Threshold autoregressive model blind identification based on array clustering

Jean-Marc Le Caillec

Version of Record: https://www.sciencedirect.com/science/article/pii/S0165168421000931
Manuscript_247c84d0f449c68a430b1b79020bb187

Threshold Autoregressive Model Blind Identification based on array clustering

Jean-Marc Le Caillec

IMT Atlnatique UMR CNRS 6285 Lab-STICC, Technople Brest Iroise CS 83818 29238 Brest Cedex 3

Abstract

In this paper, we propose a new algorithm to estimate all the parameters of a Self Exited Threshold AutoRegressive (SETAR) model from an observed time series. The aim of this algorithm is to relax all the hypotheses concerning the SETAR model for instance, the knowledge (or assumption) of the number of regimes, the switching variables, as well as of the switching function. For this, we reverse the usual framework of SETAR model identification of the previous papers, by first identifying the AR models using array clustering (instead of the switching variables and function) and second the switching conditions (instead of the AR models). The proposed algorithm is a pipeline of well-known algorithms in image/data processing allowing us to deal with the statistical non-stationarity of the observed time series. We pay a special attention on the results of each step over the possible discrepancies over the following step. Since we do not assume any SETAR model property, asymptotical properties of the identification results are difficult to derive. Thus, we validate our approach on several experiment sets. In order to assess the performance of our algorithm, we introduce global metrics and ancillary metrics to validate each step of the proposed algorithm.

Keywords: SETAR model, blind identification, SVM

1. Introduction

The space model or dynamic equation is a key point in the wide field of the signal processing/time series analysis to understand the (physical/physiological/financial)

*Corresponding Author
Email address: jm.lecaillec@imt-atlantique.fr (Jean-Marc Le Caillec)

mechanisms generating an observed signal or time series. In some cases, a nonlinear/nonstationary state model is necessary to include all the signal/time series features (see [1, 2, 3, 4] for nonlinear, nonstationary or long memory random time series among the numerous possible dynamic equation models or statistical properties of the time series). Self Exited Threshold AutoRegressive (SETAR or simply TAR, although a discussion on the difference between these two modelling is given in [5]) models, since their introduction by Howell Tong in [6], have been paid attention in the nonlinear/nonstationary time series modelling or signal processing domains. The stability/ergodicity/stationarity conditions of such switching models have been studied in several papers. Indeed, in the papers referred in this introduction, stationarity stands for the system stability (probabilistic meaning) leading to some confusion with the statistical meaning. In fact, the output of such models is non stationary in the statistical meaning, since for instance the local time series variance depends on the active regime as discussed again below. In particular, in pioneering papers, Petrucelli [7, 8] has shown that SETAR models based on AR models with one root on or outside the unit circle can be stable for models of order 1 while obviously a sole AR model is not. The probailistic meaning of stationarity is derived from Markov chain approaches, which have been explicitly tackled in [8, 9, 10] for SETAR models with two AR models of order 1 and generalized to multivariate (AR order greater than 1) approach in [11] and multi-model (more than 2 AR models) in [12, 13, 14, 15, 16]. The SETAR models identification remains an open question although it exists some relevant contributions but under some restrictive assumptions on the model structure or on the statistical properties of the innovation. The first group of papers dealing with SETAR model identification is based over the sequential estimation of the SETAR model, that is first the switching variables and conditions and second the AR model estimates (order and parameters) with the underlying hypothesis that the number of regimes is known. In the original paper, Tong and Lim propose an algorithm searching first the switching variable (or more accurately the lag) based on the quantile of times series ([6] section 8). In [7], Petruccelli proposes a least square approach limited to a 2 AR model of order 1. In Chan [21], the author proved the consistency of these least squares estimates of the AR model coefficients and the as well as that of the lag (i.e. time series past value) driven the regime switching by finally assuming that this lag is known. This proposed method can be inconsistent when the innovation/residual variance is not equal for the two regimes. Several papers have proposed a Bayesian approach and then a

Reference	Known number of regimes	Known AR orders	One switching variable	Known delay	Same innovation variance	Gaussian innovation
[6]	Yes(2)	No	Yes	No	No	No
[8]	Yes(2)	Yes	Yes	Yes	Yes	No
[7]	Yes(2)	Yes	Yes	Yes	No	No
[16]	Yes(2)	No	Yes	No	No	No
[21]	Yes(2)	No	Yes	No	Yes and No	No
[25]	Yes(k)	No	Yes	No	No	No
[26]	Yes(2)	No	No	No	No	No
[23]	Yes (k)	No	Yes	No	No	Yes
[22]	Yes(2)	No	Yes	Yes	Yes	Yes

Table 1: SETAR identification algorithm and underlying assumptions. Papers are ranked by years .

Maximum Likehood derivation of the AR coefficients. For instance, in [6], the identification process is based under the assumption of 2 regimes and Gaussian innovation (see also [22]). Some papers are focused on a particular point of the SETAR identification such as the threshold governing the switching [23] or the number of regimes [24]. A summary of this topic of SETAR modelling, statistical properties and identification of the underlying assumption of these aforementioned methods can be found for instance in chapter 3 of [3] and a sum up of the underlying hypotheses for SETAR identification is given in table 1 (for some papers). As seen in this table, all the papers assume a known number of regimes. However, even if some papers propose a generic value of regimes (i.e. k), they do not propose a process to select the optimal number of regimes. The aforementioned papers, devoted to SETAR modelling and identification, can be divided into two categories that is theoretical and empirical papers. The theoretical papers manage to derive some interesting theorems but under some (very) restrictive hypotheses that cannot be easily verified by a practitioner through hypothesis testing for instance. This paper belongs to the second category since we propose a computational approach to identify the SETAR model, but asymptotic properties are difficult to prove since we relax all the assumptions on the SETAR model. It belongs to the emerging trend of identifying/estimating models with nonstationary output through array clustering reducing the necessary knowledge of the system modelling (see [27] for instance). For this reason, in section 5, we test several SETAR models (AR models and switching variables/conditions) combined with several values of the algorithm hyperparameters. For the identification purpose, we reverse the usual process (detailed above) by first identifying "patterns" among the time series samples

assuming that they belong to a same regime and then estimating the switching region and variables. In fact, our algorithm is based on a pipeline of well-known clustering/estimation/classification algorithms. Thus, in what follows, we have detailed the basics of each algorithm in order to understand the effects on the performance of each step over the following one. In section 2, we detail the SETAR model as well as the key points of this model identification/validation. In section 3, we focus on the first point of the SETAR identification (AR model identification), while the second point (switching variables and condition) is developed in section 4. Numerical results are given in section 5, with details on the experiments and performance metrics in the first two subsections.

2. Self-Exited Threshold AutoRegressive model Identification

We consider a switching regime model driven by thresholds that are hard limits (unlike Smooth Threshold AutoRegressive, STAR, models see [28] for instance). Assuming R regimes, this model can be written as:

$$Y(n) = \sum_{r=1}^{R} \Big(\sum_{i=1}^{p_r} a_i^r Y(n-i) + e_r(n) \Big) I\Big(g(Y(n-i_1), \cdots, Y(n-i_k)) \in B_r \Big) \quad (1)$$

where a_i^r is the *ith* coefficient of the *rth* regime autoregressive model of order p_r and we denote the array $A_r = [a_1^r, \cdots, a_{p^r}^r]$, $I()$ being the indicator function and $g(Y(n-i_1), \cdots, Y(n-i_k))$ is the condition governing the switching between the different regimes, B_r being disjoint sets. In particular, model eq. (1) allows e_r, the zero mean innovation/residual, to follow different laws and possibly to exhibit different skewness and variance depending on the regime. $(i_1, \ldots, i_k)$ are k unknowns delays (k being not known either) indexing the switching variables, (i.e. past lags). In what follows, we assume that $g()$ can be a multivariate output function (i.e. $\mathbb{R}^k \rightarrow \mathbb{R}^{k'}$, $k' \leq k$, as in the case of the nested SETAR model case [29] for instance). Asymptotical properties of model of eq. (1) are generally difficult to establish without introducing restricting assumptions. A trivial stability condition is that all the AR models have roots inside the unit circle. Another sufficient condition for SETAR stability in eq. (1) is that $\sum_{i=1}^{p_r} |a_i^r| < 1$ ([17] Lemma 3.1, [12], [21], [16]). However, this condition is not necessary as proved by the examples by Petrucelli. Thus, SETAR models can be built upon non stable AR models [18, 19] and can be extended to other heteroskedastic models (such as GARCH models [20]). In this paper we consider SETAR models based on AR models having possibly a root on or outside the unit circle (random walk

[18], [19]) and thus the output of models of eq. (1) is non stationary [3] (statistical meaning), possibly non Gaussian and obviously this model is nonlinear. In particular SETAR models can include ARIMA models having single or multiple zeros on the unit circle (that produce well-known non stationary output [30]), but also AR models with zeros outside the unit circle, these models being not useful by themselves, since they lead to strongly divergent/explosive time series when used alone. Moreover, nonlinear autoregressive models (see [31] for instance) could be included in our algorithm, but it would strongly increase the computational complexity. The more general approach for identifying SETAR models can be divided into four main points and a fifth optional point.

- The first involves estimating the number of regimes R in eq. (1).
- The second is to estimate the AR model parameters for each regime, (i.e. a_i^r for $i \in 1, \cdots, p_r$ and $r \in 1, \cdots, R$).
- The third is to determine which variables drive the switching of regimes that is the lags $i_1, \cdots i_k$ in eq. (1).
- Obviously the fourth purpose is to estimate the structure of the function $g()$ involved in the switching regime in eq. (1).
- A last step is a validation step of the estimated model but as seen below.

