Data Mining and Machine Learning for Environmental Systems Modelling and Analysis

Thesis by

Jose Roberto Ayala Solares, B.Sc., M.Sc.

Submitted for the degree of

Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Automatic Control and Systems Engineering

Sheffield, England, United Kingdom

September, 2017

Acknowledgments

I would like to sincerely thank my supervisor Dr. Hua-Liang Wei for his invaluable support and encouragement throughout the course of this research. It was an honor to work with him. His enthusiasm and valuable feedback for research made my study very enjoyable, exciting and ultimately fruitful with rich experience. I also thank Prof. Grant Bigg and Prof. Michail Balikhin for their extensive support, advice and guidance in this journey.

You will teach to fly, but they will not fly your dream.
You will teach to live, but they will not live your life.
However, in every flight, in every life, in every dream
it will remain the footprint of the path that you taught.

Mother Teresa of Calcutta

I thank all my friends, past and present, that have made of Sheffield a special place. In particular, I want to thank Brigette Lee, Fedon Moudopoulos, Arnav Vishwasrao, Carlos Luna, Sachin Prabhu, and Matei Neagu for expanding my worlds and making of this a memorable experience.

Each friend represents a world in us,

a world possibly not born until they arrive,

and it is only by this meeting

that a new world is born.

Anaïs Nin

I thank my parents and brothers for their continuous encouragement and support. Los llevo siempre en mi pensamiento y los amo con todo mi ser.

When you look at your life,

the greates happinesses are family happinesses.

Joyce Brothers

Lastly, I would like to thank the Mexican National Council of Science and Technology (CONACYT) for providing support and resources for this research work.

The world needs dreamers and the world needs doers.

But above all, the world needs dreamers who do.

Sarah Ban Breathnach

$R^2\infty$

ABSTRACT

Data Mining and Machine Learning for Environmental Systems Modelling and Analysis

Jose Roberto Ayala Solares

This thesis provides an investigation of environmental systems modelling and analysis based on system identification techniques. In particular, this work focuses on adapting and developing a new Nonlinear AutoRegressive with eXogenous inputs (NARX) framework, and its application to analyse some environmental case studies. Such a framework has proved to be very convenient to model systems with nonlinear dynamics because it builds a model using the Orthogonal Forward Regression (OFR) algorithm by recursively selecting model regressors from a pool of candidate terms. This selection is performed by means of a dependency metric, which measures the contribution of a candidate term to explain a signal of interest.

For the first time, this thesis introduces a package in the R programming language for the construction of NARX models. This includes a set of features for effectively performing system identification, including model selection, parameter estimation, model validation, model visualisation and model evaluation. This package is used extensively throughout this thesis.

This thesis highlights two new components of the original OFR algorithm. The first one aims to extend the deterministic notion of the NARX methodology by introducing the distance correlation metric, which can provide interpretability of nonlinear dependencies, together with the bagging method, which can provide an uncertainty analysis. This implementation produces a bootstrap distribution not

only for the parameter estimates, but also for the forecasts. The biggest advantage is that it does not require the specification of prior distributions, as it is usually done in Bayesian analysis.

The NARX methodology has been employed with systems where both inputs and outputs are continuous variables. Nevertheless, in real-life problems, variables can also appear in categorical form. Of special interest are systems where the output signal is binary. The second new component of the OFR algorithm is able to deal with this type of variable by finding relationships with regressors that are continuous lagged input variables. This improvement helps to identify model terms that have a key role in a classification process.

Furthermore, this thesis discusses two environmental case studies: the first one on the analysis of the Atlantic Meridional Overturning Circulation (AMOC) anomaly, and the second one on the study of global magnetic disturbances in near-Earth space.

Although the AMOC anomaly has been studied in the past, this thesis analyses it using NARX models for the first time. The task is challenging given that the sample size available is small. This requires some preprocessing steps in order to obtain a feasible model that can forecast future AMOC values, and hindcast back to January of 1980.

In the second case study, magnetic disturbances in near-Earth space are studied by means of the Kp index. This index goes from 0 (very quiet) to 9 (very disturbed) in 28 levels. There is special interest in the forecast of high magnetic disturbances given their impact on terrestrial technology and astronauts' safety, but these events are rare and therefore, difficult to predict. Two approaches are analysed using the NARX methodology in order to assess the best modelling strategy. Although this phenomenon has been studied with other techniques providing very promising results, the NARX models are able to provide an insightful relationship of the Kp index to solar wind parameters, which can be useful in other geomagnetic analyses.

Table of Contents

List of Algorithms

List of Figures

List of Tables

Nomenclature

n_u	maximum lag for input sequence
n_y	maximum lag for output sequence
AIC	Akaike Information Criterion
AICc	Corrected Akaike Information Criterion
AMOC	Atlantic Meridional Overturning Circulation
ANOVA	Analysis of Variance
BFOR-dCor	Bagging Forward Orthogonal Regression using distance Correlation
BIC	Bayesian Information Criterion
CRAN	Comprehensive R Archive Network
CV	Cross-Validation
EEG	Electroencephalography
ERR	Error Reduction Ratio
ESR	Error-to-Signal Ratio
GFRF	Generalised Frequency Response Function
IEDSS	Intelligent Environmental Decision Support System
LARS	Least Angle Regression

LASSO	Least Absolute Shrinkage and Selection Operator
LOOCV	Leave-One-Out Cross-Validation
MAE	Mean Absolute Error
ME	Mean Error
MI	Mutual Information
MIC	Maximal Information Coefficient
MINE	Maximal Information-based Nonparametric Exploration
MPO	Model Predicted Output
NAR	Nonlinear AutoRegressive
NARX	Nonlinear AutoRegressive with eXogenous inputs
OFR	Orthogonal Forward Regression
OSA	One-Step Ahead
PCA	Principal Component Analysis
PCR	Principal Component Regression
PE	Prediction Efficiency
PRESS	Predicted Residual Sum of Squares
RJMCMC	Reversible Jump Markov Chain Monte Carlo
RMSE	Root Mean Squared Error
SE	Standard Error
WN	Wavelet Network

Chapter 1

Introduction

1.1 Background

A great amount of physical phenomena around the world can be described through signals. A signal is defined as a function that contains information about the behaviour of a phenomenon. This information is contained within patterns that vary in time and/or space [1].

Signals interact with systems, which can be described as objects whose components synergy results in an observable output within an environment [2]. Therefore, a system can be seen as a process that receives input signals and produces other output signals [1]. This interaction between signals and systems is important because it helps us to understand the dynamics of the system. In fact, nature is full of a large and rich variety of systems that possess certain behaviour that have captured the attention of humans for centuries.

Traditionally, the scientific approach to understand the dynamics of a system consists of recording a series of observations from the system, and then generating a hypothesis that tries to explain the behaviour of such dynamics [3]. Most of the time, this hypothesis takes the form of a mathematical model that maps input attributes to output values [4]. Mathematical models are a fundamental notion in many branches of science and technology [5].

One of the main advantages of the technology of the 21st century is the ability to acquire, store, retrieve and distribute great quantities of data from almost any system in every field or discipline. In fact, 90% of the world's data have been generated over the last 2 years at rates that have no precedent in the history of mankind [3, 6, 7]. Most of the time, these data contain valuable information that it is difficult to translate into relevant knowledge [8, 9].

In recent years, a change in the scientific process has been taking place, where huge amounts of data have enabled the construction of empirical or data-based models in tasks like financial forecasting, medical diagnosis, computer vision, network traffic analysis, weather prediction, on-demand Internet streaming media, astronomy, detection of dark matter in space, among others, where an analytical solution is difficult to obtain [3, 7, 10, 11]. The main assumption in these data-based models is that the behaviour of the system should, in principle, be recoverable from input-output measurements [12]. This process is commonly referred to as *knowledge mining* or *learning from data*.

Learning from data has derived in a variety of branches with a strong foundation in mathematics, statistics and computer sciences. The two most popular branches are *Data Mining* and *Machine Learning*. These two topics have considerable overlap among them. The central task is to discover or learn insightful patterns from an observational data set [3, 13, 14]. The difference lies in the amount of data they handle. Data mining puts an emphasis on data sets that are huge [13, 15].

Although the use of data provides a different perspective to solve problems compared with more conservative approaches where the underlying physics that governs a system under study is used, learning from data is a non-trivial task. Despite all the advances in data gathering, the gap between the generation of data and our understanding of them increases as well [16], and several works have focused only on looking for patterns in data without considering pre- and post-processing steps that can improve considerably the data mining and learning tasks [14, 17]. Nowadays, there is an ongoing effort to extract useful information from large data sets [3].

Among the several techniques available for data modelling, one of the most popular approaches is the use of Nonlinear AutoRegressive with eXogenous inputs (NARX) models. These are nonlinear recursive difference equations that find a mapping between lagged explanatory variables and a variable of interest. Such an approach has been implemented mainly in MATLAB, although nowadays there are several open-source alternatives (like the R programming language) that have gained popularity due to the flexibility and ease of use to solve data modelling problems.

Data mining and machine learning have been applied successfully in business-related problems. Such a success has been extended to other areas however, until recent years, data mining and machine learning techniques started to be used to analyse and understand environmental systems.

The term *environmental system* should be understood in a broad and inclusive sense as it can be referred to any systems, such as geo-, hygro- and ocean-environmental, or space weather, that could affect our environment [18]. Environmental systems involve an interaction of biological, physical, chemical, geological, ecologic, climatic, and social processes, among others [19]. The analysis of such systems is important because it can improve decision making in environmental management for the development, implementation and maintenance of environmental protection policies, or control design for systems that interact with environmental variables [20–22]. Nevertheless, the high complexity of these systems limits the formulation of mathematical theories or deterministic models. Furthermore, environmental systems are highly nonlinear, interact at different spatial and temporal scales, and evolve over time so the stationarity assumption does not hold [14].

1.2 Motivation

Although several environmental data sets are available, there are scenarios where data collection is expensive and difficult, e.g. tropical deforestation [23], wildfires [20], wastewater treatment plants [24], Atlantic Ocean's major current circulation [25], geomagnetic disturbances in near-Earth space [26], among many others. Furthermore, traditional statistical analysis does not work in either limited, or vast and complex data sets [20]. Environmental data can come from various sources and feature complex spatial patterns at different scales due to combination of several spatial phenomena or various influencing factors of different origins [22]. In some cases, the original observations are taken with significant measurements errors and may contain significant uncertainty as well as a number of outliers or missing values. In addition, the spatial and temporal sampling may not capture the inherent behaviour of an environmental system. Therefore, new techniques are required to deal with the high complexity of data from environmental processes. Among the challenges that need to be tackled, there is a need to improve automated pre- and post-processing techniques, develop algorithms that can perform an online learning, combine existing techniques that can handle the difficulties of environmental data, find ways that can fuse data with existing knowledge, and develop effective mechanisms that merge the strengths of human cognition with those of the data mining and learning algorithms [14, 19, 27, 28].

The research in this thesis focuses mainly on environmental scenarios where limited data are available (as opposed to big data problems with mega- or gigabytes of data). Traditional machine learning algorithms can handle such scenarios, although most of them have difficulties to handle time-variant information [29], and are unable to provide a good understanding of the inner dynamics of a system [5], which is usually of great interest in environmental problems. To overcome such issues, in this work the NARX methodology is applied and further extended to provide interpretability of nonlinear dependencies, uncertainty analysis, and to handle both continuous and categorical data, which are common in environmental systems [22].

1.3 Overview

This thesis is organised as follows:

- Chapter 2 provides an in-depth review of the main concepts in machine learning and system identification that are used throughout this thesis. A special emphasis is put into the NARX model and the Orthogonal Forward Regression algorithm, along with a broad discussion of different techniques that have been developed to identify systems. This chapter also provides an overview of how system identification techniques have been applied to the modelling and analysis of environmental systems.

- Chapter 3 describes the R programming language and its use for data analysis and system identification. The newly developed NARX R package is also discussed, together with improvements to the original Orthogonal Forward Regression algorithm. Some examples are shown that highlight the usefulness of this package to build NARX models.

- Chapter 4 deals with two improvements to the NARX methodology: provide interpretability of nonlinear dependencies, and accommodate uncertainties in the parameter estimates, as well as the identified model and the computed predictions. For the first case, the distance correlation metric is implemented, which is a new metric able to detect all types of nonlinear or non-monotone dependencies between random vectors. For the second case, the bagging method is used, which runs an algorithm several times on resampled data and the results obtained are combined to predict a numerical value via averaging (for regression problems) or via voting (for classification problems). The new scheme is referred as Bagging Forward Orthogonal Regression using distance Correlation (BFOR-dCor) algorithm.

