Filomat 39:24 (2025), 8551–8566
https://doi.org/10.2298/FIL2524551P

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Mahalanobis Distance-Based Modification of the *RENES* Method for Environment State Estimation in Generalized RrINAR Models of Higher Order

Bogdan A. Pirković[a,*], Milena S. Stojanović[b], Milena P. Živković[a]

[a] *University of Kragujevac, Faculty of Science, Radoja Domanovića 12, 34000 Kragujevac, Serbia*
[b] *University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia*

Abstract. The dynamic of integer-valued autoregressive model in random environment is governed by the realization $\{z_n\}_{n=1}^{\infty}$ of a Markov chain referred to as the random environment process. At given moment $n \in \mathbb{N}$, the realization z_n defines the environment conditions and determines all model parameters at that moment. In most cases, the K-means clustering technique has been used to estimate $\{z_n\}_{n=1}^{\infty}$, which is a necessary step in models application. However, the application of the K-means technique is not always the optimal solution, as it disregards certain information and may yield suboptimal results in some scenarios. To enhance clustering performance for data sequences corresponding to generalized random environment integer-valued autoregressive time series of higher order, the so-called *RENES* clustering method was developed. Despite its advantages, *RENES* also has some drawbacks and is highly complex to implement. To address these challenges, we propose a modification of the *RENES* method based on the Mahalanobis distance, designed to simplify the algorithm and improve its practical applicability while preserving clustering accuracy. The effectiveness of this modification was evaluated using the same simulations and real-life data where the *RENES* method had previously demonstrated its validity, and notable improvements were observed.

1. Introduction

Integer-valued autoregressive (*INAR*) models, first introduced in [6] and [1], have proven to be highly effective in modeling integer-valued data over time. These models rely on thinning operator, which transforms a given integer-valued random variable X to the sum of X independent and identically distributed random variables. Over the years, several significant contributions to this class of models have appeared in [5], [14], [15] and [11]. Despite their versatility, the behavior of such models can vary considerably

2020 *Mathematics Subject Classification*. Primary 62P99.
Keywords. Random environment, Mahalanobis distance, K-means, *INAR*, estimation, *RENES*
Received: 26 May 2025; Revised: 05 July 2025; Accepted: 17 July 2025
Communicated by Aleksandar Nastić
This work was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia through the Agreements No. 451-03-137/2025-03/ 200122, No. 451-03-136/2025-03/ 200122 and No. 451-03-137/2025-03/200124
* Corresponding author: Bogdan A. Pirković
Email addresses: bogdan.pirkovic@pmf.kg.ac.rs (Bogdan A. Pirković), milena_aleksic93@yahoo.com (Milena S. Stojanović), milena.zivkovic@pmf.kg.ac.rs (Milena P. Živković)
ORCID iDs: https://orcid.org/0000-0003-3546-2750 (Bogdan A. Pirković), https://orcid.org/0000-0003-2566-3436 (Milena S. Stojanović), https://orcid.org/0000-0001-8567-7050 (Milena P. Živković)

depending on the environment conditions in which they are observed. A significant advancement in this matter was made in [9], where authors presented the first *INAR* model in random environment, calling it the random environment *INAR* model of order 1 with geometric marginals (*RrNGINAR*(1)). To this end, the authors first introduced the r-state random environment process $\{Z_n\}$, $n \in \mathbb{N}_0$, defined as a Markov chain taking values in $E_r = \{1, 2, \ldots, r\}$. Further, the marginal distribution of the *RrNGINAR*(1) model at time n is governed by the realization of the random environment process z_n observed at the same time, which we denote as $X_n(z_n)$. Additionally, $X_n(z_n)$ follows a geometric distribution with expectation $\mu_{z_n} \in \{\mu_1, \mu_2, \ldots \mu_r\}$. The model itself is defined as

$$X_n(z_n) = \alpha * X_{n-1}(z_{n-1}) + \varepsilon_n(z_n, z_{n-1}),$$

where $\alpha \in (0, 1)$. The operator $\alpha* : X \mapsto \sum_{i=1}^{X} U_i$ is the negative binomial thinning operator, which maps integer-valued random variable X to the sum of X independent random variables, each following the same geometric distribution with distribution parameter $\frac{\alpha}{1+\alpha}$. Consequently, the distribution of $\alpha * X$ conditional on $X = x$ is negative binomial with parameters x and $\frac{\alpha}{1+\alpha}$.

Over the years, more advanced *INAR* models were developed. In [10], random environment *INAR* models of higher orders were introduced. In addition to the marginal distribution parameter, the authors proposed that the order of the model is influenced by the environment state at each moment $n \in \mathbb{N}$. A further significant advancement occurred in [4], where the authors extended previous assumptions by suggesting that the thinning parameter α_{z_n} at time n is also dependent on the environment state z_n at that same time. The corresponding model $\{X_n(z_n)\}_{n=1}^{\infty}$ is referred to as a generalized random environment *INAR* model of higher order with geometric marginals and negative binomial thinning operator (denoted as *RrNGINAR*$(\mathcal{M}, \mathcal{A}, \mathcal{P})$) and is given by the following recursive relation:

$$X_n(z_n) = \begin{cases} \alpha_{z_n} * X_{n-1}(z_{n-1}) + \varepsilon_n(z_n, z_{n-1}) & \text{w.p. } \phi^{z_n}_{1,P_n}, \\ \alpha_{z_n} * X_{n-2}(z_{n-2}) + \varepsilon_n(z_n, z_{n-2}) & \text{w.p. } \phi^{z_n}_{2,P_n}, \\ \vdots & \\ \alpha_{z_n} * X_{n-P_n}(z_{n-P_n}) + \varepsilon_n(z_n, z_{n-P_n}) & \text{w.p. } \phi^{z_n}_{P_n,P_n}. \end{cases} \tag{1}$$

Here, $\mathcal{M} = \{\mu_1, \ldots, \mu_r\}$, $\mathcal{A} = \{\alpha_1, \ldots, \alpha_r\}$, $\mathcal{P} = \{p_1, \ldots, p_r\}$ represent the model's parameter sets. μ_{z_n} denotes the mean of the marginal geometric distribution of $X_n(z_n)$, α_{z_n} is the thinning parameter and p_{z_n} indicates the maximum value the order P_n can take for a fixed state $z_n \in \{1, \ldots, r\}$. Depending on how the model behaves after a state change, two different versions of the *RrNGINAR*$(\mathcal{M}, \mathcal{A}, \mathcal{P})$ model can be identified. The first is denoted *RrNGINAR*$_{max}(\mathcal{M}, \mathcal{A}, \mathcal{P})$ and is created in the way that $P_n = \min\{p_n^*, p_{z_n}\}$ for each $n \in \mathbb{N}$. Thus, when the state change occurs, the order of the model resets to one. From that point onward, the order increases as long as the model remains in the same state, continuing until it reaches a predefined maximum value p_{z_n}. Once this maximum is attained, the order remains constant at that value until the next state change happens. The other version, referred to as *RrNGINAR*$_1(\mathcal{M}, \mathcal{A}, \mathcal{P})$, slightly modifies the way that orders reach their maximum values. Namely, here we have that $P_n = 1$ if $p_n^* < p_{z_n}$ and $P_n = p_{z_n}$ otherwise. In other words, when the state change occurs and the order of the model resets to one, it does not necessarily increase to the maximum order. Instead, the model order remains at one until the conditions that permit its progression to the maximum value within the given state are satisfied. In both versions, $p_n^* = \max\{i \geq 1 : z_{n-i} = \cdots = z_{n-1}\}$ represents the number of predecessors of z_n that are mutually equal.