For this last point, our algorithm being based upon a recursive minimization of the residuals, approaches verifying the residual whiteness and their minimum variance are not well suited. In our case, a cross validation approach has been developed for model validation, thus separating the time series into estimation basis of N samples and validation basis of N_v samples, the total time series length being $N_T = N + N_v$. For the results of section 5, we set $N = \frac{3}{4}N_T$ (usual value for cross validation). In the next two sections, we detail the two main parts of our algorithm, first the AR models estimate and second the switching variables/condition.

3. Number of regimes and AR parameters estimation

As stated in the introduction, a data processing point of view can be adopted to find similar "patterns" in the observed signal/time series in order to retrieve signal sample sequence belonging to a same regime. Thus, the first step involves gathering the time series lags into arrays of length $d + 1$ (the value of this

parameter is discussed at the end of the section), that is $\mathbf{Y}_d(n) = (Y(n), Y(n-1), \ldots, Y(n-d))$. According to the model of definition (1), each array has to be labelled according to the regime generating the first array component that is $Y(n)$. This label, denoted $l(n)$ is an integer between in $\{1, \cdots, R\}$, R being unknown, but assumed to be lower than a value denoted R_{max}. Similarly, the length d, i.e. the AR model order, is assumed to lay within the set $\{1, \cdots, d_{max}\}$, with the underlying assumption that all the (AR) orders are lower or equal to d_{max}. For the sake of consistency with the frontier estimate, we only form $\mathbf{Y}_d(n)$ for $n = 1, ..., N - d_{max}$ (although we could have formed $N - d$ arrays). Moreover, we also assume that all the switching lags $i_1, \cdots, i_k$ are lower or equal to d_{max} as seen in section 5. Our algorithm aims to recursively label all the array $\mathbf{Y}_d(n)$ (section 3.2) with an estimated AR model (section 3.1), the process being repeated for unlabeled arrays until that the number of unlabeled arrays N_u is below a threshold.

3.1. Estimating one AR model

As previously stated, this step takes for input the unlabeled arrays. At the beginning of our algorithm all the arrays being unlabeled, we have $N_u = N - d_{max}$. From the image processing point of view, the problem of identifying an AR model generating $Y(n)$, that is the first component of $\mathbf{Y}_d(n)$ is similar of finding R subspaces linearly binding the components of $\mathbf{Y}_d(n)$ as in eq. (1). Our problem is close to the problem of finding a plane within a point cloud for which a wide literature is available such as the RANdom SAmples Concensus (RANSAC) algorithm and its improved version [32], [33]. The idea of the RANSAC algorithm (and our algorithm) is to recursively estimate the model and discarding the outliers (points being not within the "margin of tolerance") until to find a stable solution (that is a "consensus"). Since the RANSAC is an iterative process the starting point of this algorithm requires a seed, randomly chosen as seen in the next point. However, our problem of finding a subspace, in which arrays $\mathbf{Y}_d(n)$ laid, is more complex than the RANSAC approach, since we have to find several subspaces/planes crossing at the origin since $e_r(n)$ is zero mean in eq. (1). In order to not find consensus around the origin (when randomly choosing the seed as in the RANSAC algorithm), we have searched a seed as "far" as possible from the origin i.e. the arrays $\mathbf{Y}_d(n)$ with the highest L_2 norm. In other words, we estimate a "local" standar deviation (L2-norm) of time series samples contained in of $\mathbf{Y}_d(n)$ and we select N_s arrays with the highest norm/standard deviation. This number is defined as $N_s = N_u P_{se}$. Sim-

ulations have shown that P_{se} does not have any effects on the algorithm results presented in section 5 and for this reason it was set to 0.1. The idea behind this seed initialization is, since the AR models have different variance output and since the L_2 norm is a local estimate of the standard deviation, the arrays with the largest norm are assumed to be a sequence of the output producing the largest AR output variance (possibly a non-stable AR model).
Once these N_s being selected, d arrays $\mathbf{Y}_d(n)$ are required to estimate the p_r AR model parameters (under the hypothesis that $d = p_r$ and all these d arrays belong to a same regime, that is the first component is generated with the same regime), thus verifying $\mathbf{Y}_d(n)^T(1 \quad A_r) = e_r(n)$. The parameters are obtained by Ordinary Least Squares (OLS) solution.

$$\tilde{A}_d^i = -\mathbf{M}^T(\mathbf{M}.\mathbf{M}^T)^{-1}\mathbf{Z} \quad \text{with} \quad \mathbf{M} = \begin{pmatrix} Y(n_1-1) & \cdots & Y(n_1-d) \\ Y(n_2-1) & \cdots & Y(n_2-d) \\ \vdots & \ddots & \vdots \\ Y(n_d-1) & \cdots & Y(n_d-d) \end{pmatrix} \text{ and } \mathbf{Z} = \begin{pmatrix} Y(n_1) \\ Y(n_2) \\ \vdots \\ Y(n_d) \end{pmatrix} \tag{2}$$

where $(n_1, \cdots, n_d) \in \{1, \cdots, N_s\}^d$ are integers verifying $n_i \neq n_j$ (for the sake of inversion of $\mathbf{M}.\mathbf{M}^T$). Then, we obtain $N_a = \binom{N_s}{d}$ estimates denoted $\tilde{A}_d^i$ in eq. (2) with $i = \{1, \cdots, N_a\}$. Except the intrinsic variance of the OLS estimates, the error on the $\tilde{A}_d^i$ estimates is also induced :

- First, by a set of arrays $\mathbf{Y}_d(n_1), \cdots, \mathbf{Y}_d(n_d)$ not all belonging to the same regime. Thus, in this case, arrays $\tilde{A}_d^i$ are randomly spread through the space of dimension d, while when the arrays in the OLS of eq. (2) belong to the same regime, $\tilde{A}_d^i$ is close to the parameters of the AR model generating this first component of $\mathbf{Y}_d(n_1), \cdots, \mathbf{Y}_d(n_d)$. Thus, the next step involves detecting clusters (using a clustering algorithm) of $\tilde{A}_d^i$ as seen below.

- Second, by the bad conditioning of matrix $\mathbf{M}.\mathbf{M}^T$ when several arrays $\mathbf{Y}_d(n_1), \cdots, \mathbf{Y}_d(n_d)$ are close to each other (when $n_1, \cdots, n_d$ are close lags for instance). Then, the matrix $\mathbf{M}.\mathbf{M}^T$ is close to be non invertible and leads to outlier $\tilde{A}_d^i$ estimates. These outliers disturb the convergence of the clustering algorithm since the L_2 norm is involved in the objective function of the Fuzzy-C means algorithm (see below eq. 3). Thus, these outliers have to be discarded before performing the clustering algorithm

The clustering is performed using a Fuzzy-C mean algorithm since we have to find a centroid within the $\tilde{A}_d^i$ estimates, the number of these arrays close to this

centroid having to be the highest possible. The choice of a fuzzy approach avoids hard decisions during the clustering that can be fairly not robust for points randomly and continuously spread through the space $\mathbb{R}^d$ as stated above. To discard the outliers, the arrays $\tilde{A}_d^i$ are ranked according to their increasing L_2 norm, their rank being denoted $r(i)$. The idea is that the norm of $\tilde{A}_d^i$ growths linearly for AR coefficients estimated with well-conditioned matrix whereas the outliers exhibit strong departure from this "smooth" growing (badly conditioned matrices with small eigenvalues). In order to define a criterion to consider an array as an outlier, first a linear regression of the array norm, between the smallest array, i.e. $r(i) = 1$, and the median array $r(i) = N_a/2$ (in order to keep at least the half of the estimated arrays). Thus, we estimate the norm residual, for $1 \leq r(i) \leq N_a/2$, defined as the difference between the array norm and the linear regression :

$$\zeta(r(i)) = ||\tilde{A}_d^{r(i)}|| - \frac{r(i)-1}{N_a/2-1}\Big(||\tilde{A}_d^{N_a/2}|| - ||\tilde{A}_d^1||\Big) - ||\tilde{A}_d^1|| \tag{3}$$

The norm residual variance is estimated as:

$$\hat{\sigma}_\zeta^2 = \sum_{r(i)=1}^{N_a/2} \zeta^2(r(i)) - \Big(\sum_{r(i)=1}^{N_a/2} \zeta(r(i))\Big)^2 \tag{4}$$

For possibly outlier arrays, having a norm higher than the median array, i.e. $N_a/2 < r(i) \leq N_a$, we also calculate the norm residual as in eq. (3). The arrays, verifying $|\zeta(r(i))| > 3\hat{\sigma}_\zeta$, are declared to be outliers and are not considered as Fuzzy-C mean algorithm input. This threshold of $3\hat{\sigma}_\zeta$ is derived from the Gaussian distribution and corresponds to the 5-95% quantile. Thus, we perform the Fuzzy-C mean clustering on the $\bar{N}_a$ ($N_a/2 \leq \bar{N}_a < N_a$) remaining arrays and obviously, $\bar{N}_a$ depends on d. The Fuzzy-C mean algorithm aims to recursively calculate the parameters minimizing the objective function $\underset{c_j,\omega_{i,j}}{\text{argmin}} \sum_{j=1}^{N_c}\sum_{i=1}^{\bar{N}_a} \omega_{i,j}^m ||\tilde{A}_d^i - c_j||$ where c_j is a centroid, N_c the maximum number of centroids (in what follows we set $N_c = 3$, the value of this parameter having not any effects over the algorithm results) and m a sharpness parameter, (the value of this parameter has to be discussed) strictly greater than 1. $\omega_{i,j}$ is the belonging measure of estimate $\tilde{A}_d^i$ to cluster j defined as :

$$\omega_{i,j} = \left(\sum_{k=1}^{N_c} \frac{||\tilde{A}_d^i - c_j||}{||\tilde{A}_d^i - c_k||}\right)^{-2/(m-1)} \tag{5}$$