- Chapter 5 investigates the Atlantic Meridional Overturning Circulation. This is an interesting real-case scenario where the NARX R package described in Chapter 3 is used to identify a model that is able to hindcast and forecast northward flow in the upper layer, and southward flow in the deep ocean of the Atlantic. Furthermore, the most important regressor is analysed by means of the BFOR-dCor algorithm described in Chapter 4.

- Chapter 6 proposes a novel approach that combines logistic regression with the NARX methodology. This enables the construction of NARX models that can be used for binary classification problems. The Orthogonal Forward Regression algorithm is adapted for this purpose, and the biserial correlation coefficient is defined, which measures the strength of the association between a continuous variable and a dichotomous variable.

- Chapter 7 investigates global magnetic disturbances in near-Earth space using NARX models. The main objective is the understanding and analysis of the dependent relationship of the Kp index on solar wind speed and dynamic pressure variables. Two approaches are explored. The first one consists of a recursive sliding window scheme in which a window of a given length is used to train a model and to forecast future values based on previous predictions. The second approach involves the identification of a specific model for a horizon of interest. In addition, the logistic NARX approach, described in Chapter 6, is tested in a binary version of the Kp index.
- Chapter 8 concludes the work done in this thesis, and provides suggestions for future directions of research.

1.4 Contributions

This work aims to investigate data mining and machine learning approaches, and develop new data-driven modelling methods and algorithms that can be used for environmental system analysis.

The following are the main contributions of this thesis to the scientific community:

1. Chapter 3: *an R package for the construction of NARX models*. Given the lack of software with a standard library for building NARX models, this R package is the first one developed for this purpose. It eases the task of building NARX models while providing a set of features for effectively performing model selection, parameter estimation, model validation, model visualisation and model

evaluation. The package implements the traditional Orthogonal Forward Regression algorithm together with several improvements that include nonlinear dependency metrics, and methods for selecting the appropriate number of model terms. This package is an invaluable tool and its usefulness is shown throughout this thesis.

2. Chapter 4: *a novel bagging method based on distance correlation metric for nonlinear model structure detection and parameter estimation.* In general, the commonly used dependency metrics such as correlation function and mutual information may not work well in some cases. Furthermore, there are always uncertainties in model parameter estimates. Thus, a new approach is proposed to overcome this by using a distance correlation metric incorporated with a bagging method. The combination of these two features enhances the performance of existing forward selection approaches in that it provides the interpretability of nonlinear dependency and an insightful uncertainty analysis for model parameter estimates. The results of this chapter were published in [30].

3. Chapter 5: *forecast and hindcast of the Atlantic Meridional Overturning Circulation (AMOC).* The NARX methodology is applied for the first time to the analysis of the AMOC. Significant regressors that contribute to the explanation of this phenomenon are identified. These suggest that the difference in density between the deep-water formation areas and the upstream Gulf Stream source region seven months ago provide the best indication of variation in the AMOC strength.

4. Chapter 6: *a novel logistic NARX methodology that allows the use of NARX models to analyse binary classification problems.* In most cases, NARX models are applied to regression problems where all variables involved are continuous, and little attention has been paid to classification problems where the output signal is a binary sequence. Therefore, this novel classification algorithm

combines the NARX methodology with logistic regression and the proposed method is referred to as logistic NARX model. Such a combination is advantageous since the NARX methodology helps to deal with the multicollinearity problem while the logistic regression produces a model that predicts categorical outcomes. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models where users can identify which input variables play an important role individually and/or interactively in the classification process, something that is not achievable using other classification techniques. The results of this chapter were published in [29].

5. Chapter 7: *information about the relative contributions of solar wind speed and dynamic pressure to the changes in the Kp index.* Although previous studies have confirmed the role of solar wind speed and dynamic pressure as drivers of the Kp index, a new analysis is performed that provides further information of the relationship of the Kp index to solar wind using NARX models. The analysis also highlights a bias issue that is the result of the uneven distribution in the Kp index data, and the use of a regression model to predict a categorical output variable. The results of this chapter were published in [31].

1.4.1 Refereed Journals

The present research has been published in several journal papers, which are listed below:

- J. R. Ayala Solares & H.-L. Wei. "Nonlinear model structure detection and parameter estimation using a novel bagging method based on distance correlation metric", Nonlinear Dynamics (2015), 82, 201-215.

- J. R. Ayala Solares, H.-L. Wei, R. J. Boynton, S. N. Walker, and S. A. Billings. "Modeling and prediction of global magnetic disturbance in near-Earth space: A case study for Kp index using NARX models", Space Weather (2016), 14, 899–916.

- J. R. Ayala Solares, H.-L. Wei and S. A. Billings. “A novel logistic-NARX model as a classifier for dynamic binary classification”, Neural Computing and Applications (2017). Neural Computing and Applications (2017), 1-15.
- J. R. Ayala Solares, H.-L. Wei and G. Bigg. “The variability of the Atlantic Meridional Circulation since 1980, as hindcast by a systems model”, Journal of Geophysical Research Oceans (2017). Under review.

1.4.2 Peer-Reviewed Conference

The present research was presented at the following conference:

- J. R. A. Solares and H.-L. Wei. “A New Distance Correlation Metric and Bagging Method for NARX Model Estimation.” In: The University of Sheffield Engineering Symposium Conference Proceedings, Vol. 1, 2014.

1.4.3 Presentation

The present research was presented at the following seminar:

- J. R. Ayala Solares. “Introduction to R for Data Analysis and System Identification.” In: The University of Sheffield Automatic Control and Systems Engineering Students Seminar, 2016.

Chapter 2

General Concepts

2.1 Introduction

The acquisition of data in a vast diversity of fields has increased in recent years. This has allowed several research areas to tackle problems from a data-based modelling approach. In particular, system identification has benefited from this idea where traditionally a mathematical model was derived based on a comprehensive physical insight of all the events that take part in a system. Such a comprehensive approach may be intractable to obtain given the difficulty to fully describe the intrinsic mechanics of a system.

This chapter provides an in-depth review of the data deluge phenomenon and why it is possible to learn from data. It greatly focuses on system identification putting special attention to the NARX model and the Orthogonal Forward Regression algorithm, along with a broad discussion of different techniques that have been developed to perform model structure selection, parameter estimation and model validation. Finally, an overview of how system identification techniques have been applied to the modelling and analysis of environmental systems is given.

2.2 Data Deluge

In recent years, the acquisition and storage of data across many disciplines have become easier. In fact, data are being generated in every field of study at enormous rates never seen before in the history of mankind. Data are the collection of quantitative and/or qualitative values that belong to a certain population [32]. One of the main concerns for scientists, researchers, businessmen, among others, is how to extract useful information or knowledge buried within the data. Several authors have stated a series of steps that should be followed when dealing with this issue [3, 14]:

1. Problem formulation: it does not matter how much data are available if there is not a clear question in mind that may be answered by analysing the data. This step requires the understanding of the domain under study and the acquisition of useful prior knowledge that contributes towards the goal of the end-user.

2. Data manipulation: collected data come in raw form, which are difficult to analyse. This step transforms raw data into tidy data by a series of minor actions that correspond to data cleaning, pre-processing, reduction and projection [33].

3. Knowledge discovery: depending on the problem and the goal, a variety of different algorithms can be applied to mine knowledge from the data. This step requires the selection of the algorithm and tuning of its parameters.

4. Post-processing: transformation of the results so that they are meaningful to the end-user.

2.3 The Learning Problem and Feasibility of Learning

Machine Learning is an important problem that focuses on approximating observed data and generalising from it to unobserved data [10]. In essence, data mining and machine learning work with problems where [10,34]:

1. a pattern exists

2. we can not pin them down mathematically

3. we have data about them

Broadly speaking, data-based problems can be classified in different types. The main learning paradigms are as follows:

Supervised Learning focuses on learning input-output mappings from data. It can be divided into regression and classification problems [35].

Unsupervised Learning focuses on learning structure on data where the output is unknown. The main techniques are clustering and projection [4].

Reinforcement Learning focuses on learning a sequence of actions that will maximise a reward [34].

In this thesis, the research is restricted to *supervised learning* problems. These are described using the following mathematical components:

- Input $\mathbf{x}$: a D-dimensional vector where D corresponds to the number of attributes.

- Output y: a scalar quantity.

- Target function $f : \mathcal{X} \rightarrow \mathcal{Y}$: unknown function that maps from the input domain $\mathcal{X}$ to the output domain $\mathcal{Y}$.

- Hypothesis $g : \mathcal{X} \rightarrow \mathcal{Y}$: created formula that approximates the target function.

- Data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)$: examples that will be used to learn the hypothesis.

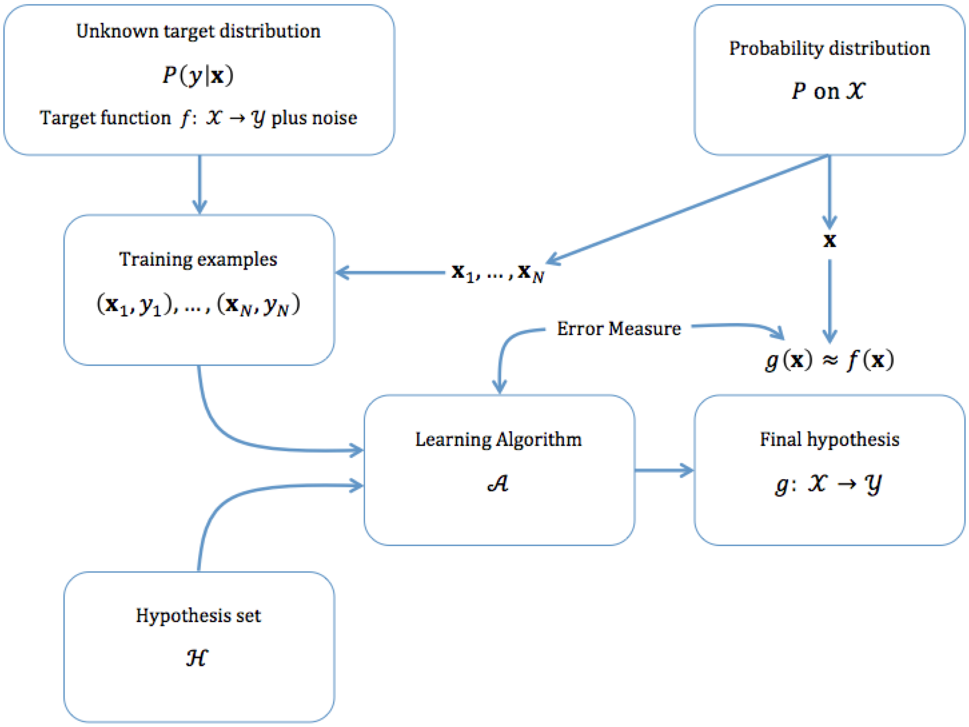
Unknown target distribution
$P(y|\mathbf{x})$
Target function $f: \mathcal{X} \to \mathcal{Y}$ plus noise
Probability distribution
P on $\mathcal{X}$
Training examples
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
$\mathbf{x}_1, \ldots, \mathbf{x}_N$
$\mathbf{x}$
Error Measure
$g(\mathbf{x}) \approx f(\mathbf{x})$
Learning Algorithm
$\mathcal{A}$
Final hypothesis
$g: \mathcal{X} \to \mathcal{Y}$
Hypothesis set
$\mathcal{H}$

Figure 2.1: Learning Diagram. The two components of the learning process that we are interested in are the hypothesis set, which will contain the final hypothesis, and the learning algorithm that identifies such a final hypothesis. Original source: [10].

Figure 2.1 shows the connections of the components of learning. The target function remains unknown during the whole learning process. The only information that is available is through the training examples collected. These examples will go into a learning algorithm, which will select the final hypothesis from a set of candidate hypotheses. It is assumed that the input vectors $\mathbf{x}$ are generated independently and identically distributed from a fixed probability distribution [36]. This is an important feature that quantifies the relative importance of the input vector $\mathbf{x}$ and makes learning feasible [34]. Also, the values of the output variable y are assumed to come from a conditional probability distribution on the input vectors $\mathbf{x}$ to account for the fact that, in real-life applications, there is noise in the output values. The two components of the learning process that we are interested in are the hypothesis set, which will contain the final hypothesis, and the learning algorithm that identifies such a final hypothesis. The error measure is a quantity of the performance of the learning algorithm or the final hypothesis. The error measure on the learning algorithm is known as the in-sample error while on the final hypothesis is known as out-of-sample error. The goal of the learning process is to select a hypothesis from the hypothesis set $\mathcal{H}$ with the best in-sample performance and hope it generalizes well in out-of-sample. In general, the generalisation depends on the size and quality of the training examples [10, 34].