The following assumptions must be satisfied for both versions of the model.

a) For all $i \in E_r$, $p \in \{1, 2, \ldots, p_i\}$ and $k \in \{1, 2, \ldots, p\}$, it holds that $\phi^i_{k,p} \in [0, 1]$. Besides, $\sum_{k=1}^{p} \phi^i_{k,p} = 1$ for all $i \in E_r$ and $p \in \{1, 2, \ldots, p_i\}$.

b) For all $i \in E_r$, the counting sequence associated with the negative binomial thinning operator a_i* is assumed to be independent of all other random variables appearing in (1).

c) For fixed $i, j \in E_r$, $\{\varepsilon_n(i,j)\}_{n\in\mathbb{N}}$ is a sequence of independent and equally distributed random variables.

d) $\{Z_n\}$, $\{\varepsilon_n(1,1)\}$, $\{\varepsilon_n(1,2)\}$,..., $\{\varepsilon_n(r,r)\}$ are sequences of mutually independent random variables.

e) Random variable $X_n(l)$ is independent of both Z_m and $\varepsilon_m(i,j)$ for all $n < m$ and all $i, j, l \in E_r$.

Finally, let $z_{n-1} = q$ and $z_n = s$ for some $q, s \in E_r$. If $0 \leq \alpha_s \leq \frac{\mu_s}{1+\max_{q\in E_r}\mu_q}$, then the distribution of the random variable $\varepsilon_n(q,s)$ can be written as

$$\varepsilon_n(q,s) \overset{d}{=} \begin{cases} Geom\left(\frac{\mu_s}{1+\mu_s}\right), & w.p.\ 1-\frac{\alpha_s\mu_q}{\mu_s-\alpha_s}, \\ Geom\left(\frac{\alpha_s}{1+\alpha_s}\right), & w.p. \quad \frac{\alpha_s\mu_q}{\mu_s-\alpha_s}. \end{cases}$$

This also holds for both $RrNGINAR_{max}(\mathcal{M},\mathcal{A},\mathcal{P})$ and $RrNGINAR_1(\mathcal{M},\mathcal{A},\mathcal{P})$ models.

Modeling using random environment *INAR* models first requires estimating the environment state for each realization in the observed sample (see [4]). The K-means technique has long been a popular choice due to its relatively simple algorithm and ease of application. In general, K-means divides a set of N p-dimensional data points $\{x_n\}_{n=1}^N$ into K clusters by minimizing the objective function

$$O = \sum_{n=1}^{N}\sum_{j=1}^{K} h_{nj} d(\mathbf{x}_n, \boldsymbol{\mu}_j), \tag{2}$$

where for all n and j, $\mathbf{x}_n$ is the column vector, $h_{nj} = 1$ if $\mathbf{x}_n$ belongs to cluster j and $h_{nj} = 0$ otherwise, while $\boldsymbol{\mu}_j$ is a centroid of the j-th cluster. Furthermore, d represents the Euclidean distance metric, given by

$$d(\mathbf{x}_n, \boldsymbol{\mu}_j) = \sqrt{\sum_{i=1}^{p}(x_{ni}-\mu_{ji})^2} = \sqrt{(\mathbf{x}_n-\boldsymbol{\mu}_j)^T(\mathbf{x}_n-\boldsymbol{\mu}_j)}. \tag{3}$$

Initially, this approach performed well, particularly in models where realizations within the each environment state were sufficiently similar. As noted in [2] and [8], K-means works best when clusters are spherically distributed. The mentioned similarity of data within each state causes the data to be organized into spherical clusters, allowing K-means to separate clusters with high accuracy and correctly assign environment states to most realizations. However, with the development of more complex *INAR* models in random environment, some issues emerged. Namely, these models introduced significant variability in realization values within the same state. In other words, realizations within the same environment state may exibit both very high and very low values, leading clusters to take on an elongated (ellipsoidal) rather than spherical shape. Since K-means assumes spherical clusters, its performance in such cases was significantly reduced. Another challenge arises when realizations from different states start to have similar values, i.e. when mean values within different states are approximately equal. Even if data within each state are spherically distributed, the spheres will overlap considerably. In this case, K-means fails to accurately separate the states, and forms clusters that do not reflect the true structure of the data.

Considering the aforementioned challenges, the authors in [13] developed a new clustering method to address them. Although the method is in theory applicable to data sequences corresponding to any *INAR* time series in a random environment, the authors have specifically focused on the case where the data to be clustered correspond to the $RrNGINAR(\mathcal{M},\mathcal{A},\mathcal{P})$ time series (described in [4]). They began with the observation that the K-means algorithm considers only the realization value x_n when estimating the environment state z_n at time n, $n \in \mathbb{N}$. However, the model parameters μ_{z_n}, α_{z_n} and P_n also contain valuable information about z_n. To minimize the information loss, the authors' primary objective was to construct a three-dimensional sequence of 'pre-estimators' derived from real-life realizations, that replicates the behavior of $\{(\mu_{z_n}, \alpha_{z_n}, P_n)\}_{n=1}^{\infty}$. Finally, the K-means algorithm was applied on the resulting three-dimensional

data sequence. In this manner, the information about z_n stays preserved. It is important to highlight that the given sequence of so-called 'pre-estimators' serves only as a supportive tool for estimating $\{z_n\}_{n=1}^{\infty}$, and does not constitute an alternative set of model parameter estimates.

Although it has brought improvements in clustering data corresponding to the $RrNGINAR(\mathcal{M},\mathcal{A},\mathcal{P})$ time series, the *RENES* method also has certain drawbacks. First and foremost, the method is highly complex, making its implementation far from straightforward. Secondly, it partially relies on the standard K-means algorithm, thereby inheriting some of its limitations. The shortcomings of the *RENES* method will be discussed in more detail in the next section.

In this manuscript, we introduce a modified *RENES* method (*MRENES*) designed to address the weaknesses of the original approach. The main idea is to minimize the number of steps preceding K-means within the *RENES* algorithm as much as possible, thereby reducing and the number of parameters required for the method to operate. Additionally, the metric used in the K-means technique has been adjusted to better accommodate the structure of the data being clustered.

The manuscript is organized as follows. In Section 2, we give the *RENES* clustering algorithm and outline all the identified shortcomings. These shortcomings motivate us to introduce modifications to the *RENES* in Section 3, leading to the development of the *MRENES* clustering method. Section 4 provides an extensive simulation study of the newly proposed *MRENES* method in scenarios with 2 and 3 different environment states. In Section 5 we present the results of applying the *MRENES* method to real-life data. Finally, Section 6 concludes the manuscript.

2. Troubles with *RENES*

To enable the reader to fully understand the essence of the *RENES* method and identify the source of its shortcomings, we will first describe, step by step, how the method works. For more details about *RENES*, see [13].