Under the restriction $m > 1$ we observe that $\omega_{i,j}$ is within $[0, 1]$. Obviously, when an array has a belonging measure close to 1, it means that the array is close to the centroid. The centroid is updated using

$$c_j = \sum_{i=1}^{\bar{N}_a} \omega_{i,j}^m \tilde{A}_d^i / \sum_{k=1}^{N_c} \omega_{i,k}^m \tag{6}$$

As previously mentioned, our problem is to detect whether it exists a homogeneous subset around a centroid. For this, we estimate the rate of arrays verifying $\omega_{i,j} > T_B$, T_B being a threshold to be also set. When a high number of arrays verifies this property, it means that a cluster exists and its centroid is (possibly) the AR coefficients of one of the regimes of eq. (1). After testing all the values, $d = \{1, \cdots, d_{max}\}$, we obtain $N_c.d_{max}$ rates, for the N_c centroids obtained for the tested values of d. The centroid with the highest rate of arrays is identified as the AR coefficients of one of the regimes (in particular the order p_r is given by the dimension of the centroid). A sum up of this section is given in algo. 1.

input : N_u unlabeled arrays $\mathbf{Y}_d(n)$
output: r^{th} AR model coefficients
for $d \leftarrow 1$ **to** d_{max} **do**
  Select the $N_s = N_u.P_{se}$ arrays with the highest norms;
  for $n_1 \leftarrow 1$ **to** N_s,$i_2 \leftarrow 1$ **to** N_s, $n_2 \neq n_1$,...,$n_d \leftarrow 1$ **to** N_s, $n_d \neq n_1$,..., $n_d \neq n_{d-1}$ **do**
    $\tilde{A}_d^i \leftarrow$ `OrdinaryLeastSquare` $(\mathbf{Y}_d(n_1), ..., \mathbf{Y}_d(n_d))$
  end
  $r(i) \leftarrow$ `Rank` $(\tilde{A}_d^i)$ /* L_2 *norm*/;
  for $r(i) \leftarrow 1$ **to** $N_a/2$ **do**
    `Select` $(\tilde{A}_d^{r(i)})$;
    $\zeta(r(i)) \leftarrow$ `ResidualEstimation` $(\tilde{A}_d^{r(i)})$;
  end
  $\hat{\sigma}_\zeta \leftarrow$ `StdEstimation` $(\zeta(1), \cdots, \zeta(N_a/2))$;
  for $r(i) \leftarrow N_a/2 + 1$ **to** $N_a/2$ **do**
    $\zeta(r(i)) \leftarrow$ `ResidualEstimation` $(\tilde{A}_d^{r(i)})$;
    if $|\zeta(r(i))| < 3\hat{\sigma}_\zeta$ **then**
      `Select` $(\tilde{A}_d^{r(i)})$;
    end
  end
  $C_1^d, ..., C_{N_c}^d, R_1^d, ..., R_{N_c}^d \leftarrow$ `FuzzyCmeans` $(\tilde{A}_d^1, ..., \tilde{A}_d^{\bar{N}_a}, T_b, m)$,
end
Return $A_r = C_j^d$ with the highest R_j^d for $d = 1, ..., d_{max}$ and $j = 1, ..., N_c$

Algorithm 1: One AR model estimation

3.2. Labelling the arrays

To identify the arrays fitting the AR model estimated in the previous point, we estimate the residuals, that is the difference between $Y(n)$ and its estimated value using AR model. After estimating the residual pdf (using a Kernel Density Estimate, KDE, method [34]), we focus our attention on the pdf mode (maximum) the closest to zero. In fact, a residual close to 0 means that the estimated AR model fits the array $\mathbf{Y}_d(n)$ and inversely a discrepancy between the AR model and the array can lead to a residual outlier (pdf tails). Since there is not any hypothesis concerning the innovation law, as stated in introduction, a method to identify the outlying residuals (and thus the arrays that do not fit the model) has to be designed.

For this, the slopes on each side on the central mode are estimated. A linear regression is performed over a sliding window of (five points in our case). The windows (for each side) leading to the smallest Mean Square Error for the regression give the slope of the two sides of the central mode (obviously the slopes can be different when the innovation is skewed for instance, see section 5). The crossing of the two slopes with the x-axis gives the two thresholds (see figure 1). When the residuals are within the two thresholds (x^+, x^-) range, the corresponding array is labelled according to this currently estimated AR model. The section is summed in algo (2)

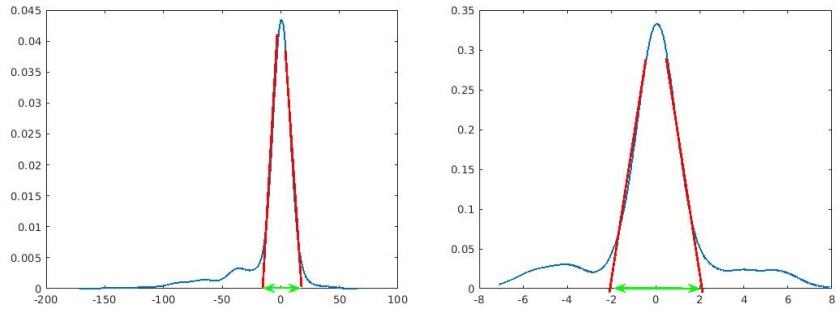

Figure 1: Example of array labelling according to their residual value (green double arrow) Exponential innovation (right) Gaussian innovation (left)

3.3. Estimating the SETAR model

Obviously to fully estimate the SETAR model, the two previous points have to be repeated until that the number of not labelled arrays N_u is below a threshold given by $(N - d_{max})P_{st}$ with P_{st} the rate of unlabeled arrays (see table 2) or the number of regimes reaches R_{max}. As detailed in section 5, this

input : The r^{th} estimated AR model (denoted A_r)
output: Labeled/unlabeled time series lags
for $n \leftarrow 1$ **to** N_u **do**
  $\zeta(n) \leftarrow$ `ResidualEstimation` $(\mathbf{Y}_d(n), A_r)$;
end
$\hat{f}_\zeta(x) \leftarrow$ `PdfEstimate` $(\zeta(n))$;
Detect x_m the pdf mode the closet to $x = 0$;
while $\frac{d\hat{f}_\zeta(x)}{dx} < 0$ *and* $x > x_m$ **do**
  $MSE_+(x)$=`SlopeEstimation` $(\hat{f}_\zeta(x), \hat{f}_\zeta(x + \Delta_x))$,x++;
end
Select the slope with the minimal $MSE_+(x)$ and calculate its crossing x^+ with the x-axis.;
Same for $x < x_m$, the x-axis crossing being denoted x^- ;
for $n \leftarrow 1$ **to** N_u **do**
  if $x^- < \zeta(n) < x^+$ **then**
    $l(n) = r$
  end
end

Algorithm 2: Array labelling

parameter has some effects on the final algorithm results. In fact, a too large value of P_{st} can stop the SETAR identification process without identifying all the regimes while a too low value leads to identify additional and spurious AR systems. Obviously, the recursive identification of the AR models can lead to re-identify previously identified AR models. In order to really identify a new AR model, a similarity criterion between the currently estimated AR model and the previously identified ones is calculated as :

$$||A_i - A_r|| \leq K \frac{||A_i|| + ||A_r||}{2} \tag{7}$$

where A_i is a previously identified AR model and A_r, the currently estimated AR model (center of the selected centroid) and K a constant that we have chosen equal to $K = 0.2$ in section 5 (other values have also been tested as briefly discussed in this section). If the criterion (7) is verified, then the two AR models are merged, the labelled arrays derived under the redundant AR model are re-labelled with i.

3.4. Relabeling the arrays

When all the AR models are assumed to be identified i.e. $N_u \leq (N - d_{max})P_{st}$, all the arrays are finally relabeled according to the estimated AR

models leading to the smallest residuals. This last step of relabeling ensures first that each array is bound to its closest AR model among all the identified AR models (correction of possible mislabeling) and second to label the remaining unlabeled arrays of the estimation basis. The labelling is also performed over the validation basis to validate the estimated SETAR model as seen in section 4.2.

input : Time series samples
output: The AR models and the switching condition
Separate the times series into the estimation base and the validation base $N_u = N - d_{max}$ N_u *the number of unlabeled arrays*, r=1;
while $N_u > NP_{st}$ **do**
  $A_r \rightarrow$`OneARModelEstimation` $(Y(1),..,Y(n),l(1),..,l(n))$ */* algo 1 */*;
  `ArrayLabelling` $(Y(1),..,Y(n),l(1),..,l(n),A_r)$*/* algo 2 */*;
  for $i \leftarrow 1$ **to** $r-1$ **do**
    if *eq (7) is verified for* A_i *and* A_r **then**
      $A_r = A_i$ and all the lag labels equal to r are set to i ;
    else
      r++;
    end
  end
end
Relabel all the arrays $\mathbf{Y}_d(n)$ according to the estimated AR model

Algorithm 3: General Overview

4. Switching variables and function identification and model validation

4.1. Switching variables and function

The second step has to identify the variables (lags) involved in the switching of regimes as well as the function governing this switching $g()$ and finally validating the estimated model. After the (final) labelling of the previous section, we have to find one or several frontiers separating the labelled arrays $\mathbf{Y}_{d_{max}}(n)$ (in what follows we simply denote $\mathbf{Y}(n)$). We consider $\mathbf{Y}_{d_{max}}(n)$, since we search the indices $i_1, \cdots, i_k$ involved in the switching condition, that are possibly higher than the AR orders. A useful tool for this kind of separation problem is the Support Vector machine (SVM) approach. We first consider the case when only one frontier has to be found between only two identified AR models and discuss the more general case at the end of the section. The regime labels $l(n)$

are re-indexed to 1 and -1. The SVM aims to define a frontier that is to classify

$$\underset{W,b,\varepsilon(n)}{\operatorname{argmin}} \frac{1}{2} W^T W + P \sum_{n=1}^{N-d_{max}} \varepsilon(n) \quad \text{subject} \quad l(n)\Big(W^T \Phi\big(\mathbf{Y}(n)\big) + b\Big) \geq 1 - \varepsilon(n) \quad \varepsilon(n) \geq 0 \tag{8}$$

