first, all data standardised. Then, the GDCICA has utilised on principal components as a pre-processing stage. As a result of this stage, the coefficients of the matrix have extracted. Then, they were applied as input to the PAM clustering algorithm. The clustering results concluded from a time series data argued which the pre-processing technique using proposed algorithm are fantastic in results of suitable clustering of different data that follow different distributions.

Acknowledgements

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.

References

Amari, S., Cichocki, A. and Yang, H. (1996) 'A new learning algorithm for blind source separation', *Advances in Neural Information Processing Systems*, Vol. 8, pp.757–763.

Anbumalar, S., Rameshbabu, P. and Anandanatarajan, R. (2011) 'Overlapped chromatograms separation using nonnegative matrix factorization', *International Journal of Computer and Electrical Engineering*, Vol. 3, No. 5, pp.654–658.

Antony, M.J., Sankaralingam, B.P., Mahendran, R.K., Gardezi, A.A., Shafiq, M., Choi, J.G. and Hamam, H. (2022) 'Classification of EEG using adaptive SVM classifier with CSP and online recursive independent component analysis', *Sensors (Basel, Switzerland)*, Vol. 22, No. 19, p.7596.

Aziz, R., Verma, C. and Srivastava, N. (2016) 'A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data', *Genomics Data*, Vol. 8, pp.4–15.

Bach, F.R. and Jordan, M.I. (2002) 'Kernel independent component analysis', *Journal of Machine Learning Research*, July, Vol. 3, pp.1–48.

Bingham, E. and Hyvärinen, A. (2000) 'A fast fixed-point algorithm for independent component analysis of complex valued signals', *International Journal of Neural Systems*, Vol. 10, No. 1, pp.1–8.

Calhoun, V.D. and Adali, T. (2012) 'Multisubject independent component analysis of fMRI: a decade of intrinsic networks, default mode, and neurodiagnostic discovery', *IEEE Reviews in Biomedical Engineering*, Vol. 5, pp.60–73.

Cardoso, J.F. (1999) 'High-order contrasts for independent component analysis', *Neural Computation*, Vol. 11, No. 1, pp.157–192.

Cho, J.W. and Park, H.M. (2016) 'Independent vector analysis followed by HMM-based feature enhancement for robust speech recognition', *Signal Processing*, Vol. 120, pp.200–208.

Comon, P. (1994) 'Independent component analysis, a new concept?', *Signal Processing*, Vol. 36, No. 3, pp.287–314.

Comon, P. and Jutten, C. (2010) *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, USA.

Du, K.L. and Swamy, M.N.S. (2014) *Neural Networks and Statistical Learning*, Springer, London.

Ferguson, T.S. (2017) *A Course in Large Sample Theory*, Routledge, New York.

Funaro, M., Oja, E. and Valpola, H. (2001) 'Artefact detection in astrophysical image data using independent component analysis', *Proceedings of 3rd International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, San Diego, USA, pp.43–48.

Gaeta, M. (1990) 'Source separation without prior knowledge: the maximum likelihood solution', *Proceeding. EUSIPCO'90*, pp.621–624.

Guo, C., Jia, H. and Zhang, N. (2008) 'Time series clustering based on ICA for stock data analysis.', *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, pp.1–4.

Hyvärinen, A. (1999) 'Fast and robust fixed-point algorithms for independent component analysis', *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, pp.626–634.

Hyvärinen A., Karhunen, J. and Oja, E. (2001) *Independent Component Analysis and Blind Source Separation*, pp.20–60, Wiley, New York, NY, USA.

Ilmonen, P., Nordhausen, K., Oja, H. and Ollila, E. (2010) 'A new performance index for ICA: properties, computation and asymptotic analysis', in Vigneron, V., Zarzoso, V., Moreau, E., Gribonval, R. and Vincent, E. (Eds.): *Latent Variable Analysis and Signal Separation: 9th International Conference*, Springer, pp.229–236.