1. Let $\{X_n\}_{n=1}^N = \{X_n(z_n)\}_{n=1}^N$ be a sample of size $N \in \mathbb{N}$ from the $RrNGINAR(\mathcal{M},\mathcal{A},\mathcal{P})$ model. The first step in constructing the method is to derive sequences of pre-estimators $\{\tilde{\mu}_n\}_{n=1}^N$, $\{\tilde{\alpha}_n\}_{n=1}^N$ and $\{\tilde{P}_n\}_{n=1}^N$ for the parameter sequences $\{\mu_{z_n}\}_{n=1}^N$, $\{\alpha_{z_n}\}_{n=1}^N$, and $\{P_n\}_{n=1}^N$. These pre-estimators are constructed solely based on the observed sample, without knowing the realizations $\{z_n\}_{n=1}^N$ of the random environment process. More precise, for all $n \in \{1,2,\dots,N\}$, $\tilde{\mu}_n$, $\tilde{\alpha}_n$ and $\tilde{P}_n$ are defined as functions of x_n and, potentially, several neighboring realized elements of the sample. The three-dimensional sequence $\{\tilde{\mu}_n,\tilde{\alpha}_n,\tilde{P}_n\}_{n=1}^N$ such obtained is supposed to mimic the behavior of the model parameters over time. Based on the formulas given in [4], the pre-estimators are defined as follows:

$$\begin{aligned}
\tilde{\mu}_n &= X_n, \quad n \in \{1,2,\dots,N\},\\
\tilde{\alpha}_n &= \frac{\alpha_n^*}{\max\limits_{n=1,\dots,N}\alpha_n^*}, \quad n \in \{1,2,\dots,N\},\\
\tilde{P}_n &= \begin{cases} \max\limits_{K=1,\dots,p_{z_n}} pacf_K(X_1,\dots X_{2d_p+1}), & n \le d_p,\\ \max\limits_{K=1,\dots,p_{z_n}} pacf_K(X_{N-2d_p},\dots,X_N), & n > N-d_p,\\ \max\limits_{K=1,\dots,p_{z_n}} pacf_K(X_{n-d_p},\dots,X_{n+d_p}), & d_p < n \le N-d_p, \end{cases}
\end{aligned}$$

where $d_p \in \mathbb{N}$ and $pacf_K$ is the partial auto-correlation function at lag K. Furthermore,

$$\alpha_n^* = \begin{cases} A_n/B_n, & B_n \neq 0, \quad n > 1,\\ 1, & A_n = B_n = 0,\ n > 1,\\ \max\left\{\left(\frac{A_l}{B_l} : l \in \{2,\dots,N\}, B_l > 0\right)\right\}, & otherwise, \end{cases}$$

for all $n \in \{1, 2, \ldots, N\}$, whereas $A_n = (x_n - T(\tilde{\mu}_n, \mathbf{c}_m))_+$ and $B_n = \frac{1}{s}\sum_{i=1}^{s} A_{n-i}$ for $s = \min\{n-1, \tilde{P}_n\}$.

2. Although clustering of three-dimensional data $\{(\tilde{\mu}_n, \tilde{\alpha}_n, \tilde{P}_n)\}_{n=1}^N$ is feasible, the pre-estimators are further refined by involving trimmed (truncated) means. For that purpose, the function

$$T(a_n, \mathbf{c}) = \begin{cases} a_n, & n \le k \text{ or } n > N-k, \\ \sum_{j=n-k}^{n+k} c_{|j-n|} a_j, & k < n \le N-k \end{cases}$$

is used, where $N > 2k$ and $\mathbf{c} = (c_0, c_1, \ldots, c_k)$ is a $(k+1)$-dimensional vector of non-negative weights such that $c_0 \ge c_1 \ge \cdots \ge c_k,\ c_j > 0,\ j = 0, 1, \ldots, k$ and $c_0 + 2\sum_{j=1}^{k} c_j = 1$. Namely, since all of the pre-estimators $\{\mu_{z_n}\}_{n=1}^N$, $\{\alpha_{z_n}\}_{n=1}^N$, and $\{P_n\}_{n=1}^N$ are defined as functions of sample realizations, their values may vary considerably within the same state if realizations themselves fluctuate significantly. Given that the pre-estimators are intended to approximate model parameters-which are assumed to be fixed within each state-such variability is undesirable. To mitigate this issue, a trimmed mean is introduced to reduce excessive fluctuations in the values of pre-estimators and to stabilize them around representative levels.

3. The function

$$S(a_n, \mathbf{c}) = \frac{T(a_n, \mathbf{c}) \cdot N}{\sum_{n=1}^{N} T(a_n, \mathbf{c})}$$

is applied in order to equalize the impact of each particular coordinate of the three-dimensional sequence $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^N$ on clustering procedure.

4. By introducing three additional parameters, $C_m, C_a, C_p \in \mathbb{R}$, the influence of each coordinate on the clustering process has been regulated.

5. Finally, the *RENES* procedure ends with the application of the K-means clustering method to the three-dimensional data sequence

$$\left\{\left(C_m S(\tilde{\mu}_n, \mathbf{c}_m), C_a S(\tilde{\alpha}_n, \mathbf{c}_a), C_p S(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^N.$$

At first glance, the *RENES* method appears quite complex to implement, which is also its main drawback. Specifically, in order to assess the states of the environment, the method first requires the estimation of seven unknown parameters $(d_p, \mathbf{c}_m, \mathbf{c}_a, \mathbf{c}_p, C_m, C_a, C_p)$. This raises a logical question: Is it possible to apply K-means clustering earlier in the *RENES* algorithm, for example, after Step 2? Doing so would eliminate Steps 3 and 4, thereby reducing the number of unknown parameters required by the method. While this adjustment does simplify the *RENES* method, it also introduces several other challenges.

i. As mentioned in the introduction, K-means performs best when data within clusters are spherically distributed. The coefficients C_m, C_a and C_p from the fourth step of the *RENES* method are specifically introduced to provide this. More precisely, they are selected in such way to maximize the sphericity of data within clusters. Omitting these coefficients disrupts sphericity, which prevents the K-means from performing clustering with the desired accuracy.

ii. The coordinate values of the three-dimensional data sequence $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^N$ can vary significantly from one another. Specifically, pre-estimators $\{\tilde{\mu}_n\}_{n=1}^N$ can take arbitrarily large values, depending on the phenomenon being analyzed. In contrast, pre-estimators $\{\tilde{\alpha}_n\}_{n=1}^N$ always falls within the range $[0, 1]$, while $\{\tilde{P}_n\}_{n=1}^N$ aren't greater than 5 in most cases. According to (3), if one pre-estimator is consistently much larger than the others, it will have a disproportionately greater

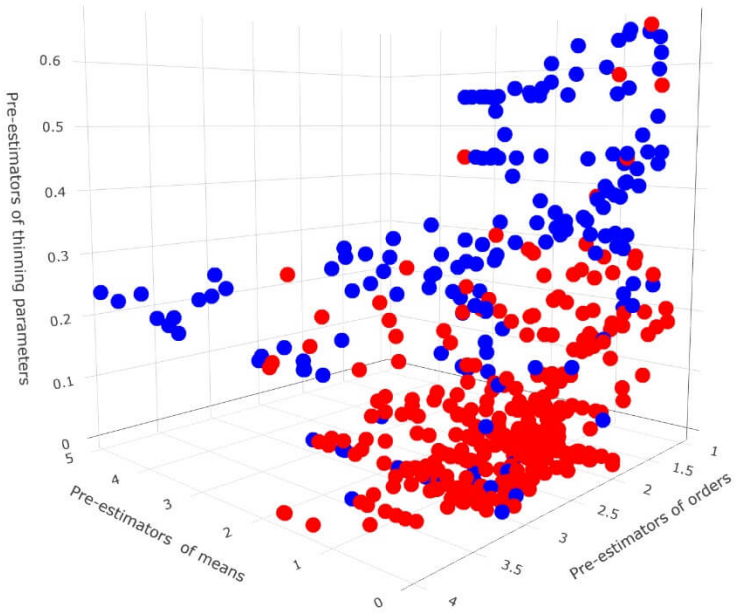

Figure 1: Three-dimensional pre-estimators $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ associated with the corresponding exact states. The pre-estimators were obtained based on the simulated $R2NGINAR(2,4)$ time series.

