electronics

MDPI

Article

On the Application of the Stability Methods to Time Series Data

Vicky Deng and Ciprian Doru Giurcăneanu *

Department of Statistics, University of Auckland, Auckland 1142, New Zealand; vickydqp@gmail.com
* Correspondence: c.giurcaneanu@auckland.ac.nz; Tel.: +64-9-923-2819

Abstract: The important problem of selecting the predictors in a high-dimensional case where the number of candidates is larger than the sample size is often solved by the researchers from the signal processing community using the orthogonal matching pursuit algorithm or other greedy algorithms. In this work, we show how the same problem can be solved by applying methods based on the concept of stability. Even if it is not a new concept, the stability is less known in the signal processing community. We illustrate the use of stability by presenting a relatively new algorithm from this family. As part of this presentation, we conduct a simulation study to investigate the effect of various parameters on the performance of the algorithm. Additionally, we compare the stability-based method with more than eighty variants of five different greedy algorithms in an experiment with air pollution data. The comparison demonstrates that the use of stability leads to promising results in the high-dimensional case.

Keywords: stability; time series; prediction; vector autoregressive model with exogenous variables; air pollution data

Citation: Deng, V.; Giurcăneanu, C.D. On the Application of the Stability Methods to Time Series Data. *Electronics* **2023**, *12*, 2988. https://doi.org/10.3390/electronics12132988

Academic Editors: Simeone Marino, Radu Ciprian Bilcu and Ionut Schiopu

Received: 17 May 2023
Revised: 25 June 2023
Accepted: 5 July 2023
Published: 7 July 2023

1. Introduction

1.1. Motivation

According to the presentation from [1], which considers the evolution of the statistical inference over the past decades, one of the prominent research topics after 1990 was high-dimensional statistical modeling, where the sample size (n) is much smaller than the number of covariates (p). This problem is difficult because the large value of p makes the total number of candidate variables impractically large [2].

For the sake of concreteness, let us assume that we possess the response vector $\boldsymbol{y} = [y_1 \cdots y_n]^\top$, where $(\cdot)^\top$ denotes transposition, and the matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_p]$ of p potential predictors. In many cases, the vector $\boldsymbol{y}$ and the columns of $\boldsymbol{X}$ are centered; the columns of $\boldsymbol{X}$ are normalized so that the Euclidean norm is the same for all of them. In signal processing, the matrix $\boldsymbol{X}$ is called a dictionary and the columns of the matrix $\boldsymbol{X}$ are dubbed atoms. We wish to represent $\boldsymbol{y}$ as a *sparse* linear combination of the columns of $\boldsymbol{X}$, i.e., $\boldsymbol{y} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$, where most of the entries of the vector $\widehat{\boldsymbol{\beta}}$ are equal to zero. It is evident that the residual sum of squares (RSS) is given by $\|\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|_2^2$, where $\|\cdot\|_2$ is the symbol for the Euclidean norm. However, if $p = 100$ and we want vector $\widehat{\boldsymbol{\beta}}$ to have exactly five non-zero entries, the naive approach that considers all possible subsets of five atoms for selecting the best one is totally impractical because the number of subsets that should be evaluated is $\binom{100}{5} = 75,287,520$.

1.2. Background and Related Works

1.2.1. Greedy Algorithms

A possible solution can be obtained in the following way. First, construct a sequence of linear models and then choose the best model from this sequence using either cross-validation or an information theoretic (IT) criterion. This strategy relies on greedy algorithms. One of the greedy algorithms often employed in signal processing is the Matching

Pursuit Algorithm (MPA) [3]. According to Scopus, the term "matching pursuit algorithm" appears in the title/abstract/keywords of 257 documents that have been published in journals/conference proceedings/book chapters that contain in their titles the term "signal processing". In MPA, $\widehat{\boldsymbol{\beta}}$ is initialized with the vector of length p whose entries are equal to zero, hence the initial value of RSS equals $\|\boldsymbol{y}\|_2^2$. At each iteration of MPA, the column of $\boldsymbol{X}$ is selected, which leads to the largest reduction of RSS. The entry of $\widehat{\boldsymbol{\beta}}$ that corresponds to the selected column of $\boldsymbol{X}$ is updated, whereas all other entries of $\widehat{\boldsymbol{\beta}}$ remain unchanged. The total number of iterations is large and gives the total number of competing models [4]. The selection of the best model by IT criteria is not straightforward because the criteria for Gaussian linear regression should be altered before applying them to the models generated by MPA. The interested reader can find in [5] a list of 22 IT criteria that are suitable for MPA as well as more technical details about the algorithm itself.

A variant of MPA commonly employed in signal processing is the Orthogonal Matching Pursuit (OMP) [6,7]. An indication of the fact that OMP is more popular than MPA in the signal processing community is provided by the number of Scopus documents found when, in the search mentioned above, we replace "matching pursuit algorithm" with "orthogonal matching pursuit". This time the result is 592, which is clearly greater than the result produced by the previous search. The technique used by OMP is the same as in MPA, in the sense that, at each iteration, the column of $\boldsymbol{X}$ that leads to the largest reduction of RSS is selected. The main difference is that all the entries of $\widehat{\boldsymbol{\beta}}$ that correspond to the columns of $\boldsymbol{X}$ selected at the current and past iterations are updated by least squares, hence the IT criteria for Gaussian linear regression can be used. For example, in [8], 12 different IT criteria have been employed for selecting the best model from the candidates yielded by OMP.

In the same reference, three other greedy algorithms are presented: Relaxed Matching Pursuit (RMP) [4,7], Frank-Wolfe Algorithm (FWA) [9] and Constrained Matching Pursuit (CMP) [4]. These algorithms are applied less frequently to signal processing problems. Performing a search for "Frank-Wolfe" on Scopus, where the other settings are the same as in the searches made before, returns only 35 documents. At the same time, it is remarkable that FWA and CMP are solvers for the Lasso problem [10], which can be formulated as a penalized estimation with an ℓ_1-penalty. For more details, see [4,11] and the references therein. Note that five algorithms for the Lasso problem are reviewed in [12]. Our search on Scopus (with the settings above) finds "lasso" in 514 documents, which demonstrates that Lasso is extensively used in signal processing research.

1.2.2. Stability Methods

It is known that one of the disadvantages of Lasso is the lack of control over the selection of false or irrelevant variables (see [13] for a discussion on this topic). An option for controlling the false discovery proportion is to consider the *stability* of the selection under subsampling. In connection with this approach, it was proposed in [14] that instead of applying Lasso to the whole data set of size n, Lasso is applied repeatedly to subsets of size $n/2$, and a variable chosen frequently when running the experiments is deemed to be relevant. For simplicity, we assume that n is even. More importantly, a variable is included in the model only if the empirical probability of being selected in the experiments is greater than a fixed threshold, which is chosen by the practitioner. The major difficulty is that the experiments should be executed for all $\binom{n}{n/2}$ subsets of size $n/2$. Another variant for stability-based variable selection was introduced in [15]. An important difference between the method from [15], which is called Complementary Pairs Stability Selection (CPSS), and the one from [14] is that CPSS does not consider in each experiment a subset of size $n/2$, but a pair of subsets of size $n/2$ whose intersection is the empty set. It is not needed to run experiments for all the pairs with these properties; it is enough to execute the subsampling B times, where B is a tuning parameter. The Lasso selection is applied to each subset in the pair and the statistics concerning how many times a particular variable is selected are

computed by taking into account the selection results obtained for each subset, in each experiment. The decision of including a variable in the model is based on the comparison of the computed statistic with a threshold. More interestingly, instead of a bound on false discovery proportion, CPSS asserts the bounds in the following terms: (i) "the expected number of variables chosen by CPSS that have low selection probability under the base selection procedure" and (ii) "the expected number of high selection probability variables that are excluded by CPSS". In the case that is of interest for us, the base selection procedure is Lasso, but it is evident that the stability methods from [14,15] can be applied to other selection procedures as well. As a continuation of the series of statistics from Scopus that we have presented above, we mention that the reference [15] is cited 171 times on Scopus, but there is no citation in journals/conference proceedings/book chapters that contain in their titles the term "signal processing". A possible explanation might be that CPSS as well as the stability method from [14] were designed for independent and identically distributed data. It seems that the only work in which CPSS was altered to be suitable for time series is [16]. The key point of the modification of CPSS proposed in [16] consists in sampling from data blocks that are 'almost' independent.

According to [17], the stability has been employed in statistical inference for a long time; it was applied not only in the cases where the data perturbation was produced by subsampling, but it was also used in conjunction with other perturbation schemes as jackknife or bootstrap. For instance, in the signal processing literature, the stability was proven to be instrumental in finding the number of groups when performing data clustering (see, for example [18]). A comprehensive analysis of the stability-based methods for clustering that have been proposed during the last decades can be found in [19]. More recently, the stability was used for the identification of differential equations from noisy spatio-temporal data [20]. The results of extensive simulation studies that evaluate the capabilities of the stability methods for high-dimensional biomedical data were reported in [21,22]. The aim of the study conducted in [21] was to compare the abilities of four base selection procedures to correctly identify the true predictors in artificial data when the stability criterion is applied. The results have shown that Lasso can lead to modest results when there is correlation among the significant variables. We should mention that the study did not use the stability methods from [15,16].