2.4 System Identification

System identification is a challenging and interesting engineering problem that has been extensively studied for decades. It is an experimental approach that aims to identify and fit a mathematical model of a system based on experimental data that record the system inputs and outputs behaviour [5, 37, 38]. It is assumed that the mathematical model that is being searched is a well-behaved function that is consistent with the data. Linear system identification has been extensively used in past years, however, its applicability is limited since the linearity assumption is strict and in real-life, most of the systems of interest are nonlinear [39]. Extensive research has been developed in the nonlinear realm for system identification since the 1980s [5]. Some of the most popular models for nonlinear system identification include: piecewise linear models, Volterra series models, generalised additive models, neural networks, wavelet models, and state-space models. The reader is referred to [5, 40] for a broad discussion on these models. In particular, the Nonlinear AutoRegressive with eXogenous inputs (NARX) methodology has proved to be a well-suited scheme for nonlinear system identification problems [5, 12].

In general, system identification consists of three steps: Model Structure Detection, Parameter Estimation, and Model Validation [12, 37, 38, 41].

2.4.1 The NARX model

The NARX model is a nonlinear recursive difference equation with the following general form:

$$y(k) = f\Big(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)\Big) + e(k) \tag{2.1}$$

where $f(\cdot)$ represents an unknown nonlinear mapping; $y(k)$, $u(k)$ and $e(k)$ are the output, input and prediction error sequences with $k = 1, 2, \dots, N$; N is the number of observations, and the maximum lags for the output and input sequences are n_y and n_u, respectively [42]. Most approaches assume that the function $f(\cdot)$ can be approximated by a linear combination of a predefined set of functions $\phi_i\Big(\boldsymbol{\varphi}(k)\Big)$, therefore Eq. (2.1) can be expressed in a linear-in-the-parameters form

$$y(k) = \sum_{i=1}^{m} \theta_i \phi_i\Big(\boldsymbol{\varphi}(k)\Big) + e(k) \tag{2.2}$$

where θ_i are the model parameters, $\phi_i\Big(\boldsymbol{\varphi}(k)\Big)$ are the predefined functions that depend on the regressor vector of past outputs and inputs $\boldsymbol{\varphi}(k) = \Big[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)\Big]^T$, and m is the number of functions in the set [12]. One of the most commonly used NARX models is the polynomial NARX representation, where Eq. (2.2) can be explicitly written as

$$\begin{aligned} y(k) \quad = \quad & \theta_0 + \textstyle\sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n} \sum_{i_2=i_1}^{n} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + \cdots \\ & + \textstyle\sum_{i_1=1}^{n} \cdots \sum_{i_\ell=i_{\ell-1}}^{n} \theta_{i_1 i_2 \dots i_\ell} x_{i_1}(k) x_{i_2}(k) \dots x_{i_\ell}(k) + e(k) \end{aligned} \tag{2.3}$$

where

$$x_i(k) = \begin{cases} y(k-i) & 1 \le i \le n_y \\ u(k-i+n_y) & n_y+1 \le i \le n = n_y + n_u \end{cases} \tag{2.4}$$

and ℓ is the nonlinear degree of the model. A NARX model of order ℓ means that the order of each term in the model is not higher than ℓ. The total number of potential terms in a polynomial NARX model is given by

$$M = \begin{pmatrix} n+\ell \\ \ell \end{pmatrix} = C_{\ell}^{n+\ell} = \frac{(n+\ell)\,!}{n!\cdot\ell!} \tag{2.5}$$

where $n = n_y + n_u$. Equation (2.2) can be rewritten in a vector form as

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\theta} + \mathbf{e} \tag{2.6}$$

where

$$\mathbf{y} = \begin{bmatrix} y(1) & y(2) & \dots & y(N) \end{bmatrix}^T$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_m \end{bmatrix}^T$$

$$\mathbf{e} = \begin{bmatrix} e(1) & e(2) & \dots & e(N) \end{bmatrix}^T$$

$$\mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \dots & \boldsymbol{\phi}_m \end{bmatrix} = \begin{bmatrix} \phi_1\Big(\boldsymbol{\varphi}(1)\Big) & \phi_2\Big(\boldsymbol{\varphi}(1)\Big) & \cdots & \phi_m\Big(\boldsymbol{\varphi}(1)\Big) \\ \phi_1\Big(\boldsymbol{\varphi}(2)\Big) & \phi_2\Big(\boldsymbol{\varphi}(2)\Big) & \cdots & \phi_m\Big(\boldsymbol{\varphi}(2)\Big) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1\Big(\boldsymbol{\varphi}(N)\Big) & \phi_2\Big(\boldsymbol{\varphi}(N)\Big) & \cdots & \phi_m\Big(\boldsymbol{\varphi}(N)\Big) \end{bmatrix}$$

In general, the error vector **e** can be treated as an independent identically distributed zero mean noise sequence as long as the matrix **Φ** contains sufficient description of the data set [42].

NARX models can be used to describe a wide range of nonlinear systems. Some of the advantages of these models are *transparency*, so they can be related back to the underlying system, and *parsimony*, requiring just a few hundred data samples to estimate a model, which can be important in many applications where it is unrealistic to perform long experiments [5].

2.4.2 Model Structure Detection and Parameter Estimation

Model structure detection is an important and challenging problem. It has been extensively studied and there is a considerable amount of information in the literature [43]. Model structure detection consists of selecting the model order together with the candidate model terms that contribute to the system output while keeping an efficient system description [5, 44–46]. In general, most of the candidate model terms are redundant or spurious; therefore their contribution to the system output is negligible [47,48]. Furthermore, a model that includes a large number of terms tends to generalise poorly on unseen data [44,49]. Among the advantages for performing a careful model structure detection are the improvement of forecasting or classification accuracy, reduction in time and storage cost, and better understanding of the studied process [48]. Because of this, different methods have been developed that search and select the significant model terms that play a major role in the identification process. Some of these methods include hypothesis testing of differences between means via the t-test, stepwise regression, Korenberg's orthogonal structure detection routine [41, 44], clustering [49, 50], the Least Absolute Shrinkage and Selection Operator (LASSO) [38], elastic nets [51, 52], Least Angle Regression (LARS) [53], Principal Component Regression (PCR) [54], genetic programming [55,56], Bayesian approaches [57,58], Bayesian networks [28], and the Orthogonal Forward Regression and Error Reduction Ratio approach [46]. Once the structure has been identified, the model parameters can then be estimated and the significance of each model term can be analysed [57].

Orthogonal Forward Regression algorithm

In general, model structure selection and parameter estimation are performed together. One of the most popular algorithms for this is the Orthogonal Forward Regression (OFR) algorithm [5, 12, 42, 59].

The OFR algorithm was developed in the late 1980s by Billings, *et.al.* [5]. It is a greedy algorithm [53,60] that belongs to the class of recursive-partitioning procedures [27]. The algorithm ranks a set of candidate terms based on their contribution to the output data, and identifies and fits a deterministic parsimonious NARX model that can be expressed in a generalised linear regression form. Algorithm 2.1 describes the OFR algorithm [42,61,62].

The original OFR algorithm uses the Error Reduction Ratio (ERR) index as

Algorithm 2.1 Orthogonal Forward Regression

```
Input: Dictionary $D = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\}$, output signal $\mathbf{y}$, specified threshold $\eta$
Output: NARX model with significant terms selected from $D$ and corresponding parameters $\boldsymbol{\theta}$ estimated
for all $\boldsymbol{\phi}_i$ in $D$ do
    Define $\mathbf{w}_i = \phi_i/\|\phi_i\|_2$
    Compute $ERR^{(i)}(\mathbf{w}_i, \mathbf{y})$
Find $j = \underset{1 \le i \le M}{max} \{ERR^{(i)}(\mathbf{w}_i, \mathbf{y})\}$
Define $\mathbf{q}_1 = \mathbf{w}_j$
Define $a_{11} = \|\phi_j\|_2$
Define $g_1 = \mathbf{q}_1^T \mathbf{y}$
Define $err[1] = ERR^{(j)}$
Define $\mathbf{y}_{new}^{(1)} = \mathbf{y} - g_1 \mathbf{q}_1$
Remove $\boldsymbol{\phi}_j$ from $D$
Define $s = 1$
while $ESR = 1 - \sum_{k=1}^{s} err(k) \ge \eta$ do
    Define $s = s + 1$
    for all $\boldsymbol{\phi}_i$ in $D$ do
        Orthonormalize $\boldsymbol{\phi}_i$ with respect to $[\mathbf{q}_1, \ldots, \mathbf{q}_{s-1}]$ to obtain $\mathbf{w}_i$
        if $\mathbf{w}_i^T \mathbf{w}_i < 10^{-10}$ then
            Remove $\boldsymbol{\phi}_i$ from $D$
            Go to next iteration
        Compute $ERR^{(i)}\left(\mathbf{w}_i, \mathbf{y}_{new}^{(s-1)}\right)$
    Find $j = \underset{1 \le i \le M-s+1}{max} \{ERR^{(i)}(\mathbf{w}_i, \mathbf{y})\}$
    Define $\mathbf{q}_s = \mathbf{w}_j$
    Define $a_{rs} = \mathbf{q}_r^T \boldsymbol{\phi}_j$, $\forall r = 1, 2, \ldots, s-1$
    Define $a_{ss} = \left\| \boldsymbol{\phi}_j - \sum_{r=1}^{s-1} a_{rs} \mathbf{q}_r \right\|_2$
    Define $g_s = \mathbf{q}_s^T \mathbf{y}_{new}^{(s-1)}$
    Define $err[s] = ERR^{(j)}$
    Define $\mathbf{y}_{new}^{(s)} = \mathbf{y}_{new}^{(s-1)} - g_s \mathbf{q}_s$
    Remove $\boldsymbol{\phi}_j$ from $D$
Once the while loop stops and $m$ model terms have been selected, then solve $\mathbf{A}_{m \times m} \boldsymbol{\theta}_{m \times 1} = \mathbf{g}_{m \times 1}$
Return matrix of terms selected $\boldsymbol{\Phi} = [\ \boldsymbol{\phi}_1 \ \ \boldsymbol{\phi}_2 \ \ \ldots \ \ \boldsymbol{\phi}_m\ ]$ and vector of coefficients $\boldsymbol{\theta} = [\ \theta_1 \ \ \theta_2 \ \ \ldots \ \ \theta_m\ ]^T$
```

dependency metric [5]. The ERR of a term represents the percentage reduction in the total mean square error that is obtained if such term is included in the final model [63], and it is defined as the non-centralised squared correlation coefficient (or non-centralised Pearson product-moment correlation coefficient) $C(\mathbf{x},\mathbf{y})$ between two associated vectors $\mathbf{x}$ and $\mathbf{y}$ [64]

$$C(\mathbf{x},\mathbf{y}) = \frac{\left(\mathbf{x}^T\mathbf{y}\right)^2}{\left(\mathbf{x}^T\mathbf{x}\right)\left(\mathbf{y}^T\mathbf{y}\right)} \tag{2.7}$$

A comprehensive explanation of the meaning of ERR may be found in [5,46,65]. Also notice that Algorithm 2.1 requires a threshold η in the Error-to-Signal Ratio (ESR). This is defined as $ESR = 1 - \sum_{k=1}^{s} err(k)$, where $\sum_{k=1}^{s} err(k)$ corresponds to the sum of the ERRs of the model terms selected by the algorithm as it executes. The threshold η is usually set to a small number ($\eta \leq 0.01$) [5].

Mathematically, the OFR algorithm minimises the sum of squared errors defined as

$$\mathcal{L} = \mathbf{e}^T\mathbf{e} = \left(\mathbf{y} - \mathbf{W}\mathbf{g}\right)^T\left(\mathbf{y} - \mathbf{W}\mathbf{g}\right) = \|\mathbf{y} - \mathbf{W}\mathbf{g}\|_{\mathbf{2}}^{\mathbf{2}} \tag{2.8}$$

In subsequent discussions, if the OFR algorithm is used together with the ERR metric, it will be referred as OFR-ERR.

Dependency metrics

The ERR index only detects linear dependencies; therefore new metrics have been implemented recently to identify nonlinear dependencies [42, 63, 64]. One of these new metrics is entropy, which is a measure of the average information that is contained within the probability distribution function of a random variable, and it is defined as

$$H(\mathbf{x}) = -\sum_{x \epsilon \mathcal{X}} p(x) \ln p(x) \tag{2.9}$$

In [63], the authors replaced the ERR for the Shannon's Entropy Power Reduction Ratio (EPRR) that is able to deal with both Gaussian and non-Gaussian signals. Nevertheless, the main drawback of the analysis is the assumption that the variables involved are mutually independent or uncorrelated with a jointly Gaussian distribution.