influence on the clustering process. However, such an uneven influence of the pre-estimators is undesirable, as the *RENES* assumes that all pre-estimators carry approximately the same piece of information about $\{z_n\}$. To address this issue, Step 3 was introduced within the *RENES* method. Eliminating this step would cause the problem to reappear.

iii. If we examine the formulas for calculating the initial pre-estimators $\tilde{\mu}_n, \tilde{\alpha}_n$ and $\tilde{P}_n$ given in Step 1, it can be expected that a correlation between the sequences $\{\tilde{\mu}_n\}_{n=1}^{N}$ and $\{\tilde{\alpha}_n\}_{n=1}^{N}$ will emerge. This correlation is likely to persist to some extent even after the transformation in Step 2 happens. Additionally, correlations may also appear in the other two pairs of pre-estimators. However, K-means is not well-suited for clustering data sequences with correlated coordinates, as it does not take the covariance structure of the data into account (a limitation that will become evident later). This can lead to suboptimal clustering when there are significant correlations between the sequences of pre-estimators.

Considering all the previously mentioned points, we can now define the research objective. The aim is to modify the *RENES* method to reduce the number of required parameters, while avoiding negative consequences that such modifications might introduce. The way to achieve this is outlined in the next section.

3. Modified *RENES* method

We aim to explore whether the *RENES* method can be simplified by applying the K-Means algorithm directly to $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$. In order to investigate this possibility, we simulated a $R2NGINAR(2,4)$ time series and applied the first two steps of the *RENES* method to the resulting data sequence. Such obtained three-dimensional data were then associated with the corresponding exact states, which is presented in Figure 1.

The figure demonstrates that the three-dimensional pre-estimators $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ might serve as reliable indicators of the exact states. One state (the red cluster) predominantly consists of points with low values for both $T(\tilde{\mu}_n, \mathbf{c}_m)$ and $T(\tilde{\alpha}_n, \mathbf{c}_a)$, while the other state (the blue cluster) includes points with either a high $T(\tilde{\mu}_n, \mathbf{c}_m)$ value or a high $T(\tilde{\alpha}_n, \mathbf{c}_a)$ value. Significant mixing of points occurs only in the boundary region. These observations suggest that estimating environment states $\{z_n\}_{n=1}^{N}$ by clustering the three-dimensional set of pre-estimators $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ is a reasonable approach. However, the points in both states do not exhibit a spherical, but rather an elongated, ellipsoidal shape. Besides, Table 1 shows significant correlations between sequences $\{T(\tilde{\mu}_n, \mathbf{c}_m)\}_{n=1}^{N}$, $\{T(\tilde{\alpha}_n, \mathbf{c}_a)\}_{n=1}^{N}$, and $\{T(\tilde{P}_n, \mathbf{c}_p)\}_{n=1}^{N}$. Consequently, the standard K-Means technique applied to $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ produces poor results, and introduction of certain modifications is necessary.

Pre-estimators	$\{T(\tilde{\alpha}_n, \mathbf{c}_a)\}_{n=1}^{N}$	$\{T(\tilde{P}_n, \mathbf{c}_p)\}_{n=1}^{N}$
$\{T(\tilde{\mu}_n, \mathbf{c}_m)\}_{n=1}^{N}$	0.198 [0.0421]	0.268 $\left[1.064 \cdot 10^{-9}\right]$
$\{T(\tilde{\alpha}_n, \mathbf{c}_a)\}_{n=1}^{N}$		−0.437 $\left[2.212 \cdot 10^{-15}\right]$

Table 1: Correlations between sequences of pre-estimators. The table contains Spearman correlation coefficient ρ with the corresponding p-value shown in brackets.

As given in [8], this is a case where the Euclidean distance within K-means must be replaced by the Mahalanobis distance. It is given by formula

$$d_M(\mathbf{x}_n, \boldsymbol{\mu}_j) = \sqrt{(\mathbf{x}_n - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_j)}, \tag{4}$$

where for all n and j, $\mathbf{x}_n$ is again a p-dimensional column vector, while $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ represent the centroid and covariance matrix of the j-th cluster, respectively. By comparing (3) and (4), it can be said that the Euclidean distance is a special case of the Mahalanobis distance where $\boldsymbol{\Sigma}_j = \mathbf{I}_{(p\times p)}$, i.e., where data coordinates are supposed to be uncorelated. On the other hand, the Mahalanobis distance from a point to its cluster center can be understood as the Euclidean distance scaled by the inverse of the square root of the variance in the direction of the point. In other words, the presence of the inverse of the covariance matrix in (4) allows different scales on which the pre-estimators are measured. Beside this, it allows nonzero correlations between pre-estimators.

As described in [7], for standardizing a variable X one must divide X by its standard deviation. This means that Mahalanobis distance, given by (4) and applied on $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$, first transforms data into an uncorrelated, standardized data sequence and then computes the Euclidean distance between the resulting three-dimensional pre-estimators and the transformed cluster centroids. Consequently, the K-Means technique enhanced with the Mahalanobis distance should facilitate the clustering of the three-dimensional sequence $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ while avoiding all the negative consequences discussed in the previous chapter.

However, proper initialization of $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$, $j = 1, 2, \ldots, K$, is required. For that purpose we use initialization with stretching (see [12]). In essence, the "stretching" technique relies on the result that $d_M^2 \sim \chi_p^2$ asymptotically and it consists of the following steps.

L1. Form an initial cluster of size w by identifying a point near the data mode and selecting the $w - 1$ nearest neighbors.

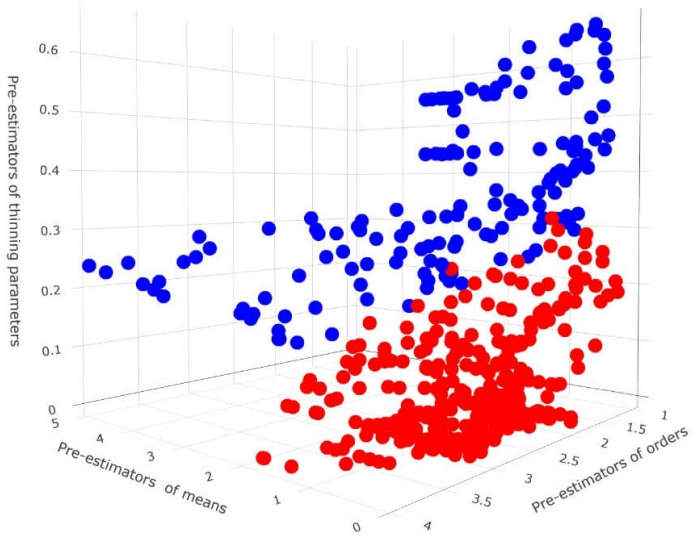

Figure 2: Clustering of the three-dimensional pre-estimators $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$, performed using K-means enhanced with the Mahalanobis distance. The pre-estimators were obtained based on the simulated $R2NGINAR(2, 4)$ time series.