1.3. Organization of the Paper and the Main Contributions

In this work, we make an attempt to introduce the stability-based selection of predictors to the signal processing community:

- We consider the algorithm from [16]. The algorithm is conceptually simple and can be regarded as a modification of the method from [15], which does not seem to be currently used in signal processing.
- We relate this algorithm to the problems of interest in multivariate signal processing by applying the algorithm to the selection of predictors in the case of a vector autoregressive (VAR) model [23]. Scopus indicates that vector autoregressive can be found in the title/abstract/keywords of 359 documents published in signal processing venues. As we are interested in estimating an entry of a time series by employing the past observations and (if available) the current observations collected for other time series (see [5] and the references therein), we use a variant of the VAR model, which is called vector autoregressive with exogenous variables (VARX) [16,23]. In Section 2, we justify why the sparse VARX model is appropriate and show how the main algorithm, with Lasso as the base selection procedure, can be applied to find the relevant predictors.
- We conduct experiments with simulated data in order to evaluate the influence of various parameters on the performance of the main algorithm. As the ground truth is known, the performance is evaluated by measuring the feature selection capabilities in terms of the true positive rate and the false positive rate. We also discuss a modified variant of the main algorithm (see Section 3).

- We compare the performance of the stability-based method with the performance of methods that rely on greedy algorithms and IT criteria/cross-validation. The comparison involves more than eighty methods and it is carried out by using air pollution data that were measured in Auckland, New Zealand. As the 'true' predictors are not known for the real-life data, the comparison of various methods is made by considering the prediction accuracy. Additionally, we analyze the predictors that are selected most often and give an interpretation based on what is known from the environmental chemistry (see Section 4).

Section 5 concludes the paper.

2. Main Algorithm

2.1. Notation

We use bold letters for both vectors and matrices. In particular, $\boldsymbol{I}$ is the identity matrix of appropriate size. The symbol $\boldsymbol{v}(i)$ denotes the ith entry of an arbitrary vector $\boldsymbol{v}$. The notation for the number of non-zero entries of $||\boldsymbol{v}||$ is $||\boldsymbol{v}||_0$ and size$(\boldsymbol{v})$ denotes the total number of entries of $\boldsymbol{v}$. The symbol $||\boldsymbol{v}||_\eta$ denotes the ℓ_η-norm, where $\eta \in \{1,2\}$. In the case of an arbitrary matrix $\boldsymbol{M}$, $\boldsymbol{M}(i,j)$ is the entry located at the intersection of the ith row and the jth column. For a set of positive integers J, the symbol $\boldsymbol{M}(J,:)$ denotes the rows of $\boldsymbol{M}$ whose indexes are given by the elements of J and $\boldsymbol{M}(:,J)$ stands for the columns of $\boldsymbol{M}$ whose indexes are given by the elements of J. The operator for transposition is $(\cdot)^\top$. For any set A, $\mathbb{1}_A(\cdot)$ is the indicator function, which has the property that $\mathbb{1}_A(a) = 1$ if $a \in A$ and $\mathbb{1}_A(a) = 0$ otherwise. The cardinality of A is denoted $|A|$.

2.2. Problem Formulation and the Lasso Solution

Assume that the time series data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$, or equivalently $\{\boldsymbol{y}_t\}_{t=1}^T$, are available. For explaining how the stability methods can be applied to the time series data, we resort to the well-known VARX model [23]:

$$\boldsymbol{y}_t = \sum_{i=1}^{p_y} \boldsymbol{A}_i \boldsymbol{y}_{t-i} + \boldsymbol{C}\boldsymbol{v}_t + \boldsymbol{w}_t, \tag{1}$$

where $\boldsymbol{y}_t, \boldsymbol{w}_t \in \mathbb{R}^{K_y \times 1}$, $\boldsymbol{A}_i \in \mathbb{R}^{K_y \times K_y}$ for $i \in \{1, \ldots, p_y\}$, $\boldsymbol{C} \in \mathbb{R}^{K_y \times p_v}$ and $\boldsymbol{v}_t \in \mathbb{R}^{p_v \times 1}$. For simplicity of the presentation, we suppose that the data have been padded, hence the identity above holds true for all $t \in \{1, \ldots, T\}$. Note that the model in (1) can be easily extended such that to consider not only the observations $\boldsymbol{v}_t$, but also past measurements of the exogenous variables, $\boldsymbol{v}_{t-1}, \boldsymbol{v}_{t-2}, \ldots$ The vectors $\{\boldsymbol{w}_t\}_{t=1}^T$ are independent and identically distributed, and they are drawn from a K_y-variate Gaussian distribution with zero mean vector and covariance matrix $\sigma_v^2 \boldsymbol{I}$, where the value of σ_v^2 is unknown.

The identity in (1) can be written more compactly as follows:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{Z} + \boldsymbol{W}, \tag{2}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_T]$ and $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A}_1 \cdots \boldsymbol{A}_{p_y} \ \boldsymbol{C} \end{bmatrix}$.

With the convention that $\boldsymbol{z}_t = \begin{bmatrix} \boldsymbol{y}_{t-1}^\top \cdots \boldsymbol{y}_{t-p_y}^\top \ \boldsymbol{v}_t^\top \end{bmatrix}^\top$ for all $t \in \{1, \ldots, T\}$, we have $\boldsymbol{Z} = [\boldsymbol{z}_1 \cdots \boldsymbol{z}_T]$. Additionally, $\boldsymbol{W} = [\boldsymbol{w}_1 \cdots \boldsymbol{w}_T]$. Observe that $\boldsymbol{Y} \in \mathbb{R}^{K_y \times T}$, $\boldsymbol{H} \in \mathbb{R}^{K_y \times (K_y p_y + p_v)}$, $\boldsymbol{Z} \in \mathbb{R}^{(K_y p_y + p_v) \times T}$ and $\boldsymbol{W} \in \mathbb{R}^{K_y \times T}$.

In many practical situations, the matrix $\boldsymbol{H}$ is assumed to be sparse. Some of the reasons for the presence of zeros in $\boldsymbol{H}$ are:

- In general, the order of the autoregressions is not known and p_y is taken to be an upper bound for the unknown order. It is expected that the estimated order is smaller than this upper bound.
- An important result in the analysis and forecasting of multivariate time series claims that, for some $a, b \in \{1, \ldots, K_y\}$, y_b does not Granger-cause y_a if and only if the entry

indexed by (a, b) is zero for all matrix coefficients $\boldsymbol{A}_i$, where $i \in \{1, \ldots, p_y\}$ [23]. We note in passing that, there is an increasing interest in novel methods for the identification of vector autoregressive models with Granger and stability constraints (see [24] and the references therein). However, in the literature that is focused on this particular identification problem, the term 'stability' refers to the following condition that should be satisfied by the matrix coefficients: The magnitudes of all the eigenvalues of the matrix

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \ldots & \boldsymbol{A}_{p_y-1} & \boldsymbol{A}_{p_y} \\ \boldsymbol{I} & 0 & \ldots & 0 & 0 \\ 0 & \boldsymbol{I} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{I} & 0 \end{bmatrix} \tag{3}$$

are strictly less than one [23].

- There might be exogenous variables in $\boldsymbol{v}_t$ that do not influence some of the entries of $\boldsymbol{y}_t$.

In our presentation, we consider the problem of selecting the 'best' predictors for the first component of $\boldsymbol{y}_t$. This is mainly motivated by the real-life prediction problem that we will discuss in Section 4. It follows from (2) that

$$\boldsymbol{Y}(1,:) = \boldsymbol{H}(1,:)\boldsymbol{Z} + \boldsymbol{W}(1,:).$$

For ease of exposition, let us dub $\boldsymbol{y} = \boldsymbol{Y}(:,1)^\top$ and $\boldsymbol{h} = \boldsymbol{H}(:,1)^\top$. Note that the total number of predictors is $p = K_y \cdot p_y + p_v$. As it is desirable for the estimated vector of coefficients $\widehat{\boldsymbol{h}}$ to be sparse, we apply the Lasso method. More precisely,

$$\widehat{\boldsymbol{h}}_\lambda = \underset{\boldsymbol{h}}{\operatorname{argmin}}\, \mathcal{L}_\lambda(\boldsymbol{h}; \boldsymbol{y}, \boldsymbol{Z}), \tag{4}$$

where

$$\mathcal{L}_\lambda(\boldsymbol{h}; \boldsymbol{y}, \boldsymbol{Z}) = \frac{\|\boldsymbol{y} - \boldsymbol{Z}^\top \boldsymbol{h}\|_2^2}{2 \cdot \operatorname{size}(\boldsymbol{y})} + \lambda \|\boldsymbol{h}\|_1. \tag{5}$$

It is obvious that $\operatorname{size}(\boldsymbol{y}) = T$, but we prefer to write $\operatorname{size}(\boldsymbol{y})$ instead of T because later on we will employ the expression of $\mathcal{L}_\lambda(\cdot;\cdot)$ when the response vector and the matrix of predictors are not $\boldsymbol{y}$ and $\boldsymbol{Z}$. Hence, in (5), we emphasize that the quantity in the denominator of the first term is equal to the length of the response vector multiplied by two. To clarify why the input arguments for $\mathcal{L}_\lambda(\cdot;\cdot)$ are not necessarily $\boldsymbol{y}$ and $\boldsymbol{Z}$ we mention that, according to [16], the Lasso procedure is applied to the entire data set as well as to subsamples of the data set.