In this cost function, P is a penalty parameter and $\varepsilon(n)$ a random variable allowing mislabeling (as in our case see section 5). W is the array describing the frontiers that is the weight of the components of array $\mathbf{Y}(n)$ involved in the switching, better said the lags $i_1, \cdots, i_k$ (a null weight induces that the variable does not have any effects on the regime switching). b a "bias" giving the threshold for the switching, thus defining the set B_i in eq. (1). $\Phi()$ is nonlinear transform allowing to design complex (nonlinear) frontier shapes. In fact, if $\Phi()$ is the identity, then the problem of separating the arrays according to their label is linearly separable (the usual cases in the aforementioned papers on SETAR identification). But more complex shapes (circles or ellipses) can be estimated as seen in section 5. Since in practice, eq. (8) is minimized using its dual problem, the key point for the frontier shape, is the SVM kernel defined as $K(n,p) = \Phi(\mathbf{Y}(n)).\Phi^T(\mathbf{Y}(p))$. The kernel function has to be chosen before performing the SVM classification and thus the drawback of our approach is "to guess" the shape of $g()$ (and thus that of $K(,)$). Two kernels have been tested, a linear kernel and a Radial Basis Function (RBF) kernel, the finally selected kernel (and frontier) being validated by the highest rate of well labelled arrays over the cross validation basis as seen in section 4.2. This process is straightforwardly generalized to a higher number (than 2) of AR models by recursively estimating the frontiers between one label against all the others. Once this first frontier is retrieved, the process is repeated, selecting one remaining label against all the others. Thus, the areas/subspaces defined by the estimated frontier(s) are labelled according to the corresponding array label. This label is used in the validation step as seen in the subsection.

4.2. Model validation

After determining the frontiers, we estimate the rate of well classified arrays over the validation basis, that is the array label is the same as that of the area/subspace within it lays (using the estimated frontiers of section 4 and the array labelling of section 3.4). Obviously the estimated model is validated when this rate/probability is higher than a given threshold (confidence level in the

Notation	Name	Tested values
P_{se}	Rate of points in the seed	10 %
m	C Fuzzy Sharpness	1.05 and 1.5
N_c	Number of clusters	3
T_B	Belonging measure threshold	0.7 and 0.9
R_{max}	Maximum of regime number	4
d_{max}	Maximum AR order	3 and 4
P_{st}	Rate of unlabeled $\mathbf{Y}_d(n)$ arrays	5 %, 2%

Table 2: Hyperparameter definition and tested values

hypothesis testing framework). We discuss the value of this threshold in the section 5.

5. Numerical simulations and results

In this section, we detail the results obtained on simulated SETAR models. As stated in section 2, the performance can be derived first with regards to the algorithm hypermeters and second to the parameters of the SETAR model generating the observed time series. In order to completely detail these results, we have devoted a first subsection to the SETAR model experiment definition, a second to the algorithm performance metrics, the algorithm performance analysis being exposed in a third subsection.

5.1. Experiment definition

In table 3, we have displayed the AR model parameters of three SETAR models. The first regime of model 1 and model 2 and the second regime of model 3 have one real root outside the unit circle (unstable regime). For each experiment, we have simulated 1000 time series of $N_T = 1024$ samples (thus $N = 768$ and $N_v = 256$). Results presented in this section are statistics of the results of our algorithm obtained over these 1000 time series. We have performed several sets of SETAR identification experiments in order to analyze the hypermeters effects d_{max}, T_B, m P_{st} (as seen in table 2). The SETAR model parameters have obviously some effects on the algorithm performance (although this point is barely tackled in the papers devoted to the SETAR identification). Thus, we have also estimated the performance algorithm with different thresholds values, switching variables and governing switching condition $g()$, innovation $e_r(n)$ variance and skewness (table 4). In this table, the last two/three columns give the estimated mean rate (over the 1000 times series) of samples generated by each regime. The details of the different experiments are :

	Regime	a_1	a_2	a_3
Model 1	1	1.3	-0.9	1.3
	2	-1.6	0.9	0
Model 2	1	1.3	0.9	0
	2	-1.	0.7	0
Model 3	1	-1.6	0.9	0
	2	1.3	-0.7	1
	3	1	0.4	0

Table 3: AR coefficients of tested SETAR models

-For the three models, exp 1 is a standard model, with a governing function $g(Y(n-n_1),\cdots,Y(n-n_{d_{max}}))$ simply equal to $Y(n-n_1)$, a Gaussian innovation variance equal to 0.5 (usual variance in the aforementioned papers) and a threshold value (see subsets B_1, B_2, B_3 in table 4) inducing that all the regimes are significantly present (for model 1, it was not possible to find a threshold in order to have a higher rate than 33 % for the first regime). This experiment can be considered as a benchmark (optimal conditions) for the other experiments exposed in the next points. In exp 2, the switching threshold is shifted and one regime overwhelms the other(s) as seen in the last columns of table 4. The aim of this second experiment is to estimate the capability of our algorithm to estimate a SETAR model having a behavior close to an AR model.

-With the same switching condition, we simulated (exp 3 and 4) time series with an innovation/residue variance of 0.1 and 1 in order to estimate the effects of this parameter on our algorithm. In fact, a high variance hides the linear (AR) structure between the times series samples and then turns the identification model difficult while a small innovation variance "gathers" arrays $\mathbf{Y}_d(n)$ around the origin whatever the regime and thus disturb the seed initialization (see section 3.1). Similarly, the innovation skewness leads to time series sample outliers that also disturb the seed initialization. Thus, in exp 7, we have tested an innovation with a (centered) exponential distribution of variance and skewness equal to 0.5 and 2 respectively.

-We also tested a model with an innovation variance depending on the regime exp 8 and 9 (for model 1 and 2). The first (resp. second) number in the column variance indicates the variance of the first (resp. second) regime.

-In the previously described experiments the switching condition is $g(Y(n-n_1),\cdots,Y(n-n_{d_{max}})) = Y(n-n_1)$ below or above a threshold. We have tested more complex linear functions (that is a separable problem for the SVM kernel see section 4), $g(Y(n-n_1),\cdots,Y(n-n_{d_{max}})) = Y(n-n_1)+0.4Y(n-n_3)$

Model	Exp.	B_1	B_2	B_3	σ_e^2	Law	% of 1	% of 2	% of 3
1	1	$(-\infty,-4)$	$[-4,+\infty)$	-	0.5	N	33	67	-
	2	$(-\infty,2)$	$[2,+\infty)$	-	0.5	N	12	88	-
	3	$(-\infty,-4)$	$[-4,+\infty)$	-	1	N	33	67	-
	4	$(-\infty,-4)$	$[-4,+\infty)$	-	0.1	N	33	67	-
	5	$(-\infty,-4)$	$[-4,+\infty)$	-	0.5	N	29	70	-
	6	$[0,20)$	$[20,+\infty)$	-	0.5	N	22	78	-
	7	$(-\infty,-4)$	$[-4,+\infty)$	-	0.5	E	38	62	-
	8	$(-\infty,2)$	$[2,+\infty)$	-	0.5/1	N	33	67	-
	9	$(-\infty,2)$	$[2,+\infty)$	-	1/0.5	N	33	67	-
2	1	$(-\infty,5)$	$[5,+\infty)$	-	0.5	N	49	51	-
	2	$(-\infty,1)$	$[1,+\infty)$	-	0.5	N	17	83	-
	3	$(-\infty,5)$	$[5,+\infty)$	-	1	N	48	52	-
	4	$(-\infty,5)$	$[5,+\infty)$	-	0.1	N	48	52	-
	5	$(-\infty,0.5)$	$[0.5,+\infty)$	-	0.5	N	49	51	-
	6	$[0,5)$	$[5,+\infty)$	-	0.5	N	29	71	-
	7	$(-\infty,-5)$	$[-5,+\infty)$	-	0.5	E	49	51	-
	8	$(-\infty,5)$	$[5,+\infty)$	-	0.5/1	N	50	50	-
	9	$(-\infty,5)$	$[5,+\infty)$	-	1/0.5	N	50	50	-
3	1	$(-\infty,-0.5)$	$[5,+\infty)$	$[-0.5,5)$	0.5	N.	37	32	31
	2	$(-\infty,-0.5)$	$[0,+\infty)$	$[-0.5,0)$	0.5	N	14	75	11
	3	$(-\infty,-0.5)$	$[5,+\infty)$	$[-0.5,5)$	1	N	31	43	26
	4	$(-\infty,-0.5)$	$[5,+\infty)$	$[-0.5,5)$	0.1	N	31	35	35
	5	$[0,2)$	$[2,18)$	$[18,+\infty)$	0.5	N	14	51	35
	6	$(-\infty,-0.5)$	$[5,+\infty)$	$[-0.5,5)$	0.5	E	33	43	24
	7	Nested	(see section 5.1)		0.5	N	31	48	21

Table 4: SETAR model switching functions and variables. N stands for normal and E for centered exponential.

for model 1 and $g(Y(n-n_1),\cdots,Y(n-n_{d_{max}})) = -Y(n-n_1)+Y(n-n_3)$ for model 2 in experiment 5. In these cases, the separation problem is always linear and thus a linear kernel $K(.,.)$ is well suited (see section 4). A nonlinear switching variable (non separable problem) $g(Y(n-n_1),\cdots,Y(n-n_{d_{max}})) = Y^2(n-n_1)+Y^2(n-n_2)$ has been used for experiment 6 of model 1 and 2 and 5 of model 3, as seen on the sets B_1, B_2, B_3 is table 4 are only defined on $\mathbb{R}^+$. A RBF kernel is optimal for this second kind of nonlinear switching condition. Finally, in the three regime model case, a nested switching condition has also been tested. The switching condition is that the regime 1 is selected when $Y(n-n_1) < -13$, the selection between regime 2 and 3 being based on $Y(n-n_2)$ ($Y(n-n_2) > -3$ for regime 2 and $Y(n-n_2) \leq -3$ for regime 3).
Except the SETAR model parameters and the algorithm hyperparameters, we also inspected the time series length effects. For this we have tested shorter time series lengths of $N_T = 512$ and $N_T = 256$ samples. For the algorithm hyperparameters, we have paid a special attention on the d_{max} effects, since this parameter gives the dimension of the space, within it the AR models have to be retrieved. The higher this parameter, the higher the probability of a false AR model identification.