Another metric extensively used is mutual information. Mutual information $I(\mathbf{x}, \mathbf{y})$ provides a measure of the amount of information that two variables share with each other [64]. It is defined as

$$I(\mathbf{x}, \mathbf{y}) = \sum_{x \epsilon \mathcal{X}} \sum_{y \epsilon \mathcal{Y}} p(x, y) \ln \left(\frac{p(x, y)}{p(x) \, p(y)} \right) \tag{2.10}$$

Mutual information has been extensively used because it captures both linear and nonlinear correlations, and has no assumption on the distribution of the data [42, 48, 64, 66, 67]. Although most of the research is promising, the mutual information is hard to interpret and its computation requires more computing power.

Recently, Reshef, *et.al.* [68] developed another metric, the Maximal Information Coefficient (MIC), which extends the notion of Pearson's correlation coefficient to nonlinear associations between pairs of variables [69]. MIC is based on concepts from Information Theory, an area founded by Claude Shannon [70]. This coefficient gives rise to the Maximal Information-based Nonparametric Exploration (MINE) statistics which focus on identification and characterisation of nonlinear associations [68]. Nevertheless, this new metric has been part of recent scientific debate leading to the conclusion that the MIC has serious power deficiencies according to [71, 72].

Regularisation, Ridge Regression and LASSO

In general, model structure detection and parameter estimation can be improved via regularisation [52]. In fact, different regularisation techniques have been incorporated in the original OFR algorithm with the purpose of reducing the variance of parameter estimates at the cost of introducing a parameter bias [45]. For example, in [45] the authors introduced a new composite cost function that minimizes model prediction error to improve the model approximation ability, while penalising the parameters covariance to ensure a good model adequacy. Such incorporation can detect a parsimonious model in an automatic manner.

Other well-known regularisation techniques are the ridge regression and LASSO [54]. The ridge regression uses the same minimisation strategy given by Eq. (2.8), but adds a penalty to the magnitude of the parameter estimates via the ℓ_2 norm, i.e.

$$\mathcal{L}_{Ridge} = \|\mathbf{y} - \mathbf{W}\mathbf{g}\|_2^2 + \lambda_2 \|\mathbf{g}\|_2^2 \tag{2.11}$$

The LASSO is a form of regularisation that instead of using only the traditional ℓ_2 penalty, minimises Eq. (2.8) together with the ℓ_1 penalty, i.e.

$$\mathcal{L}_{LASSO} = \|\mathbf{y} - \mathbf{W}\mathbf{g}\|_2^2 + \lambda_1 \|\mathbf{g}\|_1 \tag{2.12}$$

The ℓ_1 penalty is interesting since its inclusion in regression problems produces parameter estimates that are exactly equal to zero, resulting in a parsimonious model [38, 51]. In [38], the authors applied the LASSO for structure detection of polynomial NARMAX models in the presence of additive output noise. In such a work, the LASSO produced interesting results, however the resulting model was not carefully validated. The authors applied a percent fit as an indicator of the model goodness, but did not use a proper validation test. Some authors have concluded that LASSO is not able to select a group of correlated terms [52], and it is not an effective variable selection method when the number of model terms is bigger than the number of observations [51].

Elastic Nets

As mentioned before, the advantage of LASSO is that it produces a parsimonious model by setting some parameter estimates to zero. Unfortunately, the LASSO cannot select a group of correlated terms and does not work properly when the number of model terms outnumbers the number of observations [51]. Similarly, the advantage of the ℓ_2 penalty is that it improves model generalisation, however it cannot be used for model selection by itself [52].

Elastic nets were developed by Zou & Hastie [51] as a new regularisation and variable selection method. They are a recently proposed concept that minimises the traditional ℓ_2 penalty together with the ℓ_1 penalty [51, 52], i.e.

$$\mathcal{L}_{EN} = \|\mathbf{y} - \mathbf{W}\mathbf{g}\|_2^2 + \lambda_2 \|\mathbf{g}\|_2^2 + \lambda_1 \|\mathbf{g}\|_1 \tag{2.13}$$

In [52], the authors applied a two-level model identification method. At the lower level, the OFR algorithm is combined with the elastic nets to perform both model selection and parameter estimation. At the upper level, particle swarm optimisation is applied to minimize the leave-one-out mean square error in order to select the best regularisation parameters. A fully automated procedure is achieved. Still, the process is computationally expensive because instead of defining a single threshold for the ESR, two parameters λ_1 and λ_2 need to be estimated in order for the algorithm to work. Additionally, every particle needs to perform the OFR algorithm with elastic nets, and then all of them combine their results to update the regularisation parameters at every iteration.

LARS model selection

A new model selection algorithm known as Least Angle Regression (LARS) was proposed by Efron, *et.al.* [53]. The new algorithm is less greedy than the traditional OFR method. Furthermore, the LARS algorithm is easily modifiable to obtain two model-building algorithms: the LASSO and the Forward Stagewise Linear Regression.

The LARS algorithm assumes that the regressors are linearly independent. The algorithm starts similarly to the OFR, it selects the regressor most correlated to the output. Then, it takes the largest step possible in the direction of this regressor until another regressor has as much correlation with the current residual. At this point, instead of moving along this new regressor, LARS proceeds in a direction equiangular between the two regressors until a third regressor is correlated with the residual. LARS then proceeds equiangularly between the three found regressors, until a fourth variable enters, and so forth [53].

Principal Component Regression

The main idea behind Principal Component Regression (PCR) consists in preprocessing the data by means of Principal Component Analysis (PCA) before performing regression [17]. PCA finds a low-dimensional representation of the data set that contains as much variation as the original one. This new representation contains new features, known as principal components (PCs), that are created using the original regressors (a property known as feature extraction). Mathematically, the jth PC can be written as:

$$PC_j = a_{j1}\boldsymbol{\phi}_1 + a_{j2}\boldsymbol{\phi}_2 + \ldots + a_{jm}\boldsymbol{\phi}_m \tag{2.14}$$

The coefficients $a_{j1}, a_{j2}, \ldots, a_{jm}$ are known as weights and can be used to determine which regressors are more important to each PC [17]. The reader is referred to [54] for an in-depth discussion of PCA. One disadvantage of this technique is that it is an unsupervised procedure, therefore, during the creation of the PCs, the output variable is not taken into account. Alternatives to this issue are mentioned in [17]. Furthermore, given that the PCs are a linear combination of the original regressors, it is harder to provide a meaningful relationship between them and the output variable.

Markov Blanket

The Markov Blanket is a concept proposed by Koller and Sahami in [66], where they developed a Markov blanket filtering to perform a backward elimination model-building procedure.

In [48], the authors developed a forward feature selection method based on approximate Markov blanket. The algorithm employs the mutual information as the selection criteria and uses the Markov blanket as a redundant criterion. The developed algorithm does not need to predefine the number of selected features. It can identify relevant features while removing redundant ones from the set of candidate terms. This makes it efficient for building compact models. Nevertheless, the parameters of the Markov blanket are difficult to define beforehand, like the size of a Markov blanket for a specific problem. The algorithm may end up selecting some attributes that are not relevant and since it is a forward selection algorithm, it is not possible to go back and remove the irrelevant parameters.

Genetic Programming

Genetic programming is a type of evolutionary algorithm that allows the search of both model structure and parameter estimates simultaneously [3]. It is a popular optimisation approach where candidate solutions are evaluated, selected, crossed over and mutated in order to provide a solution to a problem. It is considered a grey modelling approach where a physical interpretable equation is obtained [55]. In [73], a MATLAB Toolbox was developed that implements a Genetic Programming Orthogonal Least Squares algorithm for performing data-based identification of static and dynamic models. The algorithm is fast and efficient in finding a model structure for nonlinear processes. However, the algorithm is stochastic in nature, which requires running the algorithm several times to properly assess the results.

Wavelet Networks

Wavelet Networks (WNs) are a new class of networks that make use of wavelet theory to approximate a signal [74, 75]. In [76], a practical guide was developed for the implementation of these kinds of networks. This guide covers several issues like how to define the structure of a WN, training methods and initialisation algorithms, variable significance and variable selection algorithms, model selection methods and methods to construct confidence and prediction intervals. Also, in [77] and its extensions [78, 79], a new class of WNs are introduced for nonlinear system identification.

The new WNs approximate the model structure for high-dimensional systems by a superimposition of a set of functions with fewer variables based on the ANOVA expansion. Each of these functions is decomposed in a truncated wavelet, which produces a linear-in-the-parameters problem that can be solved using least-squares type methods. These WNs are not a multiresolution decomposition since a scaling function is not involved.

2.4.3 Model Validation

A fundamental part of system identification is model validation. It consists of testing the identified model to check that the parameters estimated are unbiased and that the final model is an adequate representation of the recorded data set [5, 60, 78].

In [80], Billings and Tao developed a set of statistical correlation tests that can be used for nonlinear input-output model testing and validation:

$$\begin{cases} \phi_{\xi\xi}(\tau) = \delta(\tau) & \forall\tau \\ \phi_{u\xi}(\tau) = 0 & \forall\tau \\ \phi_{\xi(\xi u)}(\tau) = 0 & \tau \geq 0 \\ \phi_{(u^2)'\xi}(\tau) = 0 & \forall\tau \\ \phi_{(u^2)'\xi^2}(\tau) = 0 & \forall\tau \end{cases} \tag{2.15}$$

where $\xi(k) = \xi_k$ is the prediction error sequence, $u(k) = u_k$ is the input sequence, $(u^2)'_k = u_k^2 - \overline{u^2}$, $(\xi u)_k = \xi_{k+1} u_{k+1}$, and the cross-correlation function $\phi_{xy}(\tau)$ between two signals x and y is defined as

$$\phi_{xy}(\tau) = \frac{\sum_{k=1}^{N-\tau} [x_k - \bar{x}] [y_{k+\tau} - \bar{y}]}{\sqrt{\sum_{k=1}^{N} [x_k - \bar{x}]^2} \sqrt{\sum_{k=1}^{N} [y_k - \bar{y}]^2}} \tag{2.16}$$

In Eq. (2.15), the first two tests are used in linear system identification. The remaining three tests involve cross-correlation tests between the input and residuals by which all possible omitted nonlinear terms can be detected [60, 79]. If the identified model represents the system adequately, then the residuals should not be predictable from all linear and nonlinear combinations of past inputs and outputs [77].

Sometimes the input signal is unavailable, unmeasured, or unknown, especially when working with time series modelling. For such cases, Billings and Tao [80] developed a set of tests that are effective for time series model validation:

$$\begin{cases} \phi_{\xi'\xi'}(\tau) = \delta(\tau) & \forall\tau \\ \phi_{\xi'(\xi^2)'}(\tau) = 0 & \forall\tau \\ \phi_{(\xi^2)'(\xi^2)'}(\tau) = \delta(\tau) & \forall\tau \end{cases} \tag{2.17}$$

where $\xi'_k = \xi_k - \overline{\xi}$ and $(\xi^2)'_k = \xi^2_k - \overline{\xi^2}$.

If the tests are not satisfied, then it is suggested to reduce the threshold η and/or include more complex model terms within the set of candidate terms. Then new models should be trained until the validity tests are satisfied [77].

It is important to take into account that the correlation tests alone are not adequate to detect discrepancies between the observed dynamical behaviour and the model dynamics [2, 81]. To overcome this, other approaches have been developed. The most popular one is to divide the data in two sets: a training set and a test set. The first one is used for fitting the model. Once the model is trained, it is used to obtain predictions on the test set. These predictions are compared with the true values in order to assess the model's performance. The size of the test set depends on the total number of observations and how far ahead the predictions need to be forecasted, but it is a common practice to use 20% of the total sample [82].

For the particular case of NARX models, there are two types of predicted outputs: one-step ahead (OSA) output and model predicted output (MPO). The former is given by

$$\hat{y}(k) = f\Big(y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)\Big) \tag{2.18}$$

where the most up-to-date values of past outputs and inputs are used to estimate the following output. However, because the NARX model in Eq. (2.1) depends on past output values, a more reliable way to check the validity of the model is through the MPO, which uses past predicted outputs to estimate future ones,

$$\hat{y}\left(k\right) = f\Big(\hat{y}\left(k-1\right), \ldots, \hat{y}\left(k-n_y\right), u\left(k-1\right), \ldots, u\left(k-n_u\right)\Big) \tag{2.19}$$

The MPO can provide details about the stability and predictability range of the model. In [83], the authors developed a lower bound error for the MPO of polynomial NARMAX models, which can be used to detect when a model's simulation is not reliable and needs to be rejected.