L2. Estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using these w points.

L3. Construct the 95% confidence ellipsoid for the cluster and involve all the additional points falling within the ellipsoid into further calculation.

L4. Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

L5. Repeat steps L3 and L4 until no additional points are captured by the confidence ellipsoid.

L6. Finally, remove all these points from the data set and repeat steps L1-L5 until K initial clusters are formed.

For each $j = 1, 2, \ldots, K$, the most straightforward way to estimate $\boldsymbol{\Sigma}_j$ in steps L2 or L4 is to use the sample covariance matrix of the points assigned to cluster j. When a cluster contains fewer than p points or $\boldsymbol{\Sigma}_j$ is nearly singular, a regularization becomes a suitable alternative. Finally, if per-cluster estimation is unstable or noisy, one may instead use a global sample covariance matrix, computed over the entire set of N points. While such estimate is more robust, it sacrifices the flexibility of capturing cluster-specific shapes.

When we applied K-means enhanced with the Mahalanobis distance to the same data sequence $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ that generated Figure 1, we obtained the result as shown in Figure 2. As expected, the clusters are not spherical. However, that's desirable in this case, as the original data are not spherically distributed either. A comparison of Figure 1 and Figure 2 reveals a noticeable alignment between the corresponding clusters. In other words, the modified K-means method successfully identified the environment states with satisfactory accuracy in this specific instance. Although the presented figures illustrate results based on a single simulation, they indicate that the proposed approach is well-founded and merits further examination through a larger set of simulations.

Finally, we propose a modified *RENES* clustering method (*MRENES*) for data sequences corresponding to the $RrNGINAR(\mathcal{M}, \mathcal{A}, \mathcal{P})$ time series, structured as follows.

S1. Calculate pre-estimators $\{\tilde{\mu}_n\}_{n=1}^{N}$, $\{\tilde{\alpha}_n\}_{n=1}^{N}$, and $\{\tilde{P}_n\}_{n=1}^{N}$ as proposed in [13].

S2. Create $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$ same as in [13].

S3. Apply K-means with Mahalanobis distance to $\left\{\left(T(\tilde{\mu}_n, \mathbf{c}_m), T(\tilde{\alpha}_n, \mathbf{c}_a), T(\tilde{P}_n, \mathbf{c}_p)\right)\right\}_{n=1}^{N}$.

4. Simulation study

We aim to evaluate the performance of the new *MRENES* method using the same simulations where *RENES* has already demonstrated its effectiveness. If *MRENES* proves to be competitive even in this setting, this will serve as a strong argument for its further application to real-life data. To this end, we simulated time series of length 500 corresponding to the $RrNGINAR(\mathcal{M}, \mathcal{A}, \mathcal{P})$ model, considering cases with 2 and 3 different environment states. The model parameters are the same as those used in [13]. Each parameter combination is simulated in 50 replications. Additionally, for each parameter combination, both versions of the model ($RrNGINAR_{max}(\mathcal{M}, \mathcal{A}, \mathcal{P})$ and $RrNGINAR_1(\mathcal{M}, \mathcal{A}, \mathcal{P})$) are analyzed simultaneously.

We will compare the clustering results obtained using the *MRENES* method with those obtained using the *RENES* method and the standard K-means method. Since the comparison will be performed on the same simulations that were already used in [13], we are not going to estimate the optimal parameter values required for applying the *RENES* method ($d_p, \mathbf{c}_m, \mathbf{c}_a, \mathbf{c}_p, C_m, C_a, C_p$). Instead, we will take those values directly from [13]. In this way, we will also obtain parameter estimates required by the *MRENES* method ($d_p, \mathbf{c}_m, \mathbf{c}_a, \mathbf{c}_p$).

The comparison will be based on two criteria: the number of correctly estimated states (referred to as N_{CES} in the results table) and the Adjusted Rand Index (*ARI*). For all clustering methods, we report the mean values of N_{CES} and *ARI* with corresponding standard deviations, obtained across 50 simulations. Here, we provide a brief explanation of *ARI*. Let $X = \{x_n\}_{n=1}^{N}$ be a realized sample of length N, and let $A = \{A_1, A_2, \ldots, A_r\}$ and $B = \{B_1, B_2, \ldots, B_s\}$ be two different clusterings (partitions) of the set X. The overlap between clusters from A and clusters from B is defined by a contingency table (Table 2), where for all i and j $v_{i,j} = |A_i \cap B_j|$. Now, *ARI* is defined by the formula

Partitions	B_1	B_2	$\ldots$	B_s	Sums
A_1	v_{11}	v_{12}	$\ldots$	v_{1s}	a_1
A_2	v_{21}	v_{22}	$\ldots$	v_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
A_r	v_{r1}	v_{r2}	$\ldots$	v_{rs}	a_s
Sums	b_1	b_2	$\ldots$	b_s	

Table 2: Contingency table for partitions A and B.

$$ARI = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s}\binom{v_{ij}}{2} - \left[\sum_{i=1}^{r}\binom{a_i}{2}\sum_{j=1}^{s}\binom{b_j}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_{i=1}^{r}\binom{a_i}{2} + \sum_{j=1}^{s}\binom{b_j}{2}\right] - \left[\sum_{i=1}^{r}\binom{a_i}{2}\sum_{j=1}^{s}\binom{b_j}{2}\right]/\binom{N}{2}},$$

where v_{ij}, a_i and b_j are from Table 2 for all $i = 1, 2, \ldots, r$ and $j = 1, 2, \ldots, s$. In general, $ARI \in [-1, 1]$, greater values of *ARI* indicate stronger similarity between partitions and $ARI = +1$ determines the perfect match. For more information regarding *ARI*, see [3]. In this particular case, one partition represents real environment states, while the other corresponds to the clustering results obtained using one of the proposed methods.

4.1. Simulations with two environment states

Table 3 presents the parameter combinations used for simulating $RrNGINAR(\mathcal{M},\mathcal{A},\mathcal{P})$ time series with two environment states. As previously mentioned, these are the same combinations used in [13]. The clustering results obtained using the K-means, *RENES* and *MRENES* techniques will be discussed for each combination separately.

Case 1. $\mathcal{M}=(1,1.5)$, $\mathcal{A}=(0.05,0.6)$, $\mathcal{P}=(2,4)$, $p_{vec}=(0.6,0.4)$, $p_{mat}=\begin{bmatrix}0.9 & 0.1\\ 0.2 & 0.8\end{bmatrix}$,

$$\phi_1=\begin{bmatrix}1 & 0\\ 0.9 & 0.1\end{bmatrix},\ \phi_2=\begin{bmatrix}1 & 0 & 0 & 0\\ 0.1 & 0.9 & 0 & 0\\ 0.1 & 0.45 & 0.45 & 0\\ 0.1 & 0.1 & 0.4 & 0.4\end{bmatrix}.$$

Case 2. $\mathcal{M}=(3,5)$, $\mathcal{A}=(0.4,0.5)$, $\mathcal{P}=(2,5)$, $p_{vec}=(0.5,0.5)$, $p_{mat}=\begin{bmatrix}0.8 & 0.2\\ 0.25 & 0.75\end{bmatrix}$,

$$\phi_1=\begin{bmatrix}1 & 0\\ 0.4 & 0.6\end{bmatrix},\ \phi_2=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0.2 & 0.8 & 0 & 0 & 0\\ 0.4 & 0.4 & 0.2 & 0 & 0\\ 0.3 & 0.3 & 0.3 & 0.1 & 0\\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1\end{bmatrix}.$$

Table 3: Parameter combinations used for simulating $R2NGINAR(\mathcal{M},\mathcal{A},\mathcal{P})$ time series.