Model	Exp.	$P_{st} = 5\%$				$P_{st} = 2\%$			
		$m = 1.05$		$m = 1.5$		$m = 1.05$		$m = 1.5$	
		$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$
1	1	57/53	57/53	49/42	45/37	87/77	87/77	79/59	72/49
	2	59/57	59/57	58/55	52/50	75/55	74/54	74/53	68/46
	3	43/39	42/39	42/36	38/32	82/72	81/71	77/62	73/56
	4	25/22	24/21	26/24	24/22	49/40	49/40	53/43	48/38
	5	48/48	48/47	48/48	46/46	66/61	65/60	64/57	61/53
	6	55/53	55/53	42/39	22/18	70/59	69/58	55/39	32/17
	7	50/47	50/47	47/38	42/30	80/71	80/70	73/55	65/44
	8	65/60	65/60	56/48	49/42	87/75	88/75	77/55	70/47
	9	67/63	67/62	56/48	50/42	89/76	88/75	78/54	72/45
2	1	99/92	100/92	100/92	99/84	99/80	100/79	100/77	99/64
	2	25/11	24/10	31/6	28/3	26/4	26/4	32/2	29/1
	3	97/51	97/50	96/57	94/44	97/34	97/32	96/37	95/25
	4	99/96	99/96	100/96	100/91	99/88	99/88	100/85	100/76
	5	59/8	59/8	71/15	56/10	63/3	63/3	75/6	60/3
	6	4/2	4/2	4/1	2/0	6/2	5/2	5/0	2/0
	7	91/22	91/22	96/43	95/34	92/15	92/16	97/29	96/22
	8	99/83	99/82	99/86	99/76	99/64	99/63	99/65	99/50
	9	99/85	99/84	99/87	99/75	99/64	99/64	99/69	99/51
3	1	61/57	61/57	49/40	38/28	70/59	70/58	59/42	49/30
	2	0	0	0	0	0	0	0	0
	3	31/28	32/28	29/27	24/20	44/33	44/33	41/30	33/21
	4	52/50	53/50	46/39	38/31	67/61	66/60	65/49	54/37
	5	0	0	0	0	0	0	0	0
	6	22/21	22/21	19/17	15/14	36/30	35/29	31/24	27/19
	7	37/33	36/32	21/15	14/8	48/30	47/29	32/14	24/7

Table 5: Rate (%) of time series for which the AR models have been retrieved among possible spurious AR models (first number) and exactly been retrieved (second number)

Model	Exp.	$P_{st} = 5\%$				$P_{st} = 2\%$			
		$m = 1.05$		$m = 1.5$		$m = 1.05$		$m = 1.5$	
		$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$
1	1	49/48	48/48	40/39	36/35	72/71	71/70	57/55	48/45
	2	9/9	10/10	9/9	8/8	9/9	9/9	9/9	7/7
	3	36/35	35/35	34/33	29/29	65/64	65/64	59/57	52/51
	4	18/18	18/18	21/21	19/19	31/31	31/30	35/35	32/31
	5	48/48	47/47	48/48	46/46	64/60	63/60	61/57	58/53
	6	48/47	48/46	36/35	18/16	59/51	58/51	45/35	24/15
	7	32/30	33/31	33/27	29/21	42/38	42/37	44/34	42/27
	8	61/59	60/58	50/47	43/41	75/73	75/73	58/54	50/46
	9	62/61	62/61	50/47	44/42	76/74	75/73	58/53	49/44
2	1	91/91	91/91	91/91	83/83	78/78	78/78	75/75	63/63
	2	1/0	1/0	1/0	1/0	2/0	2/0	2/0	2/0
	3	2/2	2/1	2/1	2/1	1/1	1/1	2/1	2/0
	4	92/92	92/92	90/89	86/86	85/85	85/85	79/79	71/71
	5	7/0	6/0	4/0	2/0	5/0	4/0	3/0	3/0
	6	1/1	1/1	1/0	0/0	2/1	2/1	2/0	1/0
	7	52/0	52/0	22/0	23/0	53/0	53/0	24/0	26/0
	8	74/74	73/73	79/77	70/68	59/59	57/57	59/57	46/44
	9	77/77	77/77	80/78	70/68	58/58	58/58	65/63	48/45
3	1	49/46	50/47	38/34	28/23	52/47	53/48	40/34	32/25
	3	16/14	17/14	18/15	14/11	21/17	21/16	21/15	16/9
	4	22/22	23/22	20/16	14/12	28/28	29/29	24/20	17/14
	6	17/17	17/16	15/14	13/12	30/26	29/25	25/20	21/16
	7	32/31	31/30	18/15	11/8	34/28	33/27	22/13	16/7

Table 6: Rate (%) of time series for which the frontier(s) has(ve) been retrieved among possible spurious frontiers (first number) and exactly been retrieved (second number)

5.2. Performance Metrics

As the proposed algorithm, the performances metrics can be divided into three subgroups (and tables), detailed in the next three subsections, the first two being devoted to the model identification, AR models and switching conditions, and the third to the model validation.

5.2.1. AR model estimates and labelling performance

In order to estimate the AR model identification performance, we develop three metrics, having not the same importance.
-The first metric (table 5) is the rate of time series for which all the AR models have been identified, among possible spurious AR models, first number, or exactly, second number. As previously stated, an AR model is declared to be identified when the similarity criterion (between an estimated AR model and a true AR model) is verified (see eq. 7). Obviously, we consider this second number as the key performance metric (the maximum of the results is in grey cell in table 5).The second metric is the (absolute) bias and standard deviation of the AR coefficient estimates. Finally, we have estimated the number of well labelled arrays on the estimation basis, this metric being a key feature for the frontier estimation step. These last two metrics being only estimated over time series for which all the AR regimes have been retrieved, thus they are positively biased. Moreover, due to the amount of results for the AR coefficient bias and variance, we only briefly mention the results of the second metric when bringing some information.

5.2.2. Switching variables and conditions estimate

For the switching condition $g()$, we have to distinguish the case of the linearly separated case (almost all the experiments) and the nonlinear case (exp 6 for model 1 and 2 and 5 for model 3), the results of the two cases being given in table 6. For the linearly separated case (linear kernel), in order to compare the true frontier coefficients with the estimated one, we first gather W and b into a vector, that is $F = [W, b]$. W gives the weight of each past variable in the switching and b is the threshold value between B_1, B_2 and possibly B_3 (table 4). As seen section 5.1, we have $W = [1, 0, 0]$ when $d_{max} = 3$ in almost all the cases except exp 5 for which $W = [1, 0, 0.4]$ (model 1) and $W = [-1, 0, 1]$ (model 2). We have to add a last coefficient equal to 0 when $d_{max} = 4$. For the nested model 3 (exp 7). We have a first frontier $W = [1, 0, 0]$ with $b = -13$ and obviously a second $W = [0, 1, 0]$ with $b = -3$. Thus, F defines the hyperplane separating the

regimes and we denote $\hat{F} = [\hat{W}, \hat{b}]$, the estimate provided by the SVM assuming a linear kernel (see section 4). However, F is defined except a multiplicative coefficient and in order to numerically compare the frontier coefficient estimates with the true ones, we normalized the estimate by a multiplicative factor $\hat{\alpha}$ found by regressing $\hat{W}$ over W, i.e. $\hat{\alpha} = \underset{\alpha}{\text{argmin}}||W - \alpha\hat{W}||$. We use W and $\hat{W}$ instead of F and $\hat{F}$, since as seen in table 4, the values of b can be much higher than those of the coefficients of W and thus a linear regression based on the linear regression value between b and $\hat{b}$ and would lead to artificially good results. After denormalizing, the closeness of the estimated frontiers (to the true one) is estimating using criterion (7) replacing A_i and A_r by F and $\hat{F}$ (respectively). As for the AR model, a main performance metric and an ancillary one are estimated. These two metrics are first the rate of time series for which the frontier(s) is (are) retrieved (table 6), estimated only for time series for the AR regimes have been retrieved, with possible false AR model first number or exactly second number). The second metric is the bias and standard deviation of the frontier coefficients. For the same reason as for the AR model coefficients, they are not fully detailed.
For the non linearly separated case, the SVM algorithm with an RBF kernel identifies input arrays that define the frontiers between the labels, these arrays approximating a circle. Thus, we have focused our attention on the circle radius as a performance metric. In fact, the circle radius gives the switching threshold (see table 4). For exp 6 for model 1 and 2 and exp 5 for model 3, as a performance metric, we have estimated the rate of time series (given in table 6) for which the estimated radius is the within the range of the true value more or less 20 % similarly to criterion (7).