In the literature, some authors have adapted the original OFR algorithm to optimise directly the MPO in order to obtain a better long-term prediction [84]. However, these modified versions tend to be computationally expensive during the feature selection step, and a much better alternative is to use the iterative OFR [85] or ultra OFR [86] approaches.

Furthermore, in many real applications, multiple step-ahead predictions are of interest. For an autonomous system (e.g. a time series process without external input), the system output value at the current time instant k, i.e. $y\left(k\right)$, may be predicted using previous observations at time instants $k-h$, $k-h-1$, etc. Therefore, the predicted value $\hat{y}\left(k\right)$ is called the h-step ahead prediction. For an input-output system, the h-step ahead prediction $\hat{y}\left(k\right)$ is often estimated using previous output measurements $y\left(k-h\right)$, $y\left(k-h-1\right)$, ... , and previous input values $u\left(k-1\right)$, $u\left(k-2\right)$, ... , etc. So, for an input-output system model, the h-step ahead prediction is defined with respect to the system output; it is actually still one-step ahead prediction with respect to the system input.

2.5 Environmental Systems Analysis

Throughout the world, there are several environmental concerns that require monitoring and immediate attention, e.g. deforestation [23,87], wildfires [20], wastewater treatment plants [24,88], Atlantic Ocean's major current circulation [25], ocean acidification [89], geomagnetic disturbances in near-Earth space [26], agriculture [90–92], pollution [93, 94], climate change [95–97], among many others.

Recently, several works have focused on extraction of knowledge contained within databases that come from the monitoring of dynamical environmental processes [3, 14, 21, 28, 98–104]. In [19], Gibert, *et.al.* built an Intelligent Environmental Decision Support System (IEDSS) called GESCONDA. This new software tool programmed in Java, focuses on intelligent data analysis and implicit knowledge management of environmental databases. GESCONDA has a multi-layer architecture of 4 levels (data filtering, recommendation and meta-knowledge management, knowledge discovery, and knowledge management) that allows the interaction between the user and the environmental system. This multi-layer architecture eases the knowledge extraction process and permits the pre- and post-processing of the data, as well as the validation of the models produced by GESCONDA. Particularly, the knowledge discovery level handles a great variety of data mining and machine learning techniques like clustering, decision trees, rule induction, support vector machines, dynamical analysis and statistical modelling. This range of techniques allows the analysis of both quantitative and qualitative variables. However, despite all the utilities that this software provides, it seems that its development was halted because the project's website does not allow the download of the software (checked on June 2014).

Mas, *et.al.* [23] implemented a simple model to predict the spatial distribution of tropical deforestation using artificial neural networks. Although the main agents that produce deforestation are known, it is difficult to determine their relative contribution to this problem, and there is not a clear understanding of the large complexity of interactions between human and environmental factors. Such complexity motivated the authors to implement a neural network approach, which has been successfully tested in other environmental problems. The authors described the several processing steps that the data had to go through before/after using the trained neural network. The results obtained are interesting and the neural network was able to obtain a correct classification percentage of 68.6%. Also, the authors clearly mentioned the limitations of their work and are aware that their model is a black box, which is not able to explain the factors that produce the deforestation process, and long-term predictions are not reliable. They suggested that it may be impossible to develop models of deforestation processes with high power of prediction because of the high complexity involved. Still the analysis of these systems is important because it helps to create policies that help to control the negative ecological and social effects of deforestation.

In [20], the occurrence of wildfires was studied. In particular, the authors developed a data mining approach to predict burned area, which can be useful for fire fighting resource planning. A special emphasis was put on the use of real-time and non-costly meteorological data obtained from local sensors, compared with other alternatives like satellites or infrared scanners that are not suitable for the task. However, as suggested by the authors, their approach was not able to predict large fires accurately, and further research is required to determine if direct weather conditions outperform historical records.

One topic that has become important is the design, management and control of wastewater treatment plants for water resource planning [24]. Because of the high complexity of these plants, there are several factors that affect the real-time control of the system. Furthermore, the recorded data from the plants can be very noisy, imprecise and possess several missing values. A variety of techniques have been tested to obtain a prediction model that can help to control efficiently these systems. Among these techniques are fuzzy logic, genetic algorithms, artificial neural networks, probabilistic reasoning, and others. In [24], the authors built a fuzzy heterogeneous time-delay neural network to characterise a wastewater treatment plant located in Catalonia. The training of this network had a low cost compared with recurrent neural networks, and the model obtained seems accurate enough despite the fact that 78.7% of the data was missing.

One popular modelling technique are Bayesian networks. These are a probabilistic graphical modelling approach that has acquired considerable relevance during the last years [28, 105]. The structure of a Bayesian network possesses information about the (ir)relevance of model terms between each other. Once the structure is defined, conditional probabilities encode how strong the relationships are among the model terms. These properties make Bayesian networks suitable for modelling complex systems where uncertainty and missing values cannot be neglected. However, a lot of data is required to build the structure of the network and to estimate the corresponding probabilities. Also, as the size of the network increases, its computation may become intractable. Another limitation is that Bayesian networks were designed to work with discrete variables, and only recently, new methodologies have appeared that deal with continuous or hybrid variables without the need to discretise them [106, 107]. In [28], the authors made a summary of the papers published in the areas of the ISI Web of Knowledge related to Environmental Sciences (with an emphasis in the usage of Bayesian networks) from January 1990 to December 2010. It is surprising that, from the 1375 documents retrieved, only 4.7% focused on environmental issues. From these, 71.1% aimed to perform some inference from the data, 52.6% worked with discrete variables, and 37.7% did not validate the final Bayesian network. The authors concluded that Bayesian networks are still largely unexploited for Environmental Sciences.

2.6 Summary

This chapter gives an overview of the main concepts that are used in this thesis. It describes how data can be used to identify patterns that help to build models that approximate the observed data and generalise from it to unobserved data. One important application of this is system identification, which can be considered a type of supervised learning. Here a mathematical model of a system is identified and fitted based on experimental data that record the system inputs and outputs behaviour. The main steps in system identification involve model structure detection, parameter estimation and model validation. One of the most popular techniques for this task is the NARX methodology, which uses the OFR algorithm. Recently, environmental systems have brought the attention of scientists and researchers given the huge amount of data available. Such systems are highly nonlinear, interact at different spatial and temporal scales, and evolve over time so their high complexities limit the formulation of first-principles models. The next chapter introduces the computational tools used to analyse environmental systems using the NARX methodology.

Chapter 3

R for Data Analysis and System Identification

3.1 Introduction

With the increasing use of data for system identification, new tools have been required to ease the data analysis task. This chapter provides a general overview of the R programming language focusing on two aspects: data analysis and system identification problems. For the first aspect, a comprehensive description of the R packages that allow to handle data in an efficient and easy way is provided. For the second one, given the lack of sophisticated software to build NARX models, the first R package is developed for this purpose. It provides a set of features for effectively performing model selection, parameter estimation, model validation, model visualisation and model evaluation.

Note that this chapter is not an introduction of how to program with R. For detailed discussions on R, interested readers are referred to the following resources:

- R documentation [108]
- Datacamp free introduction to R [109]
- Coursera Data Science Specialisation by Johns Hopkins University [110]

- "R for Data Science" by Hadley Wickham [111]

3.2 The R Language

The R language was created by Ross Ihaka and Robert Gentleman in 1995 at the University of Auckland in New Zealand [112]. It is a dialect of the S language, which was developed at AT&T Bell Laboratories by Rick Becker, John Chambers and Allan Wilks [113, 114]. The S language had its roots in statistical analysis functionality and was only available through a commercial package under the product name S-PLUS. This was one of its key limitations. When the R language was developed, it was agreed to make it free software, which undoubtedly contributed to its popularity in the following years. In 2000, R version 1.0.0 was released to the public [114].

3.2.1 Basic Features of R

Today R is one of the most popular programming languages for statistical computing. Some of its key features are [114]:

- R is under constant improvement, and several updates are released during the year.
- R has an active and vibrant user community with several contributions shared through social media or meetings all over the world.
- R is easily extensible through modular packages, most of them created by the R user community.
- R has very sophisticated graphics capabilities which allows high-quality visualisations of high-dimensional data.
- R provides an interactive computing environment with several tools for data analysis, yet it also allows the user to become a software developer that creates new tools for this purpose.
- R is completely free.

3.2.2 Limitations of R

R is not a perfect programming language and it has several limitations. The main ones are related to its origins [114]:

- R has little built-in support for dynamic or 3D graphics.
- R functionality is based on consumer demand and user contributions. If there is not a package that performs what users want, they need to program it on their own.
- Everything in R is an object and all objects must be stored in physical memory.

However, nowadays most of these limitations are becoming less restrictive. Several alternatives have been developed that produce better graphics, and allow working with large data sets without the need to store them in memory. Also, thanks to the vast R user community, there are 11,413 packages available at the Comprehensive R Archive Network (CRAN) as on September, 2017, which may allow the users to find a package that already does a particular task.

3.3 R for Data Analysis

Today, 90% of the world's data have been generated over the last 2 years at rates that have no precedent in the history of mankind [3, 6]. Therefore, data analysis has become a valuable and vital tool to deal with data sets that contain valuable information but it is difficult to translate into relevant knowledge [8, 9]. Figure 3.1 displays the data analysis process, which consists of a series of steps that aim to discover or learn insightful patterns from an observational data set [3, 13, 14]. The steps are the following [111]:

1. Data importing: it consists in taking the data from its original source and loading it into a proper environment for its further analysis. In R, the most popular package for this step is *readr*.

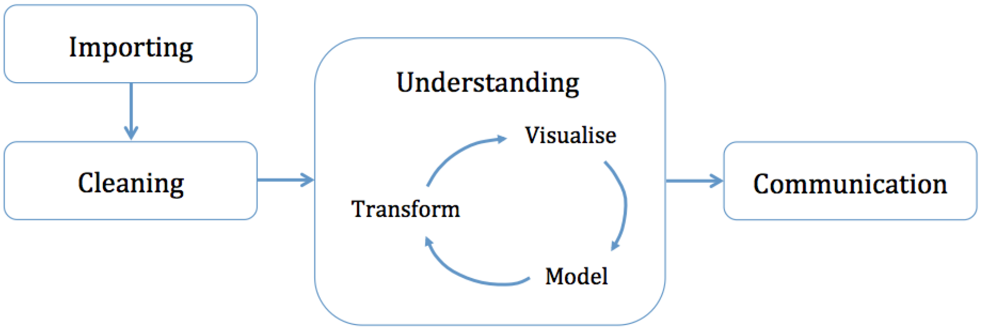

Figure 3.1: Overview of the data analysis process covering the importing, cleaning, understanding and communication steps. Original source: [111].

2. Data cleaning: most of the time, data collected comes in raw form, which is difficult to analyse. Therefore, further analysis can be extremely difficult, if not impossible, if the data is not tidy, i.e. each row is an observation, and each column is a variable. This step can take from 80% to 90% of the whole analysis process, but several packages in R have been developed to reduce the time taken in this step significantly. The most popular packages are *dplyr*, *tidyr* and *purrr*. All of these are included in the *tidyverse* package in R [111].

3. Data understanding: this step is where most of the interesting analysis takes place. It often involves a recursive approach of the following tasks:

    - Transforming: some (or all) variables may require some transformation to facilitate the visualisation and/or modelling tasks. This involves computing summary statistics of the data, imputing missing values, or creating new variables based on the original ones. The *tidyverse* package is quite useful for all of these issues.

    - Visualisation: this is a vital task in the data analysis process. Visualisation can reveal something that is not expected within the data, or hint that the data is not appropriate for the current analysis. A lot of research has been done in data visualisation, and one of the most popular R packages is *ggplot2* [115]. This package is so popular that it has been extended to other programming languages like Python.

- Modelling: this task goes along with data visualisation. Modelling identifies a mathematical or computational tool that should generalise the given data set. There are hundreds of models available and each of them makes its own assumptions about the data in order to work. Depending on the data, some of these assumptions may or may not work, and several iterations may take place before a final model is chosen. Several packages are available in R that cover many different types of models. Independently of these, one of the most popular packages for data preparation before modelling is *caret*.

4. Data communication: once the analysis is done, it is important to share the results with other people. Many analysts agree that this is like storytelling with data, where every step taken to understand the data set is described so that someone else is able to reproduce the same results. R comes with several tools for data communication, including *rmarkdown* and *R notebooks*.

3.4 R for System Identification

One of the difficulties faced during the development of this research, is the lack of software with a standard library for NARX models. Therefore, one of the objectives of this work is to develop the first package in the R language for building NARX models. This package is still under development and may be released in the near future.