In [16], it is recommended to use the entire data set for selecting the value of λ in (5). For a fixed integer $q \in (0, p)$, the value of λ is varied until $\widehat{\boldsymbol{h}}_\lambda$ has q non-zero entries, which means that q predictors have been selected. The ratio $\theta = q/p$ is regarded as the average selection probability and all the predictors for which the probability of selection is smaller than θ are deemed to be irrelevant. Note that the value of λ selected in this way is then used in connection with the subsampling procedures that are described below.

2.3. Subsampling

The major difficulty in connection with subsampling stems from the fact the entries of the time series data are not independent. In order to address this issue, it was proposed in [16] to obtain from the time series data a sequence of blocks that are 'almost' independent. These blocks are obtained as follows: A partition of $2\mu_t$ subsets of the set $\{1, \ldots, T\}$ is defined such that each subset contains exactly a_T elements. We suppose that $T/(2a_T)$ is an integer. If this condition is not satisfied, then the expression $\lfloor T/(2a_T) \rfloor$ that involves the

floor operator should be used. The subsets of the partition are $O_1, \ldots, O_{\mu_T}$ and $E_1, \ldots, E_{\mu_T}$. For $j \in \{1, \ldots, \mu_T\}$,

$$\begin{aligned} O_j &= \{i : 2(j-1)a_T + 1 \leq i \leq (2j-1)a_T\}, \\ E_j &= \{i : (2j-1)a_T + 1 \leq i \leq 2ja_T\}. \end{aligned} \tag{6}$$

Additionally, we define:

$$O = \bigcup_{j=1}^{\mu_T} O_j. \tag{7}$$

It is helpful to employ a different notation for the odd blocks (which are the O-blocks) and the even blocks (which are the E-blocks) because the data corresponding to the odd blocks are used in estimation, whereas the data corresponding to the even blocks are discarded. The key point is that any two odd blocks are separated by at least a_T time points and, because of this feature, the odd blocks are deemed to be independent, especially when a_T is large. Hence, the subsampling is performed only from the odd blocks and not from the entire time series. In order to illustrate how the subsampling works, suppose that we sample without replacement the sequence of blocks $O'_1, \ldots, O'_{\mu_T/2}$ from the set of μ_T odd blocks. For ease of notation, we denote $\boldsymbol{y}(O')$ the vector formed by the entries of $\boldsymbol{y}$ whose indexes belong to the set

$$O' = \bigcup_{j=1}^{\mu_T/2} O'_j. \tag{8}$$

Similarly, $\boldsymbol{Z}(:, O')$ is the block of the matrix $\boldsymbol{Z}$ formed by the columns of $\boldsymbol{Z}$ whose indexes belong to the set O'. Therefore, instead of estimating the vector of linear coefficients by minimizing the cost function in (5), we minimize

$$\mathcal{L}_\lambda(\boldsymbol{h}; \boldsymbol{y}(O'), \boldsymbol{Z}(:, O')). \tag{9}$$

The main difference between (5) and (9) is that in the former we utilize all T data from $\boldsymbol{y}$, whereas in the latter we use only $(\mu_T/2)a_T$ data from $\boldsymbol{y}$. As it is evident that $(\mu_T/2)a_T$ is equal to the cardinality of O', we prefer to employ the symbol $|O'|$ when we refer to the amount of data.

2.4. Stability-Based Selection

The outcome of the Lasso for the optimization problem in (9) provides an estimator for the subset $S \subset \{1, \ldots, p\}$ that comprises the 'signal' variables. According to the nomenclature from [15,16], the subset $\{1, \ldots, p\} \setminus S$ contains the 'noise' variables. To fix the ideas, we give below the definition of the Lasso-based estimator for S:

$$\widehat{S}_{|O'|} = \left\{ i : \widehat{\boldsymbol{h}}(i) \neq 0, \widehat{\boldsymbol{h}} = \underset{\boldsymbol{h}}{\operatorname{argmin}} \mathcal{L}_\lambda\left(\boldsymbol{h}; \boldsymbol{y}(O'), \boldsymbol{Z}(:, O')\right) \right\}. \tag{10}$$

The presence of the symbol $|O'|$ in $\widehat{S}_{|O'|}$ indicates the amount of data involved in estimation.

Relying on the idea of the complementary pairs from [15], we consider the subset $O'' = O \setminus O'$ [see (7) and (8) for the definitions of O and O']. It is straightforward to obtain another estimator, $\widehat{S}_{|O''|}$, for the subset S of 'signal' variables by replacing O' with O'' in (9) and then applying Lasso for solving the optimization problem. Formally, the definition of $\widehat{S}_{|O''|}$ can be written down by using O'' instead of O' in (10). Bearing in mind that we aim to select the relevant predictors, we calculate the statistic

$$\mathbb{1}_{\widehat{S}_{|O'|}}(k) + \mathbb{1}_{\widehat{S}_{|O''|}}(k) \tag{11}$$

for each index $k \in \{1, \ldots, p\}$. Obviously, the statistic above can take only the values 0, 1 and 2. Following the recipe from [15,16], we execute the subsampling not only once, but B times. We will discuss later on how the value of B can be chosen. Most importantly, at each subsampling, a different pair of sets (O', O'') is generated and the statistic in (11) is computed for each predictor. Furthermore, for each $k \in \{1, \ldots, p\}$, we calculate the sum of the statistics (11) obtained for the kth predictor in all B subsamplings and divide the result by $2B$ in order to obtain the estimator $\widehat{\Pi}_B^{av}(k)$. The possible values for $\widehat{\Pi}_B^{av}(k)$ are $\frac{0}{2B}, \frac{1}{2B}, \frac{2}{2B}, \ldots, \frac{2B}{2B}$. It allows us to find the block average selection estimator

$$\widehat{S}_\phi^{av} = \left\{k : \widehat{\Pi}_B^{av}(k) \geq \phi\right\}, \tag{12}$$

where ϕ is a threshold that, in general, can take values in the interval $(0.5, 0.9]$. The value of ϕ is chosen by resorting to theoretical results on stability selection that provide an upper bound for the expected number of falsely selected predictors. This is the major advantage of the use of stability in comparison with applying only the Lasso base procedure. The main steps of the selection method presented in this section are shown in Algorithm 1. In the next sections, we investigate the performance of the algorithm in experiments with simulated data and air pollution data.

Algorithm 1 Stability selection with the base procedure Lasso (see [16] (Algorithm 1))

Input: $\boldsymbol{y} \in \mathbb{R}^{T\times 1}$ [T measurements], $\boldsymbol{Z} \in \mathbb{R}^{p\times T}$ [p predictors],
a_T [the cardinality for each odd block], q/p [average selection probability],
ϕ [threshold], B [number of pairs (O', O'')],
$\Lambda = \{\lambda_1, \ldots, \lambda_{100}\}$ [a sequence of penalty parameters]
Initialize: $\widehat{\Pi}_B^{av}(k) = 0$, for all $k \in \{1, \ldots, p\}$
Select the penalty parameter: For $\lambda \in \Lambda$, solve

$$\widehat{\boldsymbol{h}}_\lambda = \underset{\boldsymbol{h}}{\operatorname{argmin}}\, \mathcal{L}_\lambda(\boldsymbol{h}; \boldsymbol{y}, \boldsymbol{Z}) \text{ [see (5)]}$$

and then set

$$\lambda_q = \underset{\lambda\in\Lambda}{\operatorname{argmin}}\left\{\|\widehat{\boldsymbol{h}}_\lambda\|_0 = q\right\}$$

for $n = 1$ to B **do**
Sample: From the odd blocks $O_1, \ldots, O_{\mu_T}$, sample without replacement the sequence of blocks $O'_1, \ldots, O'_{\mu_T/2}$, construct O' and set $O'' = O \setminus O'$ [see (7) and (8)]
Estimate:

$$\widehat{S}_{|O'|} = \left\{ i : \widehat{\boldsymbol{h}}(i) \neq 0, \widehat{\boldsymbol{h}} = \underset{\boldsymbol{h}}{\operatorname{argmin}}\, \mathcal{L}_{\lambda_q}\left(\boldsymbol{h}; \boldsymbol{y}(O'), \boldsymbol{Z}(:, O')\right) \right\} \text{ [see (10)]}$$

$$\widehat{S}_{|O''|} = \left\{ i : \widehat{\boldsymbol{h}}(i) \neq 0, \widehat{\boldsymbol{h}} = \underset{\boldsymbol{h}}{\operatorname{argmin}}\, \mathcal{L}_{\lambda_q}\left(\boldsymbol{h}; \boldsymbol{y}(O''), \boldsymbol{Z}(:, O'')\right) \right\}$$

$$\widehat{\Pi}_B^{av}(k) = \widehat{\Pi}_B^{av}(k) + \frac{1}{2B} \cdot \mathbb{1}_{\widehat{S}_{|O'|}}(k) + \frac{1}{2B} \cdot \mathbb{1}_{\widehat{S}_{|O''|}}(k), \text{ for all } k \in \{1, ..., p\}$$

end for
Output: $\widehat{S}_\phi^{av} = \left\{k : \widehat{\Pi}_B^{av}(k) \geq \phi\right\}$ [see (12)]