5.2.3. Model Validation metric

As stated in section 1, a cross validation approach has been chosen. In the results given below, we chosen a level of 90% of well classified arrays (over the validation basis) to validate the SETAR model (according to the AR models and frontiers estimated over the estimation basis). In the hypothesis testing framework, this rate corresponds to a *p-value* equal to 0.1. In table 7, we have gathered three error probabilities (rates), that is the rate of rejecting the validation, although the SETAR model has been correctly estimated (AR models and frontiers), the second is the rate of accepting the model although the model is not correctly identified and finally the rate of validating the SETAR model although it has been identified with a wrong frontier shape (that is an RBF

Model	Exp.	$P_{st} = 5\%$				$P_{st} = 2\%$			
		$m = 1.05$		$m = 1.5$		$m = 1.05$		$m = 1.5$	
		$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$
1	1	0/12/0	0/12/0	0/18/0	0/19/0	0/13/0	0/13/0	0/18/0	0/22/1
	2	0/65/0	0/65/0	0/64/0	0/60/0	0/74/0	0/74/0	0/76/0	0/74/0
	3	0/12/0	0/13/0	0/13/0	0/17/0	0/15/0	0/15/0	0/16/1	0/19/0
	4	0/12/0	0/12/0	0/11/0	0/13/0	0/21/1	0/21/1	0/17/1	0/19/1
	5	0/5/0	0/5/0	0/5/0	0/6/0	0/16/0	0/17/0	0/16/0	0/20/0
	6	46/6/0	45/7/0	34/9/0	16/14/0	51/14/0	50/15/0	33/22/0	15/24/0
	7	0/21/0	0/21/0	0/19/1	0/23/1	0/40/1	0/39/2	0/32/2	0/34/3
	8	0/8/0	0/9/0	0/16/0	0/21/0	0/11/0	0/11/0	0/19/0	0/22/0
	9	0/8/0	0/10/0	0/17/0	0/22/0	0/10/0	0/11/0	0/18/1	0/24/1
2	1	11/4/4	11/3/4	11/3/4	11/6/8	9/9/11	11/9/11	10/11/12	8/14/18
	2	5/15/0	5/15/0	3/14/1	1/15/1	2/11/0	2/12/0	1/13/0	0/13/1
	3	6/19/12	6/20/13	6/17/11	5/21/16	4/21/17	4/21/17	4/22/17	3/26/22
	4	18/1/1	18/1/1	23/1/2	22/3/4	16/5/6	17/4/6	21/4/8	18/8/13
	5	-/0/0	-/0/0	-/1/0	1/1/0	-/0/0	-/0/0	-/0/0	-/0/0
	6	2/2/0	2/2/0	1/3/0	0/1/0	2/4/0	2/3/0	0/4/0	0/2/0
	7	3/32/6	2/30/5	7/23/7	5/29/8	1/31/7	2/31/5	5/26/11	3/30/11
	8	7/8/7	7/8/7	6/6/6	6/10/11	5/14/16	6/12/17	6/13/17	3/17/24
	9	7/7/6	8/7/7	7/4/4	6/8/10	6/14/15	6/14/15	6/9/12	4/16/20
3	1	46/0/0	47/0/0	34/0/0	23/0/0	47/0/0	48/0/0	34/0/0	25/0/0
	2	0/83/0	0/82/0	0/82/0	0/82/0	0/86/0	0/85/0	0/85/0	0/84/0
	3	0/83/0	0/83/0	0/81/0	0/84/0	0/81/0	0/82/0	0/82/0	0/87/0
	4	0/76/0	0/76/0	0/82/0	0/87/0	0/71/0	0/70/0	0/79/0	0/85/0
	6	17/0/0	16/0/0	14/0/0	12/0/0	26/0/0	25/0/0	20/0/0	16/0/0
	7	31/0/0	30/0/0	15/0/0	8/0/0	28/0/0	27/0/0	13/0/0	7/0/0

Table 7: Rate (%) of rejecting although correctly identified SETAR model, (first number), of validating a wrongly identified model and finally to validate the SETAR model with a wrong kernel (a linear kernel instead of an RBF kernel and inversely)

model for a linearly separated case and inversely). Obviously, the probability of validating a correctly identified model is the complement to 1 of the first error probability.

5.3. Result analysis

In order to provide a complete analysis of our results, we have divided this section into a subsection dealing with the SETAR parameter effects and another to the algorithm hyperparameter effects, but we begin with a subsection devoted to the general remarks on our algorithm results.

5.3.1. General remarks

A first remark on the algorithm is that for all the times series, the order of identification of the AR models is always the same. For model 1 and 2, the second AR model, the stable one, is firstly identified. A possible reason is that similar array patterns occur more frequently for stable regimes. Similarly, the first regime is firstly identified for model 3, whilst there is not a special order for the last two regime identification.

However, for model 1 and 2, the bias (between 0.01 and 0.1) and the standard

deviation (between 0.02 and 0.6) are generally higher for the coefficients of the second (stable) AR model. A possible reason is that, the OLS of eq. (2) is more frequently performed with arrays generated by different regimes at the beginning of the algorithm, thus inducing slight departures of the AR coefficient estimates (from true ones) and thus a higher variance of the extracted centroid (AR model coefficients). Once a first set of arrays has been labelled and discarded for the following AR model estimation, the OLS is calculated over more homogeneous sets and the centroid (AR coefficients) variance is reduced. However, this observation is not retrieved for model 3, for which the bias and the standard deviation is the lowest for the first identified AR model. The AR coefficient bias and variance depend on the experiment and on the SETAR model. Obviously the more complex is the experiment/model, the higher are the bias and the variance. For instance, the bias and the variance are the highest in the case of exp 6 (nonlinear switching condition) for model 1 and 2 and for all the experiments of model 3 (compare to the other two SETAR models).

The ancillary metric of the well-classified arrays is about 98% for model 1 (all experiments). We also obtain this rate for model 2, except a decrease to 92% for exp 2, 88% for exp 5 and 93% for exp 6. For model 3, the rates are between 90% and 96%, due to this last model complexity. For this reason, a validation rate of 90% of well classified arrays over the validation basis is a realistic value. However, this validation rate could be adjusted to the number of identified regimes since obviously the rate of well classified arrays decreases with the number of regimes.

The bias (after normalization) of the frontier coefficients $\hat{F}$ (see section 5.2.2) is about 0.1-0.5 and the standard deviation between 0.03 and 0.15 for the first model, without any differences between the experiment results. For the second model, the bias and the standard deviation are between 0.03 and 0.3 (the highest values being obtained for $\hat{b}$ as for model 1). Model 3, with a bias (between 0.01 and 0.6) and a standard deviation (between 0.02 and 0.4) has higher frontier coefficient deviations than the first two models. Concerning the validation metrics table 7, we observe that the first two types of error, rejecting a correctly identified model and validating a wrong model (but with the good SVM kernel shape) are the highest probability errors. In particular the validation of an identified SETAR model with a wrong SVM kernel is null (model 1 and 3) or almost null (model 2). This result validates our approach of choosing *a priori* kernel/frontier shape and then validating this choice *a posteriori*

5.3.2. SETAR model effects

In this second subsection, we consider the effects of the SETAR model parameters.

The first experiment is a standard SETAR model, similar to those simulated in the aforementioned papers on SETAR identification. We can observe that the proposed algorithm leads to fairly satisfactory results (according the lack of hypotheses on the SETAR model in our algorithm) since we have up to 71% (model 1) and 91% (model 2) of well identified SETAR models (table 6 second number). Obviously, results for model 3 are less good with a highest rate of 48%. Intermediary results (concerning only the AR model identification table 5) are much higher than these rates since the AR models are retrieved up to 87%, 100% and 70% for model 1, 2, 3 respectively, thus showing that the main discrepancy lies in the frontiers retrieving step (SVM algorithm). For this first experiment, when all the AR models have been retrieved, we have about 98-99% of well labelled arrays (over the estimation basis) for model 1 and 2 and 92% for model 3 (with a standard deviation about 1% over the 1000 times series). It means, that even if spurious AR models are identified, there are very few arrays labelled with these spurious systems (over the two bases). The validation metrics (table 7) are slightly different for the three models. For model 1, the highest error probability is that of validating a false model. As previously stated, when a false AR model is retrieved, they are few sample arrays labelled with this spurious AR model and then the model is validated although the SETAR model is not exactly identified since the weight of the spurious AR model is weak. The second model shows more possible sources of errors. In particular, the rate of rejection of well identified SETAR models is about 10% of the time series (while it is null for model 1) and is the highest error probability (possibly due to closer residuals, the two AR model having the same order). For the three AR model (all experiments), the highest error probability (about 80%) is the error of rejecting a well identified model due to the higher number of regimes that increases the possibility of mislabeling over the cross validation basis.

The overwhelming of one regime (over the other(s)) disturbs the SETAR model estimates of our algorithm (but it could also disturb the algorithms proposed in papers of table 1 that have not been tested on this difficult case). For instance, the maximum rate of AR model retrieval is 59% (model 1) and 25 % (model 2), exp 2 table 5, but with a final SETAR identification, rate of 10.1% (model 1) and 0% (model 2), exp 2 table 6. For model 3, our algorithm did not re-

trieve the three AR models (this SETAR model is not discussed below). When the algorithm aims to identify an AR model with arrays corresponding to the non dominant regime, for which few sample arrays are available, it generally identifies one or several spurious AR models, leading to these weak results, in particular for model 2. The rate of well classified arrays remains close to the optimal for model 1 but decreases to 92% model 2, this discrepancy explaining the final weak results of SETAR identification for this model. Moreover, we observe that the overwhelming of one regime also disturb the frontier estimate (compare model 1 exp 2 in table 5 and 6) since we have to find a frontier for a class with few samples (i.e. labelled with the non dominant regime). In this case, the bias and the standard deviation are multiplied by 2 or 3 for the frontier coefficients with regards to exp 1. The highest probability error is the validation of a false model (60 % for model 1 best result). In fact, the rate of well labelled arrays corresponds to the dominant regime rate, which is close to the acceptation level, this inducing a model validation even if the second (non dominant) AR model is not retrieved.