4.1.1. R2NGINAR(2, 4) simulations

To start the procedure, 50 replications of the $R2NGINAR_{max}(2,4)$ time series and 50 replications of the $R2NGINAR_1(2,4)$ time series are simulated. For each version, Standard K-means, *RENES* and *MRENES* are applied on each replication. Table 4 shows optimal values of the parameters required for the *RENES* clustering method. Thus, optimal values of the parameters required for the *MRENES* method are also defined. The clustering results obtained using the three specified techniques across 50 simulations are presented in Table 5.

$R2NGINAR_{max}(2,4)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
8	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	6	2	9
$R2NGINAR_1(2,4)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
15	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	8	2	3

Table 4: Optimal values of the *RENES* method parameters in the case of simulated $R2NGINAR(2,4)$ time series.

$R2NGINAR(2,4)$						
Comparison	$R2NGINAR_{max}(2,4)$			$R2NGINAR_1(2,4)$		
technique	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
N_{CES}	311	323	328	316	333	356
	[30.09]	[22.79]	[11.97]	[31.79]	[20.14]	[12.57]
ARI	0.042	0.073	0.092	0.069	0.098	0.153
	[0.021]	[0.020]	[0.011]	[0.031]	[0.025]	[0.018]

Table 5: Mean values of N_{CES} i *ARI*, with corresponding standard deviations shown in brackets, reported across 50 simulated $R2NGINAR(2,4)$ time series.

As table shows, both *RENES* and *MRENES* methods outperformed the K-means technique in each of two versions of the *R2NGINAR*(2, 4) time series. This advantage was confirmed by the average number of correctly estimated states (N_{CES}) and by the average *ARI* value. However, in the case of *R2NGINAR*$_{max}$(2, 4) simulations, the difference in the accuracy of state estimation between the *RENES* and *MRENES* methods is very small, almost negligible. This is not surprising, as both methods are based on the same underlying logic. While the *MRENES* method has two fewer steps, it compensates for the omitted steps by incorporating the Mahalanobis distance. Additionally, although the *MRENES* shows a slight advantage over *RENES* in terms of average values of N_{CES} and *ARI*, its key benefit lies in simplifying the algorithm and reducing the number of required parameters. This significantly improves the practical applicability of the clustering method without compromising the estimation quality, which is an advantage that should not be overlooked.

The difference in accuracy of state estimation between the *RENES* and *MRENES* methods is more pronounced in the case of *R2NGINAR*$_1$(2, 4) simulations, where the *MRENES* method provided more accurate estimation of the environment state sequence compared to *RENES*. Interestingly, the average values of N_{CES} and *ARI* indicate that the improvement achieved by *MRENES* over *RENES* is even greater than the improvement that *RENES* achieved over K-means.

4.1.2. *R2NGINAR*(2, 5) *simulations*

Same as in previous case, 50 replications of the *R2NGINAR*$_{max}$(2, 5) time series and 50 replications of the *R2NGINAR*$_1$(2, 5) time series are simulated. Table 6 shows optimal values of the parameters required for the *RENES* clustering method. The clustering results obtained using Standard K-means, *RENES* and *MRENES* across 50 simulations are presented in Table 7.

R2NGINAR$_{max}$(2, 5)						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
17	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	4	2	3
R2NGINAR$_1$(2, 5)						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
9	(0.2,0.2,0.2)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	9	6	7

Table 6: Optimal values of the *RENES* method parameters in the case of simulated *R2NGINAR*(2, 5) time series.

R2NGINAR(2, 5)						
Comparison	*R2NGINAR*$_{max}$(2, 5)			*R2NGINAR*$_1$(2, 5)		
technique	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
N_{CES}	262	300	298	265	296	297
	[28.82]	[21.13]	[12.25]	[30.87]	[22.01]	[13.78]
ARI	0.043	0.069	0.068	0.045	0.068	0.068
	[0.020]	[0.018]	[0.012]	[0.022]	[0.018]	[0.014]

Table 7: Mean values of N_{CES} i *ARI*, with corresponding standard deviations shown in brackets, reported across 50 simulated *R2NGINAR*(2, 5) time series.

Table 7 demonstrates a strong similarity in the accuracy of state estimation between the *RENES* and *MRENES*, both of which significantly outperform the K-means. As previously noted, achieving comparable results with a considerably simplified algorithm is a noteworthy finding for *MRENES*. This same conclusion holds for both the *R2NGINAR*$_{max}$(2, 5) and *R2NGINAR*$_1$(2, 5) simulations.

Additionally, it is important to highlight that the *RENES* method exhibited significantly higher volatility compared to the *MRENES*. In simulations, the number of correctly estimated states obtained using the *RENES* method spanned the interval from 245 to 361. On the other hand, the *MRENES* method consistently

produces the N_{CES} results within 20 of the mean. Moreover, the standard deviations for *MRENES* method are lower across both criteria compared to those for *RENES*, indicating consistent clustering performance. This consistency represents an additional advantage the *MRENES* method offers.

4.2. Simulations with three environment states

In case of simulated $RrNGINAR(\mathcal{M}, \mathcal{A}, \mathcal{P})$ time series with three environment states, required parameter combinations are presented in Table 8. Again, these are the same combinations used in [13].

Case 1. $\mathcal{M} = (0.5, 1, 1.5)$, $\mathcal{A} = (0.1, 0.35, 0.6)$, $\mathcal{P} = (2, 4, 2)$, $p_{vec} = (0.3, 0.4, 0.3)$, $p_{mat} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$,

$$\phi_1 = \begin{bmatrix} 1 & 0 \\ 0.9 & 0.1 \end{bmatrix}, \phi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.8 & 0 & 0 \\ 0.2 & 0.4 & 0.4 & 0 \\ 0.2 & 0.2 & 0.3 & 0.3 \end{bmatrix}, \phi_3 = \begin{bmatrix} 1 & 0 \\ 0.1 & 0.9 \end{bmatrix}.$$

Case 2. $\mathcal{M} = (2, 4, 6)$, $\mathcal{A} = (0.2, 0.3, 0.6)$, $\mathcal{P} = (2, 4, 5)$, $p_{vec} = (0.35, 0.35, 0.3)$, $p_{mat} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$,

$$\phi_1 = \begin{bmatrix} 1 & 0 \\ 0.7 & 0.3 \end{bmatrix}, \phi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.3 & 0.3 & 0.4 & 0 \\ 0.3 & 0.2 & 0.2 & 0.3 \end{bmatrix}, \phi_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.4 & 0.6 & 0 & 0 & 0 \\ 0.2 & 0.5 & 0.3 & 0 & 0 \\ 0.25 & 0.3 & 0.2 & 0.25 & 0 \\ 0.2 & 0.2 & 0.3 & 0.1 & 0.2 \end{bmatrix}.$$

Table 8: Parameter combinations used for simulating $R3NGINAR(\mathcal{M}, \mathcal{A}, \mathcal{P})$ time series.