Ismail, I.A., Ramadan, M.A. and Samak, A.H. (2009) 'An effective iris recognition system using Fourier descriptor and principle component analysis', *International Journal of Computer and Electrical Engineering*, Vol. 1, No. 2, pp.117–120.

Jayabal, V., Badier, J.M., Pizzo, F., Villalon, S.M., Papageorgakis, C., López-Madrona, V., … and 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*, Vol. 264, pp.1–17.

Kaur, M. and Vashisht, R. (2011) 'Facial expression recognition using a noval approach and its application', *International Journal of Computer and Electrical Engineering*, Vol. 3, No. 2, pp.254–258.

Langlois, D., Chartier, S. and Gosselin, D. (2010) 'An introduction to independent component analysis: Infomax and FASTICA algorithms', *Tutorials in Quantitative Methods for Psychology*, Vol. 6, No. 1, pp.31–38.

Lassance, N., DeMiguel, V. and Vrins, F. (2022) 'Optimal portfolio diversification via independent component analysis', *Operations Research*, Vol. 70, No. 1, pp.55–72.

Learned-Miller, E.G. and Fisher III, J.W. (2003) 'ICA using spacings estimates of entropy', *Journal of Machine Learning Research*, Vol. 4, pp.1271–1295.

Lee, T.W., Girolami, M. and Sejnowski, T.J. (1999) 'Independent component analysis using an extended infomax algorithm for mixed subGaussian and superGaussian sources', *Neural Computation*, Vol. 11, No. 2, pp.417–441.

Lipshutz, D., Pehlevan, C. and Chklovskii, D.B. (2022) 'Biologically plausible single-layer networks for nonnegative independent component analysis', *Biological Cybernetics*, Vol. 116, No. 5, pp.557–568.

Lyu, Q. and Fu, X. (2022) 'On finite-sample identifiability of contrastive learning-based nonlinear independent component analysis', *Proceedings of the 39th International Conference on Machine Learning*, PMLR, Baltimore, Maryland, USA, Vol. 162, pp.14582–14600.

Meng, X., Iraji, A., Fu, Z., Kochunov, P., Belger, A., Ford, J., … and 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*, Vol. 12, No. 7, pp.617–628.

Moneta, A. and Pallante, G. (2022) 'Identification of structural var models via independent component analysis: a performance evaluation study', *Journal of Economic Dynamics and Control*, Vol. 144, pp.1–37.

Moneta, A., Entner, D., Hoyer, P.O. and Coad, A. (2010) 'Causal inference by independent component analysis with application to micro and macroeconomic data', *Jena Economic Research Papers*, Vol. 31, pp.1–32.

Naik, G. R. and Kumar, D.K. (2011) 'An overview of independent component analysis and its application', *Informatica*, Vol. 35, No. 1, pp.63–81.

Nascimento, M., Silva, F.F.E., Safadi, T., Nascimento, A.C.C., Ferreira, T.E.M., Barroso, L.M.A., … and Serão, N.V.L. (2017) 'Independent component analysis (ICA) based-clustering of temporal RNA-seq data.*PLoS ONE*, Vol. 12, No. 7, pp.1–12.

Nordhausen, K. and Oja, H. (2018) 'Independent component analysis: statistical perspective', *A Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 10, No. 5, pp.757–763.

Pearlmutter, B.A, and Parra, L.C. (1996) 'Maximum likelihood blind source separation: a context-sensitive generalization of ICA', *Advances in Neural Information Processing Systems*, Vol. 9, pp.613–619.

Pfister, N., Weichwald, S., Bühlmann, P. and Schölkopf, B. (2019) 'Robustifying independent component analysis by adjusting for group-wise stationary noise', *Journal of Machine Learning Research*, Vol. 20, No. 147, pp.1–50.

Pham, D.T, Garrat, P. and Jutten, C. (1992) 'Separation of a mixture of independent sources through a maximum likelihood approach', *European Signal Processing Conference*, pp.771–774.