As expected the innovation variance has some effects over the SETAR identification results. A higher innovation variance (of 1) dwindles the rate of the AR retrieval for model 2 and 3 and thus also decreases the rate of the SETAR model identification (exp 3 tables 5 and 6). This decreasing is less sensitive for model 1. Although the bias and standard deviation of the AR coefficients are close to those of exp 1 for the three models, the rate of well classified arrays downs to 96% (model 2) and 90% (model 3), but remains unchanged for model 1 (98%). These slightly lower rates induce a higher frontier coefficient estimate variance and have a strong effect on the optimal results (compared exp 1 and exp 3 for these two models in table 6). As expected, the probability of false validation increases for model 2 and 3 due to the higher innovation/residual variance, with limited effects for model 1 as previously stated. On the other side, a small innovation variance (of 0.1) increases the performance of our algorithm for model 2 and 3 for the AR model retrieval, but finally decreases the SETAR identification rate for model 3 (see exp 4). In this three AR model case, this dwindling is explained by the fact that the samples arrays being closer, this turns the frontiers estimate more difficult when two frontiers have to be retrieved (only 43 %). For model 1, it seems that the AR identification is also more difficult, thus leading to a lower SETAR identification rate (see table 6 exp 4) although the rate of well classified arrays over the estimation basis remains unchanged (98%). The small innovation variance has few effects on the validation performance since

we retrieve close results to those of exp 1 for model 1 (although the rate of well estimated SETAR models is about 39 %) but it increases the probability of false rejection for model 2 and strongly increases the false acceptation for model 3.
The innovation skewness (exp 7 for model 1 and 2 and exp 6 for model 3) has some effects on the identification results. For model 2, the rate of exactly identified SETAR models is divided by 2 with regards to the AR retrieval rate (compare exp 7 model 2 in tables 5 and 6) as well as for model 3. In fact, the rate of time series for which all the AR models (among spurious ones) have been retrieved is twice the rate of time series for which only the true AR models are identified (see exp 7 in table 5). For model 1, the AR retrieval rate is unchanged (compare exp 1 and 7 in table 5) while the SETAR model identification rate is divided by 2, (compare exp 1 and 7 in table 6). In this case, the rate of well labelled arrays in the experiments 1 and 7 remains close (about 98%) and the reason of the low performance of the frontier retrieval is owned to a higher variance of the mislabeled array location (induced by the innovation skewness). By loosening the array closeness constraint, $K = 0.4$ instead of 0.2 in criterion (7) for frontier validation, the SETAR identification rate fairly increases. Similar conclusions to those of model 1 can be drawn for model 3 (exp 6). The validation metrics are unchanged for model 1 and 2 (with regards to exp 1) and the false rejection probability decreases for model 3.
The switching condition also interferes on the SETAR identification results (exp 5), even if the rates of each regime are well balanced similarly to exp 1. For model 1, we observe a slight decrease (70% to 60%) of the identified AR model rate, but without any effects on the final SETAR identification rate (compare exp 5 in table 5 and 6 for model 1). For this model, the AR coefficient bias and standard deviation, the well labelled array rate, the frontier coefficient bias and standard deviation and the validation metrics remain close to those of exp 1 (and thus our algorithm is efficient to retrieve models with more complex switching condition than those based only on one past lag). On the other side, for model 2, the AR models are identified for 71% of the time series, but the rate of time series without spurious AR models decreases to 16% and the correctly estimated SETAR model rate is almost null. Moreover, for this model, this experiment exhibits a higher bias of the AR coefficients, a weaker well-labelled array rate (88%) and a higher bias and standard deviation of the frontier coefficients, leading to a nearly null SETAR model identification rate. A possible reason is that the arrays of a same regime are closer than in exp 1, since the switching variable is a linear combination of time series past values as well as

the predicted time series value $Y(n)$, thus reducing the possible range value of $Y(n)$ and finally inducing that the arrays $\mathbf{Y}_d(n)$ are closer. This closeness leads to a bad matrix conditioning in the OLS (eq. 2) and then a higher spreading of $\tilde{A}_d^i$ and turning the centroid/AR coefficient extraction more difficult.

The nonlinear switching condition also leads to different results depending on the model. For model 1, the AR retrieval rate as well as the final SETAR identification are fairly good (59% and 51% respectively), papers of table 1 being not able to identify this SETAR model. On the other side, these two rates are close to null for model 2 and 3 (for the reason that switching variable and predicted value are more linked). For model 1, we observe that our validation metric never validates a model estimated with a linear kernel (see exp 6 in table 6) for the nonlinear switching condition. However, we observe a strong probability of false rejection for model 1 (the results for model 2 being not meaningful).

For the non equal innovation variance of the two regimes, for model 1 and 2 (exp 8 and 9), we do not observe any significant dwindle of the results for the AR retrieval rate (compared to exp 1) as well as for the well labelled array rate. In particular, the order of the AR model identification (first the stable model) is unchanged for these two SETAR models whatever the innovation variance is. However, the SETAR identification rate decreases about 15% for model 2 and 3, slightly increases for model 1 (not significantly). A deeper insight shows that the reason of this decrease for model 2 lies in the SVM step, since the frontier coefficient bias and variance are twice those of exp 1. In fact, the arrays are more gathered (small innovation variance) for one regime and spreader (high variance) for the other, inducing that an accurate frontier estimate is more difficult.

Finally, the nested SETAR model (exp 7 model 3) also leads to fairly good results, since we obtain a rate of 33% AR model retrieval and 31% of SETAR identification, thus showing that this kind of model can be identified by our algorithm (unlike the algorithms listed in table 1). However, for this model, AR bias and variance of the second and third AR model are higher than in other experiments with model 3. Moreover, this experiment exhibits the highest rate of well-classified arrays (96%) and the lowest bias and variance for the frontier coefficients. In this experiment, the highest error probability is to validate a false model, but as for the other experiments of model 3 as previously stated.

Model	Exp.	$P_{st} = 5\%$				$P_{st} = 2\%$			
		$m = 1.05$		$m = 1.5$		$m = 1.05$		$m = 1.5$	
		$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$
1	1	87/89	86/89	70/80	62/73	80/50	80/50	64/43	56/39
	2	75/90	75/90	75/89	68/86	65/46	64/46	63/43	59/41
	3	84/88	84/88	71/83	63/77	76/52	76/53	67/48	61/44
	4	63/83	63/83	60/82	54/77	60/59	60/59	63/55	57/51
	5	62/77	62/77	61/75	60/73	60/34	60/34	59/36	57/34
	6	71/61	70/60	48/56	29/45	56/28	53/28	41/29	23/22
	7	84/89	83/89	70/80	59/72	76/56	76/56	64/49	55/44
	8	90/91	90/91	74/76	66/68	79/51	79/51	65/38	56/34
	9	88/92	88/92	73/78	65/66	80/50	80/51	65/41	56/34
2	1	93/77	93/76	90/89	79/82	77/48	78/47	77/57	64/52
	2	13/7	13/8	11/14	7/11	5/3	5/3	4/7	2/5
	3	60/42	59/41	58/58	44/44	46/24	45/24	46/35	34/27
	4	97/81	97/81	92/93	87/89	83/49	83/49	84/63	78/60
	5	9/7	9/7	11/9	8/5	2/2	2/1	2/2	1/1
	6	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	7	41/43	39/43	54/56	40/42	25/21	24/19	38/23	24/16
	8	81/62	80/61	82/81	68/69	62/38	61/38	66/50	53/42
	9	85/64	84/63	80/79	69/68	66/39	66/38	65/49	53/41
3	1	71/29	70/28	45/22	32/17	65/19	64/19	41/14	30/12
	3	57/40	56/39	44/28	34/22	38/21	38/21	29/15	22/11
	4	64/19	64/18	65/18	54/14	58/12	59/11	57/10	46/8
	6	50/40	50/40	38/30	31/24	46/23	46/22	36/20	29/16
	7	23/22	23/22	32/46	27/30	25/11	24/10	34/25	27/18

Table 8: Rate of time series for which all the AR models exactly retrieved for $N_T = 512$ (first number) and $N_T = 256$ (second number)

Model	Exp.	$P_{st} = 5\%$				$P_{st} = 2\%$			
		$m = 1.05$		$m = 1.5$		$m = 1.05$		$m = 1.5$	
		$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$	$T_B = 0.7$	$T_B = 0.9$
1	1	57/57	56/56	45/44	42/40	83/81	83/80	66/63	59/56
	2	68/68	67/67	68/67	64/63	72/66	72/65	74/64	69/59
	3	40/40	39/39	39/39	36/36	68/66	66/64	64/63	59/57
	4	15/15	14/14	22/22	21/21	31/29	32/29	39/36	34/32
	5	41/41	41/41	48/47	49/48	55/52	55/53	66/62	66/61
	6	55/54	54/53	41/40	19/19	62/56	61/55	47/40	22/17
	7	47/46	47/46	48/44	44/39	72/67	72/67	73/66	63/57
	8	64/64	63/63	53/53	49/49	81/79	81/79	67/64	62/58
	9	62/62	63/63	55/55	50/50	81/81	82/82	67/65	58/56
2	1	99/95	99/95	99/94	98/84	99/84	99/85	99/83	98/66
	2	11/7	12/8	10/6	10/3	11/2	12/2	11/2	10/1
	3	87/57	86/56	92/59	89/50	87/35	86/35	92/41	89/32
	4	99/98	99/98	100/99	100/94	99/88	99/88	100/90	100/82
	5	28/9	29/9	37/12	34/7	28/3	29/2	37/6	34/4
	6	3/3	3/3	0/0	0/0	3/3	3/3	1/1	0/0
	7	39/22	38/20	66/44	57/36	39/13	38/15	66/30	57/26
	8	96/84	97/84	97/92	97/83	96/65	97/64	97/68	97/56
	9	97/86	97/86	98/87	97/78	97/69	97/69	98/74	97/58
3	1	60/60	60/60	45/43	37/35	65/61	65/60	52/45	44/38
	3	27/27	26/26	25/25	18/18	36/34	35/33	31/27	24/21
	4	49/48	48/48	38/37	30/29	63/60	64/61	52/49	37/35
	6	23/23	21/21	17/17	15/15	34/32	33/32	28/27	23/22
	7	39/39	37/37	21/21	14/14	39/36	37/34	21/16	14/11