4.2.1. $R3NGINAR(2, 4, 2)$ simulations

First of all, we simulated 50 replications of the $R3NGINAR_{max}(2, 4, 2)$ time series and 50 replications of the $R3NGINAR_1(2, 4, 2)$ time series. The optimal parameter values required for the *RENES* clustering method are given in Table 9, while clustering results obtained using Standard K-means, *RENES* and *MRENES* across 50 simulations are presented in Table 10.

$R3NGINAR_{max}(2,4,2)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
17	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	9	7	2
$R3NGINAR_1(2,4,2)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
18	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	6	1	8

Table 9: Optimal values of the *RENES* method parameters in the case of simulated $R3NGINAR(2, 4, 2)$ time series.

When discussing $R3NGINAR(2, 4, 2)$ simulations, *MRENES* achieved performance almost identical to *RENES* (in the $R3NGINAR_{max}(2, 4, 2)$ version) or slightly worse (in the $R3NGINAR_1(2, 4, 2)$ version). These results are satisfactory, given other benefits the significantly simplified *MRENES* algorithm offers. Both methods significantly outperformed K-means.

4.2.2. $R3NGINAR(2, 4, 5)$ simulations

Finally, we still need to examine the performance of the *MRENES* clustering method on $R3NGINAR(2, 4, 5)$ simulations. For this purpose, we simulated 50 replications of the $R3NGINAR_{max}(2, 4, 5)$ time series and 50 replications of the $R3NGINAR_1(2, 4, 5)$ time series. The optimal parameter values required for the *RENES*

$R3NGINAR(2,4,2)$						
Comparison	$R3NGINAR_{max}(2,4,2)$			$R3NGINAR_1(2,4,2)$		
technique	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
N_{CES}	162	208	210	159	212	202
	[20.44]	[14.19]	[10.22]	[19.09]	[15.32]	[9.93]
ARI	0.013	0.026	0.028	0.012	0.028	0.024
	[0.017]	[0.015]	[0.016]	[0.017]	[0.016]	[0.016]

Table 10: Mean values of N_{CES} i *ARI*, with corresponding standard deviations shown in brackets, reported across 50 simulated $R3NGINAR(2,4,2)$ time series.

clustering method are given in Table 11. The clustering results obtained using Standard K-means, *RENES* and *MRENES* are presented in Table 12.

$R3NGINAR_{max}(2,4,5)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
12	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	10	3	1
$R3NGINAR_1(2,4,5)$						
d_p	$\mathbf{c}_m$	$\mathbf{c}_a$	$\mathbf{c}_p$	C_m	C_a	C_p
11	(0.16,0.14,0.14,0.14)	(0.16,0.14,0.14,0.14)	(0.4,0.3)	7	5	2

Table 11: Optimal values of the *RENES* method parameters in the case of simulated $R3NGINAR(2,4,5)$ time series.

$R3NGINAR(2,4,5)$						
Comparison	$R3NGINAR_{max}(2,4,5)$			$R3NGINAR_1(2,4,5)$		
technique	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
N_{CES}	170	201	221	168	199	218
	[19.37]	[15.42]	[11.01]	[18.87]	[16.03]	[10.83]
ARI	0.019	0.035	0.044	0.020	0.033	0.043
	[0.021]	[0.015]	[0.013]	[0.020]	[0.014]	[0.012]

Table 12: Mean values of N_{CES} i *ARI*, with corresponding standard deviations shown in brackets, reported across 50 simulated $R3NGINAR(2,4,5)$ time series.

As shown in Table 12, the *MRENES* method achieved noticeably better results than the *RENES* method in $R3NGINAR(2,4,5)$ simulations. Both N_{CES} and *ARI* confirm this. Additionally, both clustering methods significantly outperformed K-means.

5. Real-life data application

In this section, we will apply the same reasoning as in the simulation section. Specifically, we will evaluate the practical value of the *MRENES* clustering method using the same real-life dataset on which *RENES* has already demonstrated its effectiveness. To that end, we will analyze data representing the daily number of newly detected COVID-19 cases on the island of Mauritius, covering the period from March 18, 2020, to April 25, 2021. This dataset is available on http://www.data.europa.eu. As described in [13], both the beginning and the end of the data sequence are characterized by frequent and abrupt increases and decreases in the number of newly identified COVID-19 cases (see Figure 3). Hence, these two subperiods can be interpreted as representing a single, distinct environment state. The remaining data, which fluctuate within expected limits, will be considered as having been observed under a different environment state. Further, [13] demonstrates that the given data sequence is well-suited for modeling using $R2NGINAR(2,4)$ and $R2NGINAR(2,5)$ models, as the autocorrelation function indicates that all lags up to order 5 are statistically significant.

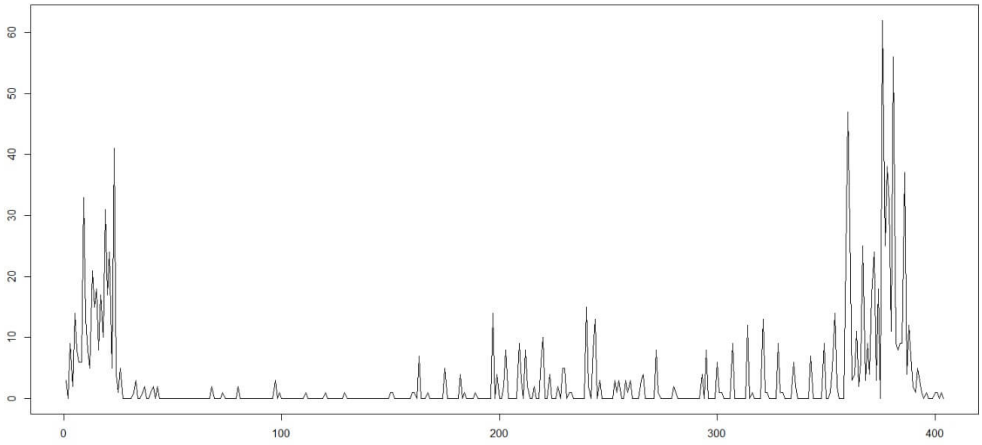

Figure 3: Daily number of newly detected COVID-19 cases on Mauritius

As we know, to assess the fitting quality of any *INAR* model in random environment, it is first necessary to estimate the sequence of environment states $\{z_n\}$. This also holds for *R2NGINAR*$(2,4)$ and *R2NGINAR*$(2,5)$ models. However, we mentioned earlier that the data with such significant oscillations is not well-suited for clustering using the K-means clustering technique. K-means in such cases fails to predict the states with sufficient accuracy, which leads to inadequate model application. As a result, significant discrepancies arise between observations and corresponding predicted values obtained using the aforementioned models. Authors in [13] showed that the *RENES* method has produced better results in this context. We will now investigate the fitting quality of given data using the same *R2NGINAR*$(2,4)$ and *R2NGINAR*$(2,5)$ models, where the sequence $\{z_n\}$ will be estimated using the *MRENES* clustering method. Parameters requested for the *MRENES* method will be taken from Section 4. The obtained results will be compared with the modeling results given in [13], that were preceded by state estimation using the K-means and *RENES* techniques. The root mean square (*RMS*) of the differences between observed and predicted values will be used as the measure of goodness of fit. Corresponding *RMS*-s obtained after application of *R2NGINAR*$_{max}(2,4)$, *R2NGINAR*$_1(2,4)$, *R2NGINAR*$_{max}(2,5)$ and *R2NGINAR*$_1(2,5)$ models are provided in Table 13.