3. Experiments with Simulated Data

3.1. Artificial Data

We generate data sets according to the VARX model in (1). In our simulations, all VARX models are guaranteed to be stable (see again the stability condition in (3)). The most important attributes are:

- Parameters: (i) number of time series: $K_y = 4$; (ii) autoregressive order: $p_y = 3$; (iii) standard deviation for the non-zero entries of the matrices $\{\boldsymbol{A}_i\}$ and for the entries of the vectors $\{\boldsymbol{v}_t\}$: $\sigma = 0.25$.
- Matrices $\{\boldsymbol{A}_i\}$: (i) We generate the matrices $\{\widetilde{\boldsymbol{A}}_i\}_{i=1}^{p_y}$ whose entries are independent outcomes from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$; (ii) let $\boldsymbol{M} \in \mathbb{R}^{K_y \times K_y}$ be a matrix that has all entries equal to one. Some entries on the off-diagonal locations of $\boldsymbol{M}$ are randomly selected and forced to be zero (the number of zero entries is restricted to be less than $0.6 \cdot K_y^2$); (iii) for $1 \leq i \leq p_y$, the element-wise product of $\boldsymbol{M}$ and $\widetilde{\boldsymbol{A}}_i$ gives the matrix $\boldsymbol{A}_i$.
- Matrix $\boldsymbol{C}$: (i) $c_{11} = c_{12} = c_{13} = 1$; (ii) all other entries are equal to zero.
- Vectors $\{\boldsymbol{v}_t\}$: (i) model for the first entry: zero-mean order-1 autoregressive process with autocorrelation function $\rho(\tau) = (-0.9)^{|\tau|}$, for $\tau \in \mathbb{Z}$; model for the second entry: zero-mean order-1 autoregressive process with autocorrelation function $\rho(\tau) = (-0.5)^{|\tau|}$, for $\tau \in \mathbb{Z}$; model for all other entries: independent outcomes from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$.
- Vectors $\{\boldsymbol{w}_t\}$: model: multivariate Gaussian distribution with zero-mean vector and covariance matrix $\sigma_v^2 \boldsymbol{I}$, where $\sigma_v^2 = (\psi\sigma)^2$ and $\psi \in \{0.1, 1, 10\}$.
- Signal-to-noise ratio (SNR): (i) formula: $10 \log_{10} \dfrac{\sigma^2}{\sigma_v^2} = 10 \log_{10} \dfrac{1}{\psi^2}$; (ii) values: -20 dB, 0 dB and 20 dB.

With these settings, we simulate 100 data sets for each value of SNR mentioned above. For each data set, we have: (i) number of measurements: $T = 10,000$; (ii) maximum order of autoregressions (used in estimation): $p_y^{\max} = 4$ (greater than the 'true' order p_y); (iii) number of endogenous predictors: $p_y^{max} \cdot K_y = 4 \cdot 4 = 16$; (iv) number of exogenous predictors: $p_v = 84$; total number of predictors: $p = 16 + 84 = 100$.

The methodology utilized for producing the artificial data has a certain degree of similarity with the one used for simulating the data in an experiment reported in [8]. There are two differences between the approach in this work and the one in [8]. The first difference stems from the fact that, in [8], all the odd blocks have the same cardinality and all the even blocks have the same cardinality, but $|O_1|$ can be different from $|E_1|$. The second difference is that, in [8], after generating the blocks $O_1, \ldots, O_{\mu_T}$, only the data in $\boldsymbol{y}(O_1)$ are used in order to select the best predictors by employing a greedy algorithm and an IT criterion. The same pair (greedy algorithm, IT criterion) is then used for selecting the predictors from the data in $\boldsymbol{y}(O_2)$ and the procedure continues until the data from each odd block is utilized in the selection of the predictors. It is decided that a particular predictor can be included in the final model only if it was selected in at least 80% of the data blocks $\boldsymbol{y}(O_1), \ldots, \boldsymbol{y}(O_{\mu_T})$.

In the section below, we focus on the parameters for Algorithm 1. For the sake of simplicity, we sometimes use the acronym BPA for Algorithm 1. This acronym is inspired by [16] (Definition 1) and means Block Pair Average.

3.2. Settings for the Parameters of the Algorithm

3.2.1. Parameter a_T [The Cardinality for Each Odd Block]

The value of a_T is usually set to an integer multiple of the time series seasonality, and set to $\sqrt{T}$ or $\log T$ in the absence of seasonality [16]. As $\mu_T = T/(2a_T)$, it is evident that a_T determines the number of odd blocks μ_T. When a_T is larger, we have longer blocks and, therefore, fewer odd blocks. However, the number of data involved in estimation does not depend on a_T because the sample size is given by $|O'| = |O''| = (\mu_T/2)a_T = T/4$. The possible influence of a_T on the estimation results comes from the fact that the distance

between any two consecutive odd blocks O_j and O_{j+1}, $j \in \{1, \ldots, \mu_T - 1\}$ is larger when a_T increases and this potentially makes the blocks more independent. In our empirical evaluations of the algorithm, we allow a_T to vary over an extensive range of values: $a_T \in \{5, 10, 50, 100, 250, 500\}$.

3.2.2. Parameter q/p [Average Selection Probability]

The magnitude of q determines the penalty parameter λ_q; the lower the q parameter, the harsher the Lasso penalty. The value $q/p = 0.4$ was set as default in [16], with $q/p = 0.2$ and $q/p = 0.6$ tested in their real-life data experiment. When $q/p = 0.2$, it might be too conservative, while $q/p = 0.6$ is less conservative and easier to let predictors in for the second screening controlled by ϕ. Hence, we want to explore the different values of $q/p \in \{0.2, 0.4, 0.6\}$.

3.2.3. Parameter ϕ [Threshold]

We will pay close attention to the relationship between q and ϕ. For a lower ϕ-threshold, it is easier for variables to be selected, while it will become harder when the ϕ-threshold is higher. For example, only the most prominent and useful variables will be selected for a threshold of 0.9. Therefore, it is expected that the final number of variables selected will be smaller when we raise the ϕ-threshold. The value for ϕ is set to 0.8 in [16], while the parameter values tested in our experiments are $\phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

3.2.4. Parameter B [Number of Pairs (O', O'')]

The computational complexity of Algorithm 1 is mainly given by (i) the cost for running Lasso over all regularization parameters in the set Λ when the dictionary size is $T \times p$ and (ii) the cost for executing Lasso $2B$ times for the fixed penalty parameter λ_q when the dictionary size is $(T/4) \times p$. More information about the computational complexity for (i) and (ii) can be found in [2,12]. The number of iterations B is set to 50 in [16]. Since more iterations make the stability-based selection method more computationally intensive, we want to find out whether lowering B can retain the same performance of the algorithm. Hence, B is tested for the values 5, 25, 50. When experimenting on B, we keep in mind that at each iteration we should have a different pair (O', O''). Given that T is fixed, this condition is mainly related to the value of a_T. Note that even for the maximum value of a_T employed in our experiments ($a_T = 500$), the total number of ways in which one can sample without replacement $\mu_T/2$ O'-blocks from μ_T odd blocks is large enough. Elementary calculations show that $\mu_T = 10$ when $a_T = 500$ and the number of ways in which the O'-blocks can be sampled from the O-blocks is $\binom{10}{5} = 252$, which is greater than 50.

3.3. Empirical Evaluation of the Effect of Various Settings

3.3.1. SNR Level

In the first experiment, we use the 100 data sets that we have generated for each SNR-value and test how BPA performs under different noise levels. While the noise level changes, the parameters of the algorithm are fixed to the following default values: $a_T = 100$, $q/p = 0.4$, $\phi = 0.8$ and $B = 50$. The true positive rate (TPR) and the false positive rate (FPR) are calculated and they are shown in Figure 1. In the figure, we see that TPR does not change much, and it stays at a pretty high level when SNR varies. However, there are more outliers and slightly lower TPR for higher SNR. The FPR drops as SNR increases, which is expected because the stability method is less likely to detect false predictors when the data are less noisy. The results suggest that BPA is quite robust to noise.

To explore further the cause of slightly lower TPR for higher SNR, we investigate the selection of λ_q and the number of predictors selected (p^*) for each SNR level. In Figure 2, we notice that the λ_q-values are generally very similar, for all levels of SNR. It is interesting that for SNR = 20 dB there are more outliers and a broader range of λ_q-values than for SNR = -20 dB.