3.4.1 The NARX R Package

The NARX R package was designed with the purpose of easing the task of building NARX models while providing a set of features for effectively performing model selection, parameter estimation, model validation, model visualisation and model evaluation. The package offers the following features:

- Implementation of the traditional OFR-ERR algorithm (Algorithm 2.1)

- Implementation of several improvements to the traditional OFR-ERR algorithm
- Functions to perform OSA (Eq. (2.18)) and MPO (Eq. (2.19)) predictions
- A function to perform OSA validation tests as described in [80]
- Functions for static and interactive visualisation
- A function to assess the performance of the trained model. Defining e_k as the error between the kth prediction $\hat{y}_k$ and the kth output value y_k, i.e. $e_k = \hat{y}_k - y_k$, then the three performance (error) metrics considered are:
  - Mean Error $ME = \frac{1}{N}\sum_{k=1}^{N} e_k$
  - Root Mean Squared Error $RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e_k^2}$
  - Mean Absolute Error $MAE = \frac{1}{N}\sum_{k=1}^{N} |e_k|$

3.4.2 Traditional Orthogonal Forward Regression Algorithm

Algorithm 2.1 was the first one implemented within the NARX R package. To describe it, consider the following example taken from [42]:

$$\begin{aligned} y\left(k\right) = &- 0.5y\left(k-2\right) + 0.7y\left(k-1\right)u\left(k-1\right) + 0.6u^2\left(k-2\right) \\ &+ 0.2y^3\left(k-1\right) - 0.7y\left(k-2\right)u^2\left(k-2\right) + e\left(k\right) \end{aligned} \tag{3.1}$$

where the input $u\left(k\right) \sim \mathcal{U}\left(-1,1\right)$, that is $u\left(k\right)$ is uniformly distributed over $[-1,1]$, and the error $e\left(k\right) \sim \mathcal{N}\left(0,0.02^2\right)$. Following [42], the maximum lags for the input and output are chosen to be $n_u = n_y = 4$ and the nonlinear degree is $\ell = 3$. The stop criterion for the OFR-ERR algorithm is when the ESR is less than 0.05, i.e. $\eta = 5\%$. A total of 500 input-output data points are generated. The data is then split into two parts: a training set of 400 points, and a testing set of the remaining

100 points. The results for the OFR-ERR algorithm are shown in Table 3.1 and Figure 3.2. It can be seen that all the model terms selected are correct except for two that are missing.

Term	Parameter		ERR (%)
	True	Estimate	
$u^2(k-2)$	0.6	0.5708	42.34
$y(k-2)$	-0.5	-0.6680	41.58
$y(k-1)\,u(k-1)$	0.7	0.6636	11.95

Table 3.1: Identified NARX model for Eq. (3.1) using the OFR-ERR algorithm. Three model terms were identified with their corresponding estimated parameter values. The true parameter values are included for reference. The ERR for each model term is shown in the last column.

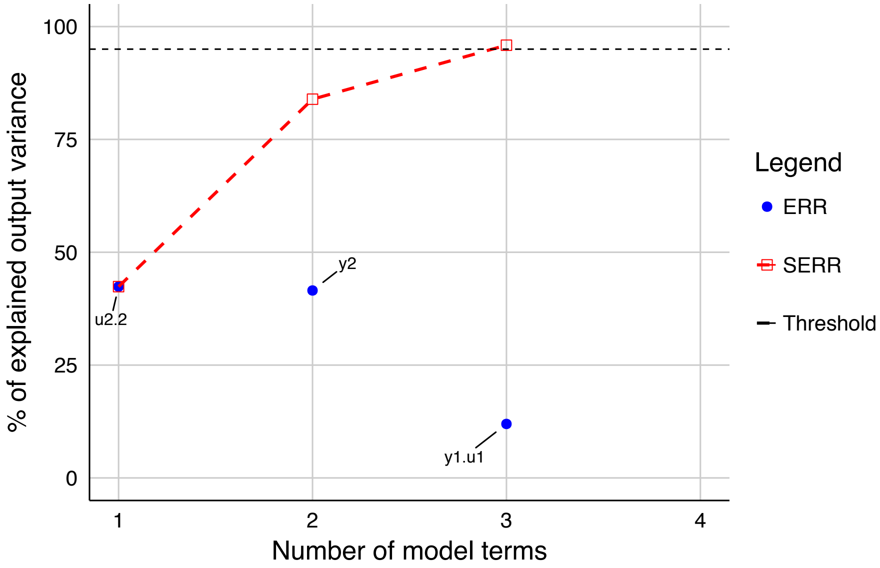

Figure 3.2: Model terms selected for Eq. (3.1) by the OFR-ERR algorithm with their corresponding ERR shown in blue dots, and the updated sum of ERR (SERR) represented in a red dashed line. The total SERR is 95.87%, which satisfies the ESR threshold of 5% shown as a horizontal black dashed line.

The testing set is used for validation purposes. Figure 3.3 shows the results of the validation tests as described by Eq. 2.15 in [80]. Here it is possible to see that the $\phi_{\xi(\xi u)}(\tau)$ test is not satisfied because the correlations have several values well outside the 95% confidence bands, which means that certain nonlinearities are not captured by the model.

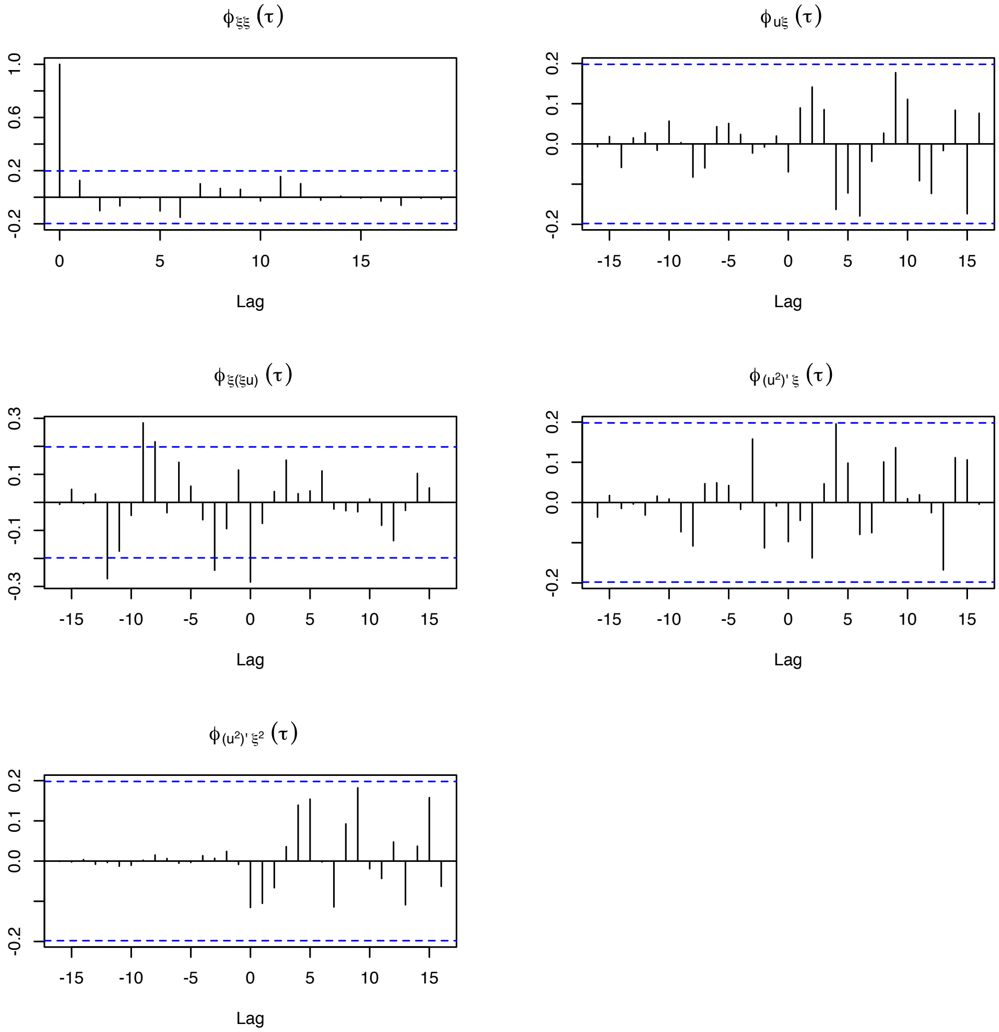

Figure 3.3: Validation tests with 95% confidence limits for the OSA output of the NARX model identified for Eq. (3.1) using the OFR-ERR algorithm. The $\phi_{\xi(\xi u)}(\tau)$ test is not satisfied because the correlations have several values well outside the 95% confidence bands, which means that certain nonlinearities are not captured by the model.

Furthermore, the performance of the model can be evaluated. Figure 3.4 displays the OSA predicted output of the model for the testing set, and Table 3.2 shows the corresponding performance metrics.

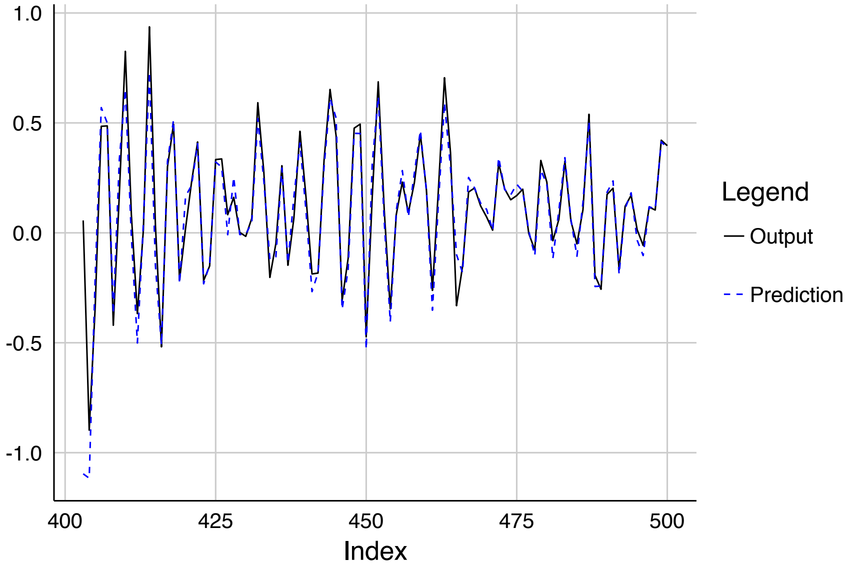

Figure 3.4: OSA output of the NARX model identified for Eq. (3.1) using the OFR-ERR algorithm, where the black solid line indicates the true measurements, and the blue dashed line represents the OSA predicted output.

	ME	**RMSE**	**MAE**
OSA	0.02136	0.1363	0.06143
MPO	0.008514	0.2041	0.1288

Table 3.2: Performance metrics for the OSA predicted output and MPO of the NARX model identified for Eq. (3.1) using the OFR-ERR algorithm. Each of the abbreviations stands for Mean Error (ME), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE), respectively. As expected, the MPO performance metrics are slightly worse than the OSA ones.

However, as stated in section 2.4.3, a more reliable way to check the validity of the model is through the MPO, which uses past predicted outputs to estimate future ones, and can provide details about the stability and predictability range of the model. Figure 3.5 and Table 3.2 show the MPO and performance metrics, respectively. As expected, these performance metrics are slightly worse than the OSA ones. Nevertheless, from Figure 3.5 it can be seen that the model output is stable although it does not capture all the nonlinear dynamics of the system given by Eq. (3.1).

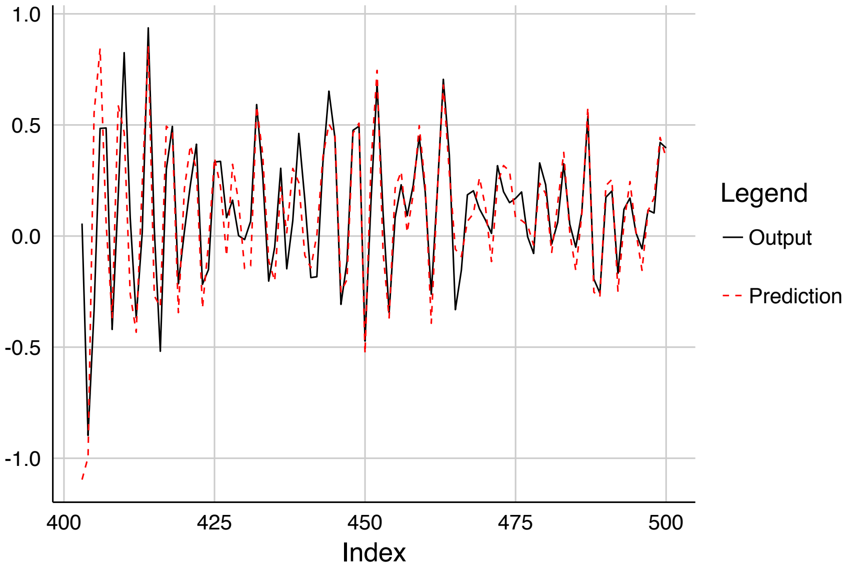

Figure 3.5: MPO of the NARX model identified for Eq. (3.1) using the OFR-ERR algorithm, where the black solid line indicates the true measurements, and the red dashed line represents the MPO.