	R2NGINAR$_{max}(2,4)$			*R2NGINAR*$_1(2,4)$		
Clustering method	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
RMS	4.259	3.870	3.762	4.216	3.827	3.617
	R2NGINAR$_{max}(2,5)$			*R2NGINAR*$_1(2,5)$		
Clustering method	K-means	*RENES*	*MRENES*	K-means	*RENES*	*MRENES*
RMS	4.150	3.768	3.830	4.152	3.797	3.841

Table 13: *RMS*-s obtained after application of two different *R2NGINAR*$_{max}(\mathcal{M},\mathcal{A},\mathcal{P})$ and *R2NGINAR*$_1(\mathcal{M},\mathcal{A},\mathcal{P})$ models on selected real-life data (three clustering methods are considered).

Table 13 clearly demonstrates that the choice of clustering method significantly influences the fitting quality for data corresponding to *R2NGINAR*$(\mathcal{M},\mathcal{A},\mathcal{P})$ time series. In particular, the fit improves substantially when the environment state sequence $\{z_n\}$ is estimated using either the *RENES* or *MRENES* methods. This observation holds for both the *R2NGINAR*$(2,4)$ and *R2NGINAR*$(2,5)$ models. However, when comparing the *RMS* values that models produce after applying the *RENES* method with those produced after using the *MRENES* method, the conclusions are not entirely straightforward. Specifically, the fitting accuracy of the *R2NGINAR*$(2,4)$ model shows a slight improvement with the application of the *MRENES* clustering technique. The improvement is more noticeable for the *R2NGINAR*$_1(2,4)$ model compared to

$R2NGINAR_{max}(2,4)$. Conversely, the fitting quality of the $R2NGINAR(2,5)$ model experiences a marginal decline after applying the *MRENES*. Nonetheless, the observed decrease is minimal and does not significantly impact the practical utility of the method itself. Moreover, it is important to highlight here the simplification brought by *MRENES* relative to *RENES*, particularly in terms of the reduced number of required parameters. The fact that we obtained nearly the same fitting quality with the help of a significantly simplified clustering method represents a meaningful contribution in its own right. This way, usefulness of the *MRENES* method is proved and the benefits of its use are confirmed. Finally, for comparing the modeling results given in Table 13 with the results obtained using various models with stationary or non-stationary nature, see [13].

6. Conclusion

This article defines a useful modification of the *RENES* method (called *MRENES*) for estimating random environment sequence $\{z_n\}$ for data corresponding to the *RrNGINAR* $(\mathcal{M}, \mathcal{A}, \mathcal{P})$ time series. Although the *RENES* method significantly improves clustering performance on the given data, it is quite complex to implement. It requires the estimation of a large number of parameters and demands numerous steps to execute the algorithm. The newly proposed *MRENES* clustering method aims to address these drawbacks by eliminating two of the four preliminary steps in the *RENES* algorithm that precede the application of K-means. The impact of the removed steps is compensated by replacing the Euclidean distance within K-means with the Mahalanobis distance. This substitution allows K-means to effectively cluster p-dimensional data with correlated coordinates measured on different scales. As a result, less preprocessing of the input data is needed, rendering two steps in the *RENES* algorithm redundant.

The performance of the *MRENES* technique was evaluated on both simulated and real-life datasets, with results compared to those obtained using *RENES* and standard K-means. The findings show that *MRENES* can be a suitable technique for clustering data generated by *RrNGINAR* $(\mathcal{M}, \mathcal{A}, \mathcal{P})$ time series. First of all, it significantly outperforms standard K-means in terms of the number of correctly estimated states and *ARI* values. Compared to *RENES* method, *MRENES* yields similar or slightly better results. However, when assessing the contribution of the *MRENES* method, one must also consider the simplifications it introduces. Achieving comparable (or even superior) results with a significantly simplified and more user-friendly approach is, in itself, a noteworthy contribution.

Future research may proceed in several directions. First, one could investigate the conditions under which K-means can be applied directly to the three-dimensional sequence of pre-estimators $\{\tilde{\mu}_n, \tilde{\alpha}_n, \tilde{P}_n\}_{n=1}^{N}$, without using the trimmed means function T. This would further reduce the burden of the clustering process. Furthermore, it would be interesting to examine the applicability of the proposed *MRENES* method for clustering high-dimensional data. In addition, identifying a procedure for the automatic selection of *MRENES* parameter values $(d_p, \mathbf{c}_m, \mathbf{c}_a, \mathbf{c}_p)$ would be of particular importance, as it would greatly facilitate the method's use and enhance its practical applicability.

References

[1] M. A. Al-Osh, A. A. Alzaid, *First-order integer-valued autoregressive (INAR(1)) process*, J. Time Series Anal. **8** (1987), 261–275.
[2] H. He, Y. He, F. Wang, W. Zhu, *Improved K-means algorithm for clustering non-spherical data*, Expert Systems **39** (2022), e13062.
[3] L. Hubert, P. Arabie, *Comparing partitions*, J. Classification **2** (1985), 193–218.
[4] P.N. Laketa, A. S. Nastić, M. M. Ristić, *Generalized Random Environment INAR Models of Higher Order*, Mediterr. J. Math. **15** (2018), 9–30.
[5] A. Latour, *Existence and stochastic structure of a non-negative integer-valued autoregressive process*, J. Time Series Anal. **19** (1998), 439–455.
[6] E. McKenzie, *Some simple models for discrete variate time series*, Water Resources Bulletin **21** (1985), 645–650.
[7] G. J. McLachlan, *Mahalanobis distance*, Resonance **4** (1999), 20–26.
[8] I. Melnykov, V. Melnykov, *On K-means algorithm with the use of Mahalanobis distances*, Statist. Probab. Lett. **84** (2014), 88–95.

[9] A. S. Nastić, P. N. Laketa, M. M. Ristić, *Random Environment Integer Valued Autoregressive process*, J. Time Series Anal. **37** (2016), 267–287.
[10] A. S. Nastić, P. N. Laketa, M. M. Ristić, *Random Environment INAR models of higher order*, REVSTAT **17** (2017), 35–65.
[11] A. S. Nastić, M. M. Ristić, A. D. Janjić, *A Mixed Thinning Based Geometric INAR(1) Model*, Filomat **31** (2017), 4009–4022.
[12] J. D. Nelson, *On K-means clustering using Mahalanobis distance* (Master's thesis, Graduate Faculty of the North Dakota State University of Agriculture and Applied Science), NDSU Libraries Institutional Repository, (2012). https://core.ac.uk/outputs/211306719
[13] B. A. Pirković, P. N. Laketa, A. S. Nastić, *On Generalized Random Environment INAR Models of Higher Order: Estimation of Random Environment States*, Filomat **35** (2021), 4545–4576.
[14] M. M. Ristić, H. S. Bakouch, A. S. Nastić, *A new geometric first-order integer-valued autoregressive (NGINAR(1)) process*, J. Statist. Plann. Inference **139** (2009), 2218–2226.
[15] H. Zheng, I. V. Basawa, S. Datta, *Inference for pth-order random coefficient integer-valued autoregressive processes*, J. Time Series Anal. **27** (2006), 411–440.