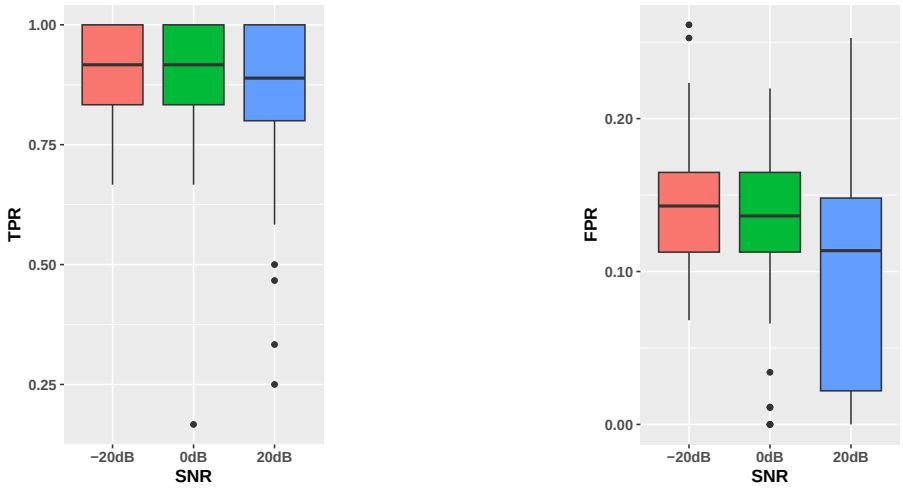

Figure 1. Boxplots for TPR (**left panel**) and FPR (**right panel**), for three different SNR levels.

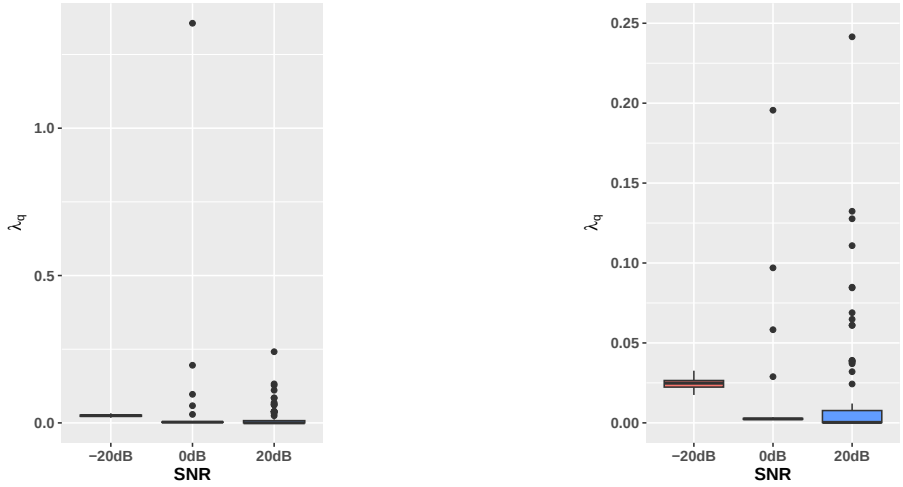

Figure 2. Boxplots for the values of λ_q, for three different SNR levels. The graph in the right panel is obtained from the one in the left panel after removing the outlier that occurs at SNR = 0 dB.

Next, we investigate the effect of the threshold ϕ on the final number of predictors selected. The results, which are shown in Figure 3, indicate that the number of the predictors selected decreases significantly after applying the rule based on the threshold $\phi = 0.8$ at the last step of the algorithm. This confirms that the Lasso regularization controlled by q (via λ_q) only serves as an instrument for the preliminary selection of the predictors [16] and ϕ has a critical role in identifying the high selection probability predictors that are included in the final model. We also notice in Figure 3 that there are fewer final predictors selected for SNR = 20 dB than for the smaller SNR values, which results in both TPR and FPR being lower for SNR = 20 dB in Figure 1.

We continue the analysis by considering various values for the parameters of the algorithm (see again Section 3.2). For conciseness, we do not report the results obtained for the parameter a_T because our experimental findings show that a_T does not have an important influence on TPR and FPR. The other parameters have a stronger impact on the outcome of the experiments.

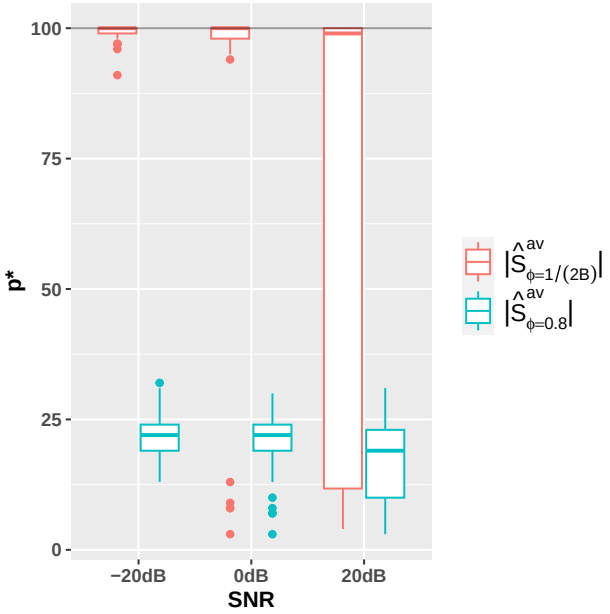

Figure 3. Boxplots for the number of the selected predictors (p^*) before and after applying the rule that involves the threshold $\phi = 0.8$. The results are presented for three different SNR levels. The other parameters are fixed at $a_T = 100$, $q/p = 0.4$, and $B = 50$. The solid black line represents the total number of predictors: $p = 100$. We employ the definition from (12) for the particular case when $\phi = 1/(2B) = 0.01$ in order to denote $|\widehat{S}^{av}_{\phi=1/(2B)}|$ the number of predictors that are selected at least once in the 50 iterations of the algorithm. The red boxes are for $|\widehat{S}^{av}_{\phi=1/(2B)}|$. The effect of the first layer of selection, which is performed by Lasso with the penalty parameter λ_q, consists in the fall of the number of the chosen predictors from the level of the black line to the level of the red boxes. The blue boxes are for $|\widehat{S}^{av}_{\phi=0.8}|$, which represents the number of predictors that are chosen in the second layer of predictor selection at least 80 times [$80 = \phi(2B)$] in the course of the 50 iterations of the algorithm. Hence, the effect of the second layer of selection is that the number of the predictors chosen falls from the level of the red boxes to the level of the blue boxes.

3.3.2. Parameter q/p [Average Selection Probability]

In Figure 4, we observe that as q/p increases, the Lasso penalty becomes weaker and more variables can pass through the selection criteria. Note that the second selection criterion is based on the threshold ϕ, which is fixed at the value 0.8. Our results suggest that keeping ϕ constant and loosening the first selection criterion by increasing q/p leads to higher TPR and higher FPR. Interestingly, the graphs in the figure show a larger increase in FPR than in TPR. The much higher FPR for $q/p = 0.6$ suggests that the loose Lasso regularization has a considerable impact on the correct final selections. The influence of $\phi = 0.8$ (the second condition being quite harsh) can only be effective at filtering out the less good variables and keeping the FPR relatively low when the variable cannot pass the first criterion easily.

3.3.3. Parameter ϕ [Threshold]

In Figure 5, we see that as ϕ increases, the FPR drops together with a slight drop in TPR. When $\phi \in \{0.5, 0.6, 0.7\}$, the values of FPR are far too large. Therefore, ϕ should not be less than 0.8. Setting $\phi = 0.9$ is acceptable, but it may be too harsh and lowers TPR. We conclude that $\phi = 0.8$ is a good choice.

3.3.4. Parameter B [Number of Pairs (O', O'')]

In Figure 6, we observe that the results for B suggest no significant difference in TPR/FPR between 25 and 50 iterations. This means that 25 iterations are sufficient to

arrive at a good selection of the final predictors. Still, if we have the computing capacity at 50 iterations, it will help improve the FPR slightly. However, when we reduce the number of iterations to 5, the FPR is clearly higher than in the cases of 25 and 50 iterations. Thus, we should probably not use too few iterations.

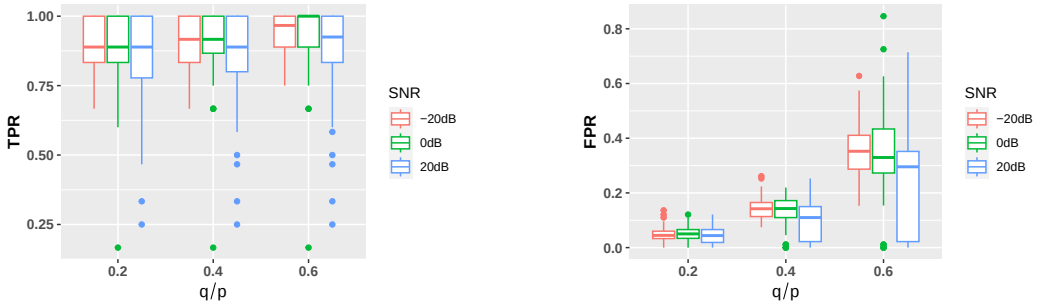

Figure 4. Boxplots for TPR and FPR obtained when q/p takes three different values and the other parameters are fixed at $a_T = 100$, $\phi = 0.8$, and $B = 50$. Remark that we use a different color for each SNR level.