Pontifex, M.B., Gwizdala, K.L., Parks, A.C., Billinger, M. and Brunner, C. (2017) 'Variability of ICA decomposition may impact EEG signals when used to remove eye blink artifacts', *Psychophysiology*, Vol. 54, No. 3, pp.386–398.

Rahmanishamsi, J. and Dolati, A. (2018) 'Rank based least-squares independent component analysis', *Journal of Statisical Research of Iran*, Vol. 14, No. 2, pp.247–266.

Rahmanishamsi, J., Dolati, A. and Aghabozorgi, M.R. (2018) 'A copula based ICA algorithm and its application to time series clustering', *Journal of Classification*, Vol. 35, No. 2, pp.230–249.

Rahman, S., Naim, S.M., Farooq, A.A. and Islam, M. (2012) 'Combination of Gabor and curvelet texture features for face recognition using principal component analysis', *International Journal of Computer and Electrical Engineering*, Vol. 4, No. 3, pp.264–269.

Rdusseeun, L.K.P.J. and Kaufman, P. (1987) 'Clustering by means of medoids', *Proceedings of the Statistical Data Analysis Based on the L1 Norm Conference*, Neuchatel, Switzerland, Vol. 31, pp.405–416.

Resnick, S. (2019) *A Probability Path*, Springer, USA.

Rincourt, S.L., Michiels, S. and Drubay, D. (2022) 'Complex disease individual molecular characterization using infinite sparse graphical independent component analysis', *Cancer Informatics*, Vol. 21, pp.1–16.

Shahina, K. and Kumar, T.S.P. (2022) 'Similarity-based clustering and data aggregation with independent component analysis in wireless sensor networks', *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 7, pp.1–15.

Shang, L., Zhang, Y. and Sun, Z.L. (2022) 'Palmprint feature extraction utilizing WTA-ICA in contourlet domain', *International Conference on Intelligent Computing*, Xi'an, China, pp.464–471.

Sheikh, M.S. and 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*, Vol. 586, p.126504.

Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A. and Jordan, M. (2006) 'A linear non-Gaussian acyclic model for causal discovery', *Journal of Machine Learning Research*, Vol. 7, No. 10, pp.2003–2030.

Sivaramakrishna, M. (2011) 'Simulation to establish correlation between PCA and reactor power using neutron detector signals', *International Journal of Computer and Electrical Engineering*, Vol. 3, No. 4, pp.489–492.

Sompairac, N., Nazarov, P.V., Czerwinska, U., Cantini, L., Biton, A., Molkenov, A. and Zinovyev, A. (2019) 'Independent component analysis for unraveling the complexity of cancer omics datasets', *International Journal of Molecular Sciences*, Vol. 20, No. 18, p.4414.

Stone, J.V. (2004) *Independent Component Analysis: A Tutorial Introduction*, The MIT Press, USA.

Tabanfar, Z., Ghassemi, F. and 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*, Vol. 71, p.103162.

Tan, L. and Zhang, H. (2012) 'Forecast of employment based on independent component analysis', in Liu, C., Wang, L. and Yang, A. (Eds.): *Information Computing and Applications. ICICA 2012. Communications in Computer and Information Science*, Springer, Berlin, Vol. 307.

Tharwat, A. (2021) 'Independent component analysis: an introduction', *Applied Computing and Informatics*, Vol. 17, No. 2, pp.222–249.

Xie, J., Douglas, P. K., Wu, Y.N., Brody, A.L. and Anderson, A.E. (2017) 'Decoding the encoding of functional brain networks: an fMRI classification comparison of non-negative matrix factorization (NMF), independent component analysis (ICA), and sparse coding algorithms', *Journal of Neuroscience Methods*, Vol. 282, pp.81–94.

Zanghaei, A., Abolhassani, M., Ahmadian, A., Ay, M.R. and Saberi, H. (2013) 'A new method to enhance the clustering algorithm', *International Journal of Computer and Electrical Engineering*, Vol. 5, No. 1, pp.120–122.

Zhou, R., Han, J., Li, T. and 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*, Vol. 60, pp.1–15.