3.4.3 Improved Orthogonal Forward Regression Algorithm

From the results above, it can be noticed that there are a couple of drawbacks with the OFR-ERR algorithm. As it has been mentioned, the ERR index only detects linear dependencies. This is a problem when there is an interest in detecting not only linear but also nonlinear dependencies. Furthermore, the algorithm strongly depends on the value of the threshold η. If it is too small, the identified model will not capture the dynamics of the system completely. However, if it is too large, the model will overfit and will not be able to generalise well on new observations. Solutions to overcome these issues are discussed in this section.

Nonlinear Dependency Metrics

In recent years, several new metrics have been proposed that are able to identify nonlinear dependencies [42, 63, 64]. The following have been implemented in the NARX R package:

- Mutual Information (MI): it provides a measure of the amount of information that two variables share with each other [64]. It captures both linear and nonlinear correlations, and has no assumption on the distribution of the data

[42, 48, 64, 66, 67]. Model terms with a high mutual information with respect to the output signal are selected. The *infotheo* package implements it in the R language. One disadvantage of this metric is that it is hard to interpret.

- Maximal Information Coefficient (MIC): it extends the notion of Pearson's correlation coefficient to nonlinear associations between pairs of variables [69, 70]. Model terms that maximise the MIC are selected. The *minerva* package implements it in the R language. However, this new metric has been part of recent scientific debate leading to the conclusion that the MIC has serious power deficiencies according to [71, 72].

- Correlation Feature Selection with Symmetric Uncertainty: it finds the best subset of predictors that have strong correlations with the outcome but weak between-predictor correlations [17]. It is defined as $G = \frac{mR_y}{\sqrt{m+m(m-1)\overline{R_x}}}$, where m is the number of model terms, R_y is the correlation between the candidate predictor and the outcome, and $\overline{R_x}$ is the average correlation between the current predictor and the $m-1$ predictors already included in the model [17, 116]. Model terms that maximise the correlation feature selection metric are preferred.

- Predicted Residual Sum of Squares (PRESS): it is defined as $PRESS_m = \frac{1}{N}\sum_{k=1}^{N}\left[y\left(k\right) - \hat{y}_m^{(-k)}\left(k\right)\right]^2$ where $y\left(k\right)$ is the kth data output, and $\hat{y}_m^{(-k)}\left(k\right)$ is the OSA prediction from a model of m model terms, fitted using a data set consisting of $N-1$ data points, which are obtained by leaving the kth data point out [47]. Model terms that minimise the PRESS are chosen.

- Distance Correlation: please refer to Section 4.3 of Chapter 4 for an extensive discussion of this metric.

Selecting the number of model terms

The selection of the appropriate number of model terms has been investigated before in [47]. In this work, this process is improved in order to find a parsimonious model in an efficient way. Instead of choosing a threshold η, it is proposed that the number of model terms is increased sequentially up to a certain upper limit m_{max}, and the best model with m terms ($m \leq m_{max}$) is selected among them. The selection can be based on the following metrics:

- Penalty metrics: these metrics provide a relative quality of a model for a given data set, penalising those models that are too complex. The most popular metrics are the following [82, 117]:
    - Akaike Information Criterion (AIC): it is defined as $AIC = N \log\left(\frac{SSE}{N}\right) + 2\,(m+2)$, where N is the number of observations, SSE is the sum of squared errors $SSE = \sum_{i=1}^{N} e_i^2$, and m is the number of model terms selected. The model with the minimum value of the AIC is often the best model for forecasting.
    - Corrected Akaike Information Criterion (AICc): it is defined as $AICc = AIC + \frac{2(m+2)(m+3)}{N-m-3}$. Similar to the AIC, the model with the minimum AICc is chosen.
    - Bayesian Information Criterion (BIC): it is defined as $BIC = N \log\left(\frac{SSE}{N}\right) + (m+2)\log(N)$. This metric penalises the number of model terms more heavily than the AIC. The model with the minimum BIC is the best one.
- Performance metrics: these metrics evaluate the accuracy of a model on a testing set. The two most popular choices are:
    - Training / Testing splitting: as mentioned in section 2.4.3, this approach consists in dividing the data in two sets: a training set and a test set. The first one is used for model development. Once the model is trained, it is used to obtain predictions on the test set. These predictions are compared with the true values in order to assess the model performance using an error metric, i.e. ME, RMSE, MAE, etc. The process is repeated for each new model term that is added and the final model is chosen using

the number of terms m that minimised the error metric. The size of the test set depends on the total number of observations and how far ahead the predictions need to be forecast, but it is a common practice to use 20% of the total sample [82].

- Cross-Validation (CV): this approach consists in randomly dividing the data set into k groups, or folds, of approximately equal size. Each fold is treated as a testing set, and the training is performed on the remaining $k-1$ folds. An error metric E is computed for each testing set and the k-fold CV estimate is computed by taking the average of these values, i.e. $CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} E_i$ [54]. This procedure is computationally expensive as it needs to be repeated for each model of size m. At the end, the number of terms m that produced the lowest $CV_{(k)}$ is chosen, and the complete data is used to identify the corresponding model. In practice, one typically performs k-fold CV using $k = 5$ or $k = 10$ [54]. It is also common to repeat 3 or 5 times the whole CV process and average the results from all repetitions to choose the best model [17].

Each of the two type of metrics have their own advantages and disadvantages. The penalty metrics are fast to compute but they strongly rely on statistical assumptions on the data. Furthermore, they provide a measure of the statistical validity of the model, but not the accuracy performance on unobserved data. On the other hand, the performance metrics provide a better estimate of the test error, and make fewer assumptions about the true underlying model [54]. However, by splitting the data, the sample size is reduced for both model training and testing. It is also computationally expensive since the process may need to be repeated several times to achieve good estimates of accuracy [117].

Consider again the system described by Eq. (3.1) to facilitate the comparison with the improved version of the OFR-ERR algorithm. It should be expected that such improvements would help us to identify the appropriate number of model terms to use. This time, mutual information is considered as nonlinear dependency metric, together with the BIC metric to determine the appropriate number of terms. The maximum number of terms to look at is $m_{max} = 10$. Figure 3.6 shows that using more than 5 model terms does not improve the quality of the model.

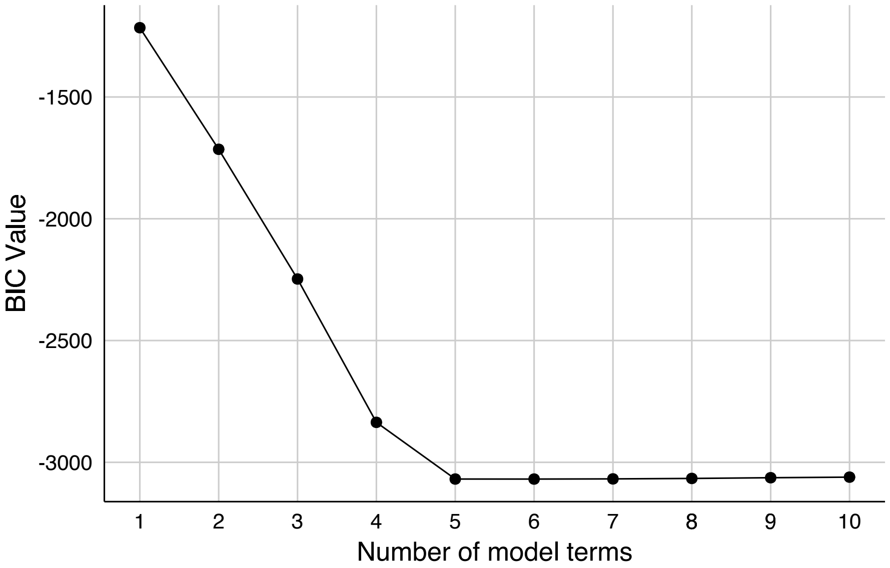

Figure 3.6: BIC plot obtained for Eq. (3.1) using the improved OFR-ERR algorithm. Low BIC values are preferred. Using more than 5 model terms does not improve the quality of the model.

Furthermore, the CV approach can be used instead of the BIC metric. As an example, 5-fold CV is considered with 2 repetitions. Again, the maximum number of terms to look at is $m_{max} = 10$. Figure 3.7 shows that the minimum CV error is obtained when the number of model terms is 6. However, applying the one standard error rule [54], i.e. select the simplest model for which the CV error is within one standard error from the minimum CV error found, then only 5 model terms are enough. Furthermore, it can be seen that using more than 5 model terms does not produce any significant improvement on the performance of the model.

The identified 5-term model is shown in Table 3.3, where it can be seen that all the model terms selected are correct.

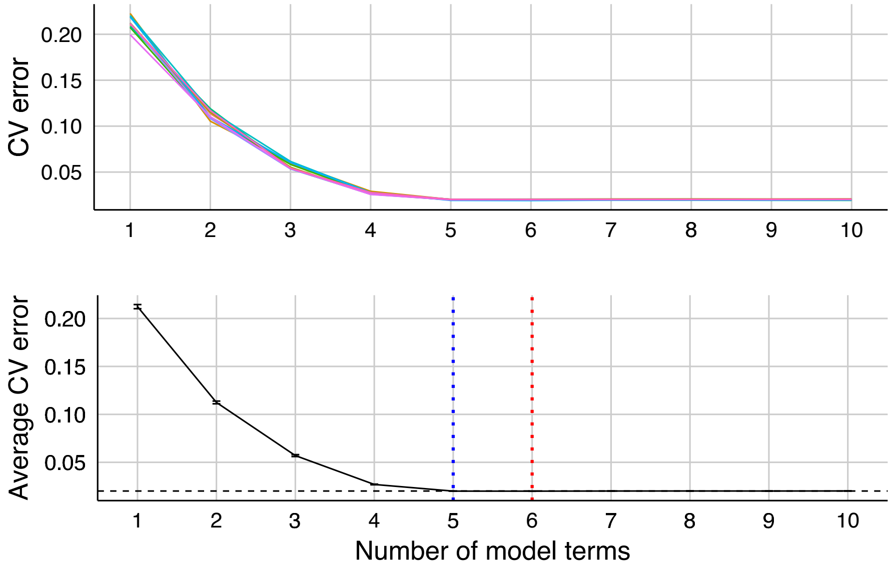

Figure 3.7: Plot obtained for 5-fold CV with 2 repetitions for Eq. (3.1) using the improved OFR-ERR algorithm. TOP: Results for each of the 10 CV errors. BOTTOM: Average of the 10 CV errors shown on top with the corresponding one standard error bars. The vertical red dotted line indicates the number of model terms that achieved the minimum average CV error, the vertical blue dotted line indicates the number of model terms that satisfies the one standard error rule, and the horizontal black dashed line is the threshold for the one standard error rule.

Term	**Parameter**		**MI**
	True	**Estimate**	
$y\,(k-2)$	-0.5	-0.50465	0.3481
$u^2\,(k-2)$	0.6	0.6037	0.5285
$y\,(k-1)\,u\,(k-1)$	0.7	0.6944	0.6012
$y\,(k-2)\,u^2\,(k-2)$	-0.7	-0.6829	0.5074
$y^3\,(k-1)$	0.2	0.1926	0.3068

Table 3.3: Identified NARX model for Eq. (3.1) using the improved OFR-ERR algorithm. Five model terms were identified with their corresponding estimated parameter values. The true parameter values are included for reference. The MI for each model term is shown in the last column.

The testing set is used for validation purposes and Figure 3.8 shows the results of the validation tests as described by Eq. (2.15). Here it is possible to see that all tests have been satisfied because the error autocorrelation has a single peak at lag 0, and all four cross-correlation tests have values within the acceptable margin of the 95% confidence bands, which means that all nonlinearities have been captured by the model.