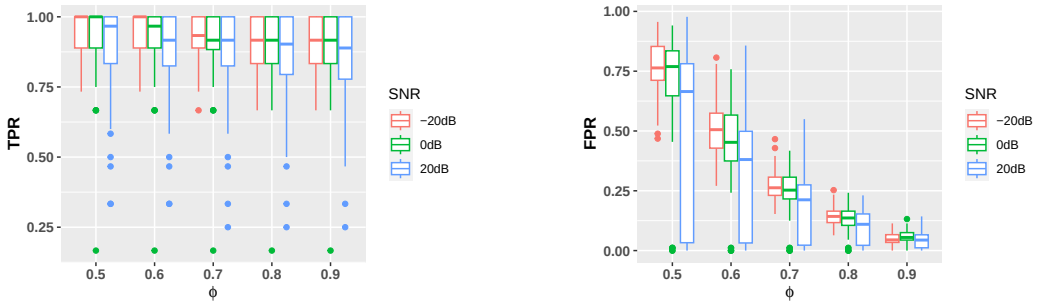

Figure 5. Boxplots for TPR and FPR obtained when ϕ takes five different values and the other parameters are fixed at $a_T = 100$, $q/p = 0.4$, and $B = 50$.

Figure 6. Boxplots for TPR and FPR obtained when B takes three different values and the other parameters are fixed at $a_T = 100$, $q/p = 0.4$, and $\phi = 0.8$.

3.3.5. A Variant of the BPA

We investigate the performance of a variant of BPA in which the Lasso selection is performed on each of the data blocks $\boldsymbol{y}(O_1), \ldots, \boldsymbol{y}(O_{\mu_T})$. Let N_k denote the number of times the kth predictor is selected, where $k \in \{1, \ldots, p\}$. It is obvious that $0 \leq N_k \leq \mu_T$. We decide that the kth predictor should be included in the final model if $\dfrac{N_k}{\mu_T} \geq \phi$, where $\phi = 0.8$. Remark that in comparison with Algorithm 1, there is no random selection of the O'-blocks and the small data blocks are not aggregated. In order to distinguish this variant of the algorithm from the original one, we call it BPA-m. Another modification in comparison with Algorithm 1 is that in the selection of λ_q only the data in $\boldsymbol{y}(O_1)$ are utilized. The reason for this change is that the use of all $T = 10,000$ data for the selection of

λ_q would give a small λ_q that cannot perform an effective regularization on data blocks of length $a_T = 100$. The original BPA does not have this issue because the small data blocks are aggregated, hence the size of $\boldsymbol{y}(O')$ as well as the size of $\boldsymbol{y}(O'')$ is equal to $T/4 = 2500$.

The comparison of the results produced by BPA and BPA-m is presented in Figure 7, which indicates that BPA performs much better than BPA-m in terms of TPR, but performs slightly worse in terms of FPR. Therefore, BPA-m is too conservative compared to BPA. We also notice that the TPR improvement from increasing SNR is very clear for BPA-m.

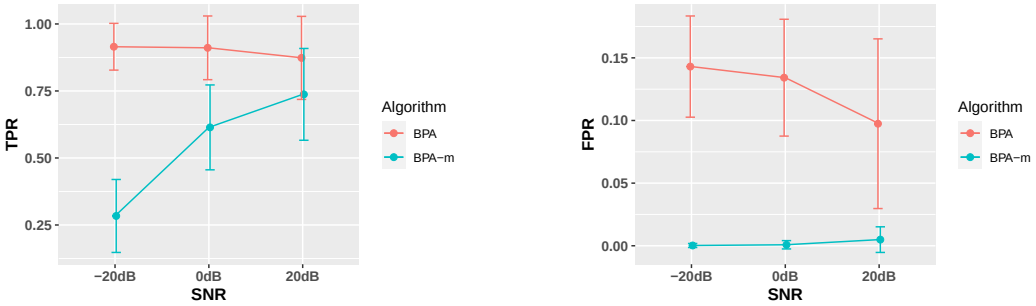

Figure 7. Mean ± standard deviation for TPR and FPR obtained when the algorithms BPA and BPA-m are employed. Three different SNR levels are considered. The other parameters are fixed at $a_T = 100$, $q/p = 0.4$, $\phi = 0.8$, and $B = 50$.

4. Experiments with Real-Life Data

4.1. Air Pollution Data

In this section, we apply the BPA stability method to the Auckland air pollution data set that was used in [5,8]. The problem that we address is the same as the one considered in [5,8] and concerns the selection of the predictors. The main difference between our approach and the methods used in the previous works comes from the fact that we do not use greedy algorithms and IT criteria/cross-validation for selecting the predictors.

The Auckland air pollution data set comprises daily measurements of the concentrations of particulate matter (PM), specifically $PM_{2.5}$ and PM_{10} (in $\mu g/m^3$) at four locations in Auckland, New Zealand. Note that $PM_{2.5}$ includes particles less than 2.5 μm in diameter; PM_{10} includes particles less than 10 μm in diameter. Therefore, $PM_{2.5}$ is a subset of PM_{10}. The sites where the data were measured are Patumahoe (PA), Penrose (PE), Takapuna (TA), and Whangaparaoa (WH). In parentheses are written the acronyms that we use in this work for the sites. The measurements were collected from 30 April 2008 to 30 June 2014.

For a measurement site, let $\theta_{\text{site}}(1), \theta_{\text{site}}(2), \ldots$ be the time series of log-transformed daily concentrations of $PM_{2.5}$. For example, $\theta_{\text{PA}}(1)$ is the log-transformed $PM_{2.5}$ measurement for Patumahoe, on the first day. Similarly, $\xi_{\text{site}}(1), \xi_{\text{site}}(2), \ldots$ are the log-transformed values for the concentrations of PM_{10} measured at the specific site, during day $1, 2, \ldots$

We want to estimate the concentration of $PM_{2.5}$ at a specific site when this concentration cannot be measured because of a malfunction of the sensors. To this end, we find a linear model that describes the relationship between the log-transformed concentration of $PM_{2.5}$ on the current day (at a specific site) and the following variables: (i) past and present log-transformed concentrations of PM_{10} for that specific site and (ii) past and present log-transformed concentrations of $PM_{2.5}$ for the other three sites.

Let $n = 365$ because there are 365 days during a normal year. For a certain site, we take the response vector to be

$$\boldsymbol{y}_{\text{site}}(t) = [\theta_{\text{site}}(t)\ \theta_{\text{site}}(t-1)\ \ldots\ \theta_{\text{site}}(t-n+1)]^\top, \tag{13}$$

where $t \geq n$ denotes an arbitrary day. For the same site, we use the notation $\boldsymbol{X}_{\text{site}}(t)$ for the matrix of predictors. The columns of the matrix are arranged into four blocks. The representations for the response vector and the matrix of predictors for each site are shown

in Figure 8a. Note that the response vector and the columns of the predictors matrix are centered and standardized.

We consider two scenarios for constructing the four blocks of the matrix of predictors.

- Scenario A—Full set of predictors (FullSet): The first block contains the measurements from last year of the log-transformed concentrations of PM_{10} for the site for which we wish to estimate the concentration of $PM_{2.5}$. The next three blocks contain the log-transformed concentrations of $PM_{2.5}$ for the other three sites that have been measured during the last year. The predictors are presented in Figure 8b. The total number of predictors is $p = 4(n+1) = 1464$, thus $p \gg n$ (high-dimensional case).
- Scenario B—Constrained set of predictors (ConSet): In this scenario, we reduce the total number of predictors by using empirical knowledge (see [5]). The predictors are presented in Figure 8c. Simple calculations show that the total number of predictors is $p = 4 \cdot 17 = 68$, thus $p < n$.

4.2. Performance Evaluation

The performance evaluation is performed by using the same methodology as in [5,8]. The procedure described below is applied for $\mathrm{N}_{TR} = 100$ runs. In each run, we use a data segment of length $3n$ (three consecutive years of measurements). The first two years are used for training (i.e., for selecting the predictors by applying BPA) and the third year is used for testing. In the first run of this experiment, we take t_0 to be 30 April 2008 and let t_1 be the last day of the second year, so $t_1 = t_0 + 2n - 1$. It follows that the training response vector is $\boldsymbol{y}_{\text{site}}(t_1)$ and the training predictors matrix is $\boldsymbol{X}^{\text{SC}}_{\text{site}}(t_1)$, where $(\cdot)^{\text{SC}}$ is used to distinguish between scenario A and scenario B. After the BPA is employed for selecting the predictors from the training predictors matrix, the corresponding linear coefficients are computed by minimizing the sum of the squared residuals (least squares). The vector of coefficients is further used together with the testing predictors matrix in the third year $\boldsymbol{X}^{\text{SC}}_{\text{site}}(t_1 + n)$ in order to produce the estimate $\widehat{\boldsymbol{y}}_{\text{site}}(t_1 + n)$. The same procedure is used in the rth run, where $r \in \{2, \ldots, \mathrm{N}_{TR}\}$, by replacing t_1 with $t_r = t_1 + (r-1)8$.