Table 9: Same as table 5 for $d_{max} = 4$

5.3.3. Hyperparameters effects

For the sake of consistency, we only compare the AR model retrieval rate for the different experiments, concerning the time series length (table 8) and the d_{max} value (table 9), the final results of SETAR identification leading to close conclusions. Comparing tables 5, 6 and 7, we observe that there are few differences for the optimal hyperparameter sets for the AR model and frontier retrieval rate as well as the model validation metric (when possible). From the results of table 5 and 6, we observe that $P_{st} = 5\%$ leads to the best results for model 2 while $P_{st} = 2\%$ is the optimal value for model 1 and 3. As stated in the previous subsection, model 2 leads to closer residuals for the two AR models and a too strict stopping condition leads to a false AR model identification. In particular, for this model we observe that the exact AR model retrieval rate is divided by 2 with regards to AR model retrieval showing thus, that in this case, spurious AR models are frequently identified. The threshold $T_B = 0.7$ seems to be an optimal belonging measure to detect a possible centroid (i.e. an AR model). When $T_B = 0.9$ provides the best results, they are very close to those obtained with $T_B = 0.7$. The sharpness parameter m value is more questionable since a value of $m = 1.05$ gives the best results for almost all the cases, but for some experiments for model 2, we observe that the optimal results occur for $m = 1.5$. Thus, the two values have to be tested and the estimated model leading to highest well labelled array rate over the validation basis has to be selected. However, as seen in the next point the optimal hyperparameter set depends on the time series length.

The times series length has obviously some effects on the SETAR model identification even if we cannot derive any statistical properties (i.e. estimate consistency) since we are in the scope of non-stationary time series. Comparing tables 5 and 8 shows that the optimal hyperparameter set is always the same for large times series. In this case almost the optimal results are provided for parameters $P_{st} = 5\%$. Obviously for small times series when few samples remain to be labelled, in particular when $P_{st} = 2\%$, spurious and very inaccurate AR models can be identified from these few samples. In more details, for model 1 and 3, the results are identical, even better for small times series than for $N_T = 1024$ (few cases, exp 2 for instance). In this experiment the number of arrays, on which the centroids are estimated, increases exponentially with the time series length (see section 3.1). On the other side model 2 shows strong improvements of the AR retrieval results for increasing time series length.

Comparing tables 5 and 9 shows that the d_{max} parameter has few effects on the results when increasing from 3 to 4. The variations are not significant except in some cases. It has some negative effects for exp 3,4,5 of model 1 for which the rate decreases between 6% and 9% for the exact AR model retrieval. For model 2, exp 5 shows a strong dwindling of the AR retrieval (71% for $d_{max} = 3$ and 37% for $d_{max} = 4$) but the exact SETAR identification rate remains close for the two d_{max} values. We retrieve this case for exp 7 (96% to 66%) with close final results (43 % and 44 %). For model 3, the results are close for the two d_{max} values. In some cases, better results are obtained for $d_{max} = 4$ for instance for exp 2 of model 1 with an increase of 10% of the final results. Thus, as previously stated, the d_{max} parameter has few effects on the results but induces an exponentially increasing computational load. This is a drawback of our approach since it limits the AR order as well as the times series lags involved in the regime switching.

6. Conclusion

We have proposed a new paradigm/algorithm in order to identify SETAR models by first identifying the AR models using array clustering algorithm and secondly the switching conditions. This inversion allows us to relax the all the hypotheses on the SETAR model (number of regimes, switching conditions ...). Our approach does not allow deriving any asymptotical property, since for instance we consider stable and unstable AR models in the SETAR model. Thus, we validated our algorithm on several experiment sets. The results show the capability of our algorithm to identify fairly complex models (i.e. with more than two AR models) with possibly non Gaussian innovation and so on. The algorithm hyperparameters have few effects on the results, since as predicable, the dwindling of the identification results is owned to the SETAR model complexity. However, a limiting parameter is the solution space (maximal) dimension for computational load reason.

References

[1] K.-S. Chan, H. Tong, Chaos: A statistical Perspective, Springer, Berlin, 2001.

[2] J. Fan, Q. Yao, Nonlinear Times Series Nonparametric and parametric Methods, Springer, Berlin, 2003.

[3] P. Franses, D. van Dick, Non-linear time series in empirical finance, Cambridge university, Cambridge, 2000.

[4] P. Robinson, Time series with long memory, Oxford university press, Oxford, 2003.

[5] H. Tong, Threshold models in time series analysis-30 years on, Tech. rep., The university of Hong-Kong, department of statistics and actuarial science (April 2010).

[6] H. Tong, K. Lim, Threshold autoregression, limit cycle and cyclical data, Journal of the Royal statistical society, series B, (methodological) 42 (3) (1980) 245–292.

[7] J. Petrucelli, On the consistency of least square estimators for thresholds AR(1) model, Journal of time series analysis 7 (4) (1986) 269–279.

[8] J. Petrucelli, S. Woolford, A threshold AR (1) model, Journal of applied probability 21 (1984) 270–285.

[9] K. Chan, H. Tong, On the use of the deterministic Lyapounov for the ergodicity of stochastic difference equation, Advances in applied probability 17 (3) (1985) 666–678.

[10] R. Chen, R. Tsay, On the ergodicity of Tar(1) processes, The annals of applied probability 1 (1991) 613–634.

[11] D. Tjötheim, Non-linear times series and Markov chain, Applied Probability Trust 22 (1990) 587–611.

[12] T. Boucher, D. Cline, Stability of cyclic threshold and threshold-like autoregressive time series models, Statistica Sinica 17 (2007) 43–62.

[13] K. Chan, J. Petrucelli, H. tong, S. Woolford, A multiple-threshols AR(1) model, Applied Probability trust 22 (1985) 267–279.

[14] H. Chen, T. Chong, J. Bai, Theory and application of TAR model with two threshold variables, Economics Review 31 (2012) 142.

[15] T. Chong, Structural change in AR(1) models, Economic Theory 17 (2001) 87–136.

[16] R. Tsay, Testing and modelling threshold autoregressive processes, Journal of the American Statistical association 84 (1989) 231–240.

[17] K. Chang, H. Tong, On the use of the deterministic Lyaponov function for the ergodicity of stochastic difference equations, Advance in Applied probability 17 (3) (1985) 666–678.

[18] M. Caner, B. Hansen, Thresholds autoregression with a unit root, Econometrica 69 (2001) 1555–1596.

[19] R. Sollis, Testing the unit root hypothesis against TAR nonlinearity using STAR-based tests, Economics letters 112 (2011) 19–22.

[20] C. Chan, M. So, On a threshold heteroscedastic model, International journal of forecasting 22 (2006) 73–89.

[21] K. Chan, Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model, The annals of statistics 21 (1993) 520–533.

[22] N. Chan, Y. Kutoyants, On the parameter estimation of threshold autoregressive models, Statistical Inference for Stochastic Processes 15 (2012) 81–104.

[23] S. Wu, R. Chen, Threshold variable determination and threshold variable driven switching autoregressive models, Statistica Sinica 17 (2007) 241–264.

[24] B. Strikholm, T. Teräsvirta, Determining the number of regimes in a threshold autoregressive using smooth transition autoregression, Tech. rep., Department of Economic Statistics Stockholm School of economics (January 2005).

[25] J. Gonzalo, J. Pitarakis, Estimation and model selection based inference in single and multiple threshold models, Journal of econometrics 110 (2002) 319–352.

[26] G. Koop, S. Potter, Bayesian analysis of endogenous delay threshold models, Journal of business & Economic Statistics 21 (2003) 93–103.

[27] D. Politis, Model-Free Prediction and Regression: A Transformation-Based Approach to Inference, Springer, Berlin, 2015.

[28] T. Terävista, Specification, estimation and evaluation of smooth transition autoregressive models, Journal of the American society association 89 (1994) 208–218.

[29] T. Astakie, D. Watts, W. Watt, Nested threshold autoregressive (NeTAR) models, International journal of forecasting 13 (1997) 105–116.

[30] G. Box, G. Jenkins, Time series Analysis forecasting and control, Holden-Day, San Francisco, 1970.

[31] J. Le Caillec, R. Garello, Nonlinear system identification using autoregressive quadratic mo dels, Signal Processing 81 (2001) 357–379.

[32] M. Fischler, R. Bolles, Random samples consensus for model fitting with application to image analysis and automated cartography, Graphics and Image Processing 24 (1981) 381–394.

[33] H. Sheng, Y. Gao, B. Zhu, K. Wang, X. Liu, Feature extraction of SAR scattering centers using M-RANSAC and STFRFT-based algorithm., Eurasip Journal on advances in signal processing 46 (2016) DOI 10.1186/s13634–016–0345–z.

[34] W. Härdle, M. Müller, S. Sperlich, A. Werwats, Nonparametric and Semiparametric Models, Springer, Berlin, 2004.