The normalized mean square error (NMSE) is computed by applying the following formula for each site and for each scenario:

$$\mathrm{NMSE}^{\text{SC}}_{\text{site}} = \frac{\sum_{r=1}^{\mathrm{N}_{TR}} \left\| \exp(\boldsymbol{y}_{\text{site}}(t_r + n)) - \exp\left(\widehat{\boldsymbol{y}}^{SC}_{\text{site}}(t_r + n)\right) \right\|^2}{\sum_{r=1}^{N_{TR}} \left\| \exp(\boldsymbol{y}_{\text{site}}(t_r + n)) \right\|^2}, \tag{14}$$

where applying $\exp(\cdot)$ to the vector means that we exponentiate each entry of the vector in the argument. Exponentiation is used to back-transform the log-transformed data to the original scale.

4.3. Experimental Results

First we write down the values of the BPA parameters that are used in this experiment. For instance, it is natural to take $T = 365$ because $n = 365$. Furthermore, we take $a_T = 3 \cdot 7 = 21$, which is a multiple integer of the seasonality. The factor 3 is selected for the convenience of the implementation as we already know from the experiments with simulated data that the value of a_T does not have an important effect on the performance of the algorithm. It follows that $\mu_T = \lfloor T/(2a_T) \rfloor = \lfloor 365/(2 \cdot 21) \rfloor = 8$. Therefore, there are eight odd blocks in total, and when running the algorithm we sample without replacement the blocks $O'_1, \ldots, O'_4$. According to (8), $|O'| = 4 \cdot 21 = 84$; it is also clear that $|O''| = |O'|$. The other BPA parameters are: $q = \lfloor 0.4 \cdot p \rfloor$, $\phi = 0.8$ and $B = 50$. Their selection is based on the results obtained with simulated data.

Response vector	Matrix of predictors
$\mathbf{y}_{\mathrm{PA}}(t) = [\boldsymbol{\Theta}_{\mathrm{PA}}(t)]_{:1}$	$\mathbf{X}_{\mathrm{PA}}(t) = [\boldsymbol{\Xi}_{\mathrm{PA}}(t)\ \boldsymbol{\Theta}_{\mathrm{PE}}(t)\ \boldsymbol{\Theta}_{\mathrm{TA}}(t)\ \boldsymbol{\Theta}_{\mathrm{WH}}(t)]$
$\mathbf{y}_{\mathrm{PE}}(t) = [\boldsymbol{\Theta}_{\mathrm{PE}}(t)]_{:1}$	$\mathbf{X}_{\mathrm{PE}}(t) = [\boldsymbol{\Xi}_{\mathrm{PE}}(t)\ \boldsymbol{\Theta}_{\mathrm{PA}}(t)\ \boldsymbol{\Theta}_{\mathrm{TA}}(t)\ \boldsymbol{\Theta}_{\mathrm{WH}}(t)]$
$\mathbf{y}_{\mathrm{TA}}(t) = [\boldsymbol{\Theta}_{\mathrm{TA}}(t)]_{:1}$	$\mathbf{X}_{\mathrm{TA}}(t) = [\boldsymbol{\Xi}_{\mathrm{TA}}(t)\ \boldsymbol{\Theta}_{\mathrm{PA}}(t)\ \boldsymbol{\Theta}_{\mathrm{PE}}(t)\ \boldsymbol{\Theta}_{\mathrm{WH}}(t)]$
$\mathbf{y}_{\mathrm{WH}}(t) = [\boldsymbol{\Theta}_{\mathrm{WH}}(t)]_{:1}$	$\mathbf{X}_{\mathrm{WH}}(t) = [\boldsymbol{\Xi}_{\mathrm{WH}}(t)\ \boldsymbol{\Theta}_{\mathrm{PA}}(t)\ \boldsymbol{\Theta}_{\mathrm{PE}}(t)\ \boldsymbol{\Theta}_{\mathrm{TA}}(t)]$

(**a**)

Measurements for $\mathrm{PM}_{2.5}$ when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Theta}_{\mathrm{site}}(t) = \begin{bmatrix} \theta_{\mathrm{site}}(t) & \theta_{\mathrm{site}}(t-1) & \cdots & \theta_{\mathrm{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{\mathrm{site}}(t-n+1) & \theta_{\mathrm{site}}(t-n) & \ldots & \theta_{\mathrm{site}}(t-2n+1) \end{bmatrix}$$

Measurements for PM_{10} when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Xi}_{\mathrm{site}}(t) = \begin{bmatrix} \xi_{\mathrm{site}}(t) & \xi_{\mathrm{site}}(t-1) & \cdots & \xi_{\mathrm{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{\mathrm{site}}(t-n+1) & \xi_{\mathrm{site}}(t-n) & \ldots & \xi_{\mathrm{site}}(t-2n+1) \end{bmatrix}$$

(**b**)

Measurements for $\mathrm{PM}_{2.5}$ when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Theta}_{\mathrm{site}}(t) = \begin{bmatrix} \theta_{\mathrm{site}}(t) & \theta_{\mathrm{site}}(t-1) & \cdots & \theta_{\mathrm{site}}(t-10) & \theta_{\mathrm{site}}(t-182) & \theta_{\mathrm{site}}(t-183) & \theta_{\mathrm{site}}(t-184) & \theta_{\mathrm{site}}(t-n+2) & \theta_{\mathrm{site}}(t-n+1) & \theta_{\mathrm{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \theta_{\mathrm{site}}(t-n+1) & \theta_{\mathrm{site}}(t-n) & \ldots & \theta_{\mathrm{site}}(t-n-9) & \theta_{\mathrm{site}}(t-n-181) & \theta_{\mathrm{site}}(t-n-182) & \theta_{\mathrm{site}}(t-n-183) & \theta_{\mathrm{site}}(t-2n+3) & \theta_{\mathrm{site}}(t-2n+2) & \theta_{\mathrm{site}}(t-2n+1) \end{bmatrix}$$

Measurements for PM_{10} when site $\in \{\mathrm{PA}, \mathrm{PE}, \mathrm{TA}, \mathrm{WH}\}$

$$\boldsymbol{\Xi}_{\mathrm{site}}(t) = \begin{bmatrix} \xi_{\mathrm{site}}(t) & \xi_{\mathrm{site}}(t-1) & \cdots & \xi_{\mathrm{site}}(t-10) & \xi_{\mathrm{site}}(t-182) & \xi_{\mathrm{site}}(t-183) & \xi_{\mathrm{site}}(t-184) & \xi_{\mathrm{site}}(t-n+2) & \xi_{\mathrm{site}}(t-n+1) & \xi_{\mathrm{site}}(t-n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{\mathrm{site}}(t-n+1) & \xi_{\mathrm{site}}(t-n) & \ldots & \xi_{\mathrm{site}}(t-n-9) & \xi_{\mathrm{site}}(t-n-181) & \xi_{\mathrm{site}}(t-n-182) & \xi_{\mathrm{site}}(t-n-183) & \xi_{\mathrm{site}}(t-2n+3) & \xi_{\mathrm{site}}(t-2n+2) & \xi_{\mathrm{site}}(t-2n+1) \end{bmatrix}$$

(**c**)

Figure 8. Air pollution data—Response vector and the matrix of predictors for Scenario A (FullSet) and Scenario B (ConSet). (**a**) The blue and purple boxes represent the blocks of predictors. The notation $[\boldsymbol{\Theta}_{\mathrm{site}}(t)]_{:1}$ stands for the first column of the matrix $\boldsymbol{\Theta}_{\mathrm{site}}(t)$. A more detailed description of the predictors is provided in the panels below. (**b**) Scenario A—Full set of predictors (FullSet): The entries of the matrix in purple $\boldsymbol{\Theta}_{\mathrm{site}}(t)$ are the log-transformed $\mathrm{PM}_{2.5}$ measurements collected during the past n days, where $n = 365$. Similarly, the entries of the matrix in blue $\boldsymbol{\Xi}_{\mathrm{site}}(t)$ are the log-transformed PM_{10} measurements collected during the past 365 days. (**c**) Scenario B—Constrained set of predictors (ConSet): The entries of the matrix in purple $\boldsymbol{\Theta}_{\mathrm{site}}(t)$ are the log-transformed $\mathrm{PM}_{2.5}$ measurements. When the estimation is made for the day t, the predictors contain recent measurements (from days $t, t-1, \ldots, t-10$) as well as measurements collected about six months ago (on days $t-182, t-183, t-184$) and about one year ago (on days $t-363, t-364, t-365$). The entries of the matrix in blue $\boldsymbol{\Xi}_{\mathrm{site}}(t)$ are the log-transformed PM_{10} measurements collected at the time points t to $t-10$, $t-182$ to $t-184$, and $t-363$ to $t-365$.

The values of NMSE computed by applying BPA are presented on the third row of Table 1. It is evident that the ConSet scenario always leads to a smaller NMSE than the

NMSE for the FullSet scenario. This is expected because in ConSet we use prior knowledge to pre-select the useful predictors. To gain more insight, we comment briefly on the top three most frequently selected predictors in N_{TR} runs for the different sites, in each scenario. We note that PM_{10} from the same site on the present day is always the top voted predictor and is always selected in each of the N_{TR} runs. The selection of this predictor can be easily understood because $PM_{2.5}$ is a subset of PM_{10}, so the concentration of $PM_{2.5}$ inevitably correlates with the PM_{10} measurement on the same day, at the same site. Then, the $PM_{2.5}$ measurements on the same day for the other sites are usually the second and third most voted predictors. This is also expected because the $PM_{2.5}$ concentration on the site that we want to predict most likely correlates with the $PM_{2.5}$ measurements recorded for the other three sites on the same day. The only exception is for the ConSet scenario in Penrose, where the third most voted predictor is the $PM_{2.5}$ measurement from Takapuna one day before the present day. This selection is reasonable because Penrose and Takapuna have some similarities (see [5,8]). This analysis is only descriptive and does not suggest any causal relationship between the predictors and the $PM_{2.5}$ measurements.

On the fourth row of Table 1 are shown the results of comparison with the NMSE's computed in [8], where the same problem for the Auckland air pollution data set was solved by using greedy algorithms and IT criteria/cross-validation. We mentioned in Section 1 that the greedy algorithms employed in [8] are MPA, OMP, RMP, FWA and CMP. In the same section, we pointed out that 22 IT criteria have been used in conjunction with MPA. As cross-validation was also used for MPA, it means that the total number of selection rules for MPA was 23. Similarly, the total number of selection rules for the other four greedy algorithms are: OMP–13, RMP–23, FWA–13 and CMP–13. Hence, the NMSE values produced by BPA for each site and for each scenario are compared with the results yielded by other 85 methods. Given the large number of NMSE's that are considered, we compute the deciles for assigning different levels of performance to them.

Table 1. The values of NMSE (in percentages), which are computed by applying the formula in (14), are shown on the third row of the table. For computing the deciles that are reported on the fourth row, for each site and for each scenario, we rank from lowest to highest 86 values of NMSE. One of these values is the NMSE reported on the third row of the table and the other 85 are the NMSE values obtained in [8] by employing various greedy algorithms and IT criteria/cross-validation. The best ranked methods are those that are assigned to the decile D1 because they produce the smallest 10% NMSEs. Similarly, the methods assigned to D2 yield the smallest 20% NMSEs, the methods assigned to D3 yield the smallest 30% NMSEs, and so on.

Site	**Patumahoe**		**Penrose**		**Takapuna**		**Whangaparaoa**	
Scenario	FullSet	ConSet	FullSet	ConSet	FullSet	ConSet	FullSet	ConSet
NMSE	5.61%	5.59%	3.75%	3.60%	6.05%	5.73%	3.17%	3.15%
Decile	D2	D7	D1	D6	D2	D10	D1	D8

We can see in Table 1 that the performance of BPA is very good for the FullSet scenario, where BPA is assigned either to D1 (top 10% methods) or to D2 (top 20% methods). This supports the argument that BPA is suitable for the high-dimensional FullSet scenario. This observation is perfectly in line with the theoretical grounds on which BPA was derived. However, the ranking of BPA is modest for the ConSet scenario, where the number of the available data is larger than the total number of the predictors.

The experimental results can be reproduced by using the Matlab code (accessed on 25 June 2023) available at https://www.stat.auckland.ac.nz/%7Ecgiu216/PUBLICATIONS.htm.

5. Conclusions, Limitations, and Future Research

In this work, we have analyzed the applicability of BPA to simulated and real-life time series data sets. During the testing of the algorithm on artificial data with various noise

levels, we have found that BPA is quite robust. Based on the experiments with simulated data, where we have investigated various parameter settings for BPA, we concluded that the parameters q/p and ϕ are the most influential on BPA's performance. Parameter q/p should be set at a medium level to allow predictors to pass through the first layer, whereas ϕ should be set at a relatively high level for having an effective selection of the predictors. We have used our findings from the simulated data experiment in the experiment with air pollution data. BPA performed very well in the FullSet scenario ($p \gg n$) of this experiment on real-life data, where it was compared with 85 methods that have been previously evaluated in [8]. At the same time, the results for the ConSet scenario suggest that BPA should not be used when $n > p$.

An area where the performance of BPA can be further investigated is to evaluate the influence of the base selection procedure. In this work, we focused on Lasso because of the popularity of Lasso in the signal processing community. It might also be interesting to extend the application of BPA to other signal processing problems that are not related to the classical VARX model.

Author Contributions: Conceptualization, V.D. and C.D.G.; methodology, V.D. and C.D.G.; software, V.D. and C.D.G.; validation, V.D.; investigation, V.D.; data curation, V.D.; writing—original draft preparation, V.D.; writing—review and editing, V.D. and C.D.G.; supervision, C.D.G.; project administration, C.D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available at the same web address where the Matlab code can be found (accessed on 7 July 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Efron, B.; Hastie, T. *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*; Institute of Mathematical Statistics Monographs, Cambridge University Press: Cambridge, UK, 2016. [CrossRef]
2. Bühlmann, P.; van de Geer, S. *Statistics for High-Dimensional Data. Methods, Theory and Applications*; Springer: Berlin/Heidelberg, Germay, 2011.
3. Mallat, S.; Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415. [CrossRef]
4. Sancetta, A. Greedy algorithms for prediction. *Bernoulli* **2016**, *22*, 1227–1277. [CrossRef]
5. Li, F.; Triggs, C.; Dumitrescu, B.; Giurcăneanu, C. The matching pursuit revisited: A variant for big data and new stopping rules. *Signal Process.* **2019**, *155*, 170–181. [CrossRef]
6. Sturm, B.; Christensen, M. Comparison of orthogonal matching pursuit implementations. In Proceedings of the Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 220–224.
7. Barron, A.; Cohen, A.; Dahmen, W.; DeVore, R. Approximation and learning by greedy algorithms. *Ann. Stat.* **2008**, *36*, 64–94. [CrossRef]
8. Li, F.; Triggs, C.; Giurcăneanu, C. On the selection of predictors by using greedy algorithms and information theoretic criteria. *Aust. N. Z. J. Stat.* **2023**. [CrossRef]
9. Frank, M.; Wolfe, P. An algorithm for quadratic programming. *Nav. Res. Logist. Q.* **1956**, *1*, 95–110. [CrossRef]
10. Tibshirani, R. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58*, 267–288. [CrossRef]
11. Frandi, E.; Ñanculef, R.; Lodi, S.; Sartori, C.; Suykens, J. Fast and scalable Lasso via stochastic Frank–Wolfe methods with a convergence guarantee. *Mach. Learn.* **2016**, *104*, 195–221. [CrossRef]
12. Zhao, Y.; Huo, X. A survey of numerical algorithms that can solve the Lasso problems. *WIREs Comput. Stat.* **2022**, e1602. [CrossRef]
13. Freijeiro-González, L.; Febrero-Bande, M.; González-Manteiga, W. A Critical Review of LASSO and Its Derivatives for Variable Selection under Dependence among Covariates. *Int. Stat. Rev.* **2022**, *90*, 118–145. [CrossRef]
14. Meinshausen, N.; Bühlmann, P. Stability selection (with discussion). *J. R. Stat. Soc. Ser. B* **2010**, *72*, 417–473. [CrossRef]
15. Shah, R.; Samworth, R. Variable selection with error control: Another look at stability selection. *J. R. Stat. Soc. Ser. B* **2013**, *75*, 55–80. [CrossRef]
16. Bijral, A. On Selecting Stable Predictors in Time Series Models. *arXiv* **2019**, arXiv:1905.07659. https://doi.org/10.48550/arXiv.1905.07659.
17. Yu, B. Stability. *Bernoulli* **2013**, *19*, 1484–1500. [CrossRef]

18. Giurcăneanu, C.; Tabus, I. Cluster structure inference based on clustering stability with applications to microarray data analysis. *Eurasip J. Adv. Signal Process.* **2004**, 545761. [CrossRef]
19. Liu, T.; Yu, H.; Blair, R.H. Stability estimation for unsupervised clustering: A review. *WIREs Comput. Stat.* **2022**, *14*, e1575. [CrossRef] [PubMed]
20. Maddu, S.; Cheeseman, B.; Sbalzarini, I.; Müller, C. Stability selection enables robust learning of differential equations from limited noisy data. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2022**, *478*, 20210916. [CrossRef] [PubMed]
21. Lu, D.; Weljie, A.; de Leon, A.; McConnell, Y.; Bathe, O.; Kopciuk, K. Performance of variable selection methods using stability-based selection. *BMC Res. Notes* **2017**, *10*, 143. [CrossRef] [PubMed]
22. Hyde, R.; O'Grady, L.; Green, M. Stability selection for mixed effect models with large numbers of predictor variables: A simulation study. *Prev. Vet. Med.* **2022**, *206*, 105714. [CrossRef] [PubMed]
23. Lutkepöhl, H. *New Introduction to Multiple Time Series Analysis*; Springer: Berlin/Heidelberg, Germany, 2005.
24. Dumitrescu, B.; Giurcăneanu, C.; Ding, Y. Identification of vector autoregressive models with Granger and stability constraints. In Proceedings of the 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2–6 September 2019; pp. 1–5. [CrossRef]