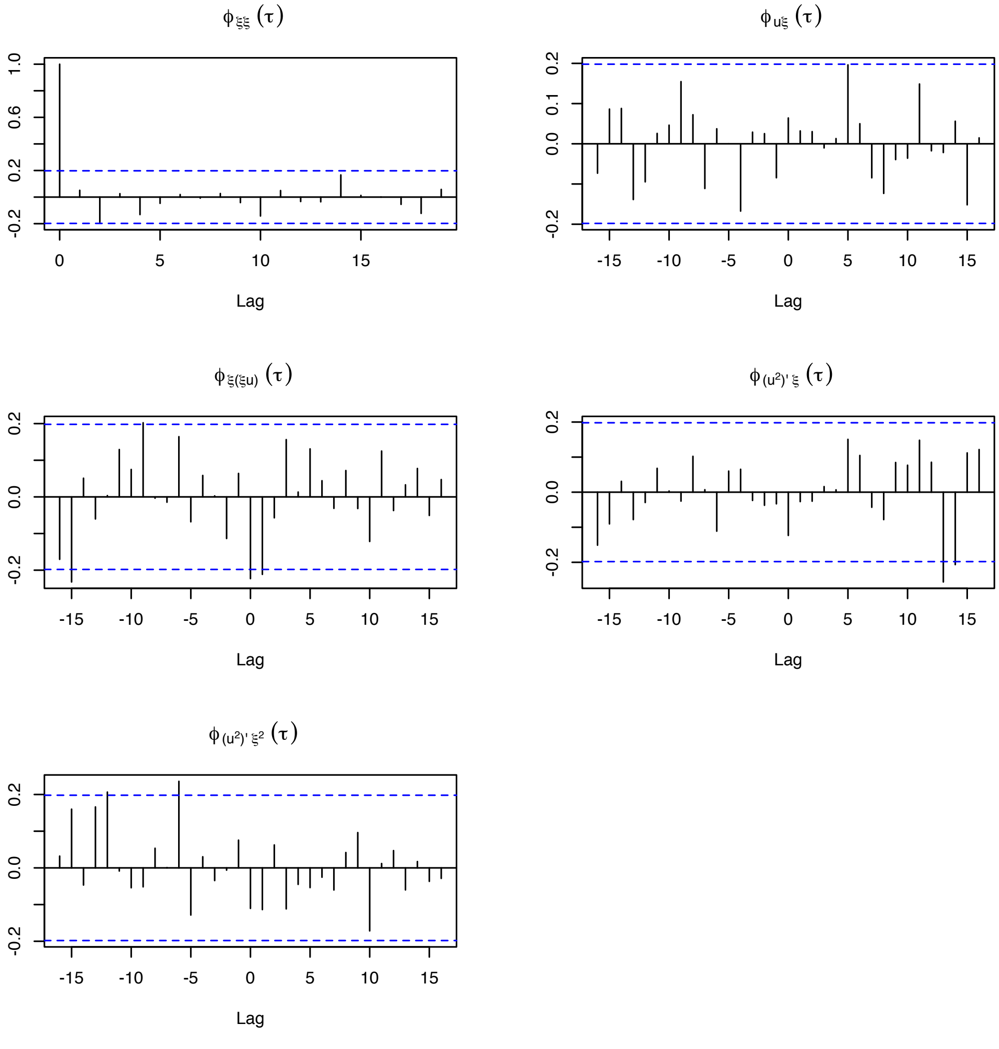

Figure 3.8: Validation tests with 95% confidence limits for the OSA predicted output of the NARX model identified for Eq. (3.1) using the improved OFR-ERR algorithm. All tests have been satisfied because the error autocorrelation has a single peak at lag 0, and all four cross-correlation tests have values within the acceptable margin of the 95% confidence bands, which means that all nonlinearities have been captured by the model.

Figure 3.9 displays the OSA predicted output of the model for the testing set, and Table 3.4 shows the corresponding performance metrics. Comparing these with the OSA results from the first trained model (Table 3.1), it is clear that the new model has a better performance.

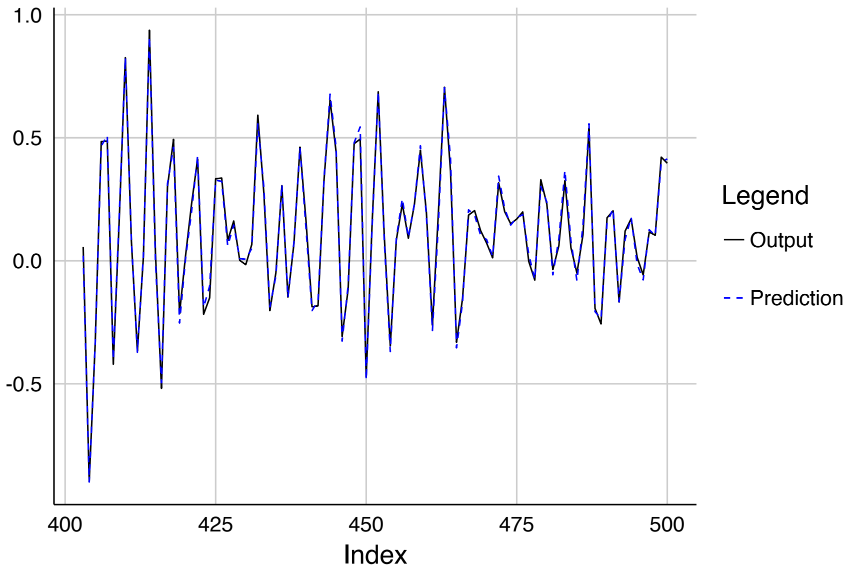

Figure 3.9: OSA predicted output of the NARX model identified for Eq. (3.1) using the improved OFR-ERR algorithm, where the black solid line indicates the true measurements, and the blue dashed line represents the OSA predicted output.

	ME	**RMSE**	**MAE**
OSA	0.000192	0.02164	0.01801
MPO	-0.000199	0.04120	0.03266

Table 3.4: Performance metrics for the OSA predicted output and MPO of the NARX model identified for Eq. (3.1) using the improved OFR-ERR algorithm. Each of the abbreviations stands for Mean Error (ME), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE), respectively. As expected, the MPO performance metrics are slightly worse than the OSA ones.

To finish the comparison, the MPO is also computed for the new model. Figure 3.10 and Table 3.4 show the MPO and performance metrics, respectively. As expected, these results are much better than the ones obtained with the traditional OFR-ERR algorithm.

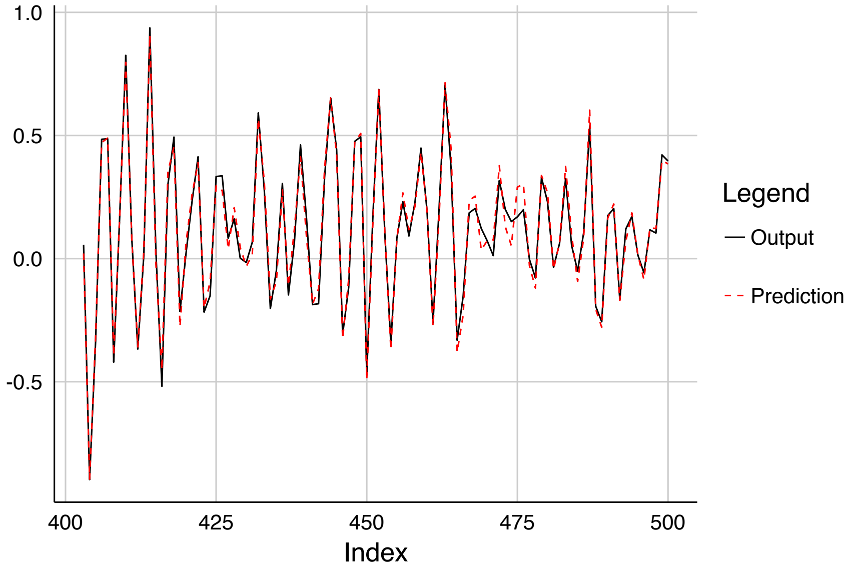

Figure 3.10: MPO of the NARX model identified for Eq. (3.1) using the improved OFR-ERR algorithm, where the black solid line indicates the true measurements, and the red dashed line represents the MPO.

The previous examples clearly show the effectiveness of the NARX R package in providing to the user a set of guiding and informative tools to build NARX models.

3.5 Summary

This chapter briefly discusses the R language and its application to data analysis and system identification. Given the great amount of data that is generated nowadays in every field of study, it is important to use the correct tools to handle them. Data analysis is a process that aims to discover insightful patterns that are buried within a data set. Such process requires a series of steps that include data importing, data cleaning, data understanding and data communication. The R language provides several packages that can be used to perform the whole data analysis process. Given the many advantages of R, it was decided to use it for system identification problems. However, no package is available for this task. Therefore, as part of the objectives of this research, an R package has been developed for building NARX models. Several utilities have been implemented in this new package that help in the detection of nonlinear dependencies within a data set, and the selection of the appropriate number of terms to be used, among other tools for visualisation and forecasting. The package is still under development and may be released in the near future.

Chapter 4

Bagging Forward Orthogonal Regression Algorithm

4.1 Introduction

It has been mentioned that the NARX methodology ranks a set of candidate terms based on their contribution to the output data and identifies parsimonious models that generalise well on new data [5]. The commonly used criterion to measure the dependency between candidate model terms and the desired output is linear correlation however, it can only identify linear dependencies. Therefore, new metrics have been implemented recently to identify nonlinear dependencies. Some of these new metrics are entropy [63], mutual information [42,64,66,67], maximal information coefficient [69], correlation feature selection with symmetric uncertainty [116], among others. Refer to Section 3.4.3 for a description of some of these dependency metrics. In particular, mutual information has been extensively used because it captures both linear and nonlinear correlations, and has no assumption on the distribution of the data [48]. Although most of the research is promising, the mutual information is hard to interpret because its maximum value is not fixed and depends on the entropy of the variables involved.

Another important issue is the need to extend the deterministic notion of the

NARX model to accommodate uncertainties in the parameter estimates, as well as the identified model and the computed predictions. Some authors have worked towards the incorporation of the Bayesian approach within the NARX methodology. An interesting example is the work by Baldacchino, *et.al.,* [58] which developed a computational Bayesian framework for NARMAX models using the Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure, an iterative sampling technique for performing inference in the context of model selection [118]. In [58], Bayesian inference was a key element to estimate not only the parameters but also the model. The results obtained are interesting, however the main drawback is that there are many assumptions in the probability distributions of the parameters involved, and the likelihood and prior distributions were selected carefully to be conjugate priors, an assumption that may not always be accurate. Furthermore, there are several implementation issues that make the reproducibility of the results difficult.

In this chapter, both the use of a *novel metric to detect nonlinearities* within the data set, and the *extension of the deterministic notion* of the NARX model are addressed. For the first case, the distance correlation metric is implemented, which is a measure that belongs to a new class of functions of distances between statistical observations and is able to detect all types of nonlinear or non-monotone dependencies between random vectors with finite first moment, but not necessarily with equal dimension [119, 120]. This is the first time that the distance correlation is introduced and implemented into the OFR algorithm [121]. For the second case, the bagging method is used. Bagging consists of running an algorithm several times on different bootstrap realisations and the results obtained are combined to predict a numerical value via averaging (for regression problems) or via voting (for classification problems). The combination of these two implementations enhances the performance of a NARX model and provides interpretability of nonlinear dependencies together with an insightful uncertainty analysis. The new algorithm is referred as the Bagging Forward Orthogonal Regression using distance Correlation algorithm. For simplicity, the discussion is restricted to polynomial models that can be expressed in a linear-in-the-parameters form.

4.2 The Bootstrap and Bagging Methods

The bootstrap method was developed by Bradley Efron [122]. It is a computer-based approach that computes measures of accuracy to statistical estimates. Bootstrapping consists of randomly sampling R times, with replacement, from a given data set where it is assumed that the observations are independent of each other. Each of the resamples is called a *bootstrap realisation* and has the same length as the original data set. The bootstrap realisations can be treated as unique data sets that produce their own results when used in a specific algorithm, method or technique. Such results contain information that can be used to make inferences from the original data set [123, 124].

The bootstrap method has been previously used for system identification of NARX models. In [41, 44], bootstrapping was used for structure detection where a backward elimination scheme was implemented to find the significant model terms. Such methodology is computationally expensive, as the bootstrap method must be applied every time a model term is eliminated. Furthermore, the methodology may not work when the lag order of the system is large. In [62], the bootstrap was used for parameter estimation of a fixed model. Although the parameter estimation was improved, by fixing the model there is no guarantee that the bootstrapped data came from the true model. The main drawback of these previous works is that the model structure needs to be correct in order for bootstrap to work [62].

In this work, the bootstrap method is applied to time series problems based on [123]. Considering that observations at a given time may depend on previously measured observations, the data set is split into overlapping blocks of fixed length B. The first and last observations appear in fewer blocks than the rest; therefore the data set is wrapped around a circle to make all data points participate equally [124]. Then the blocks are sampled with replacement until a new data set is created with the same length as the original one. This methodology is known as moving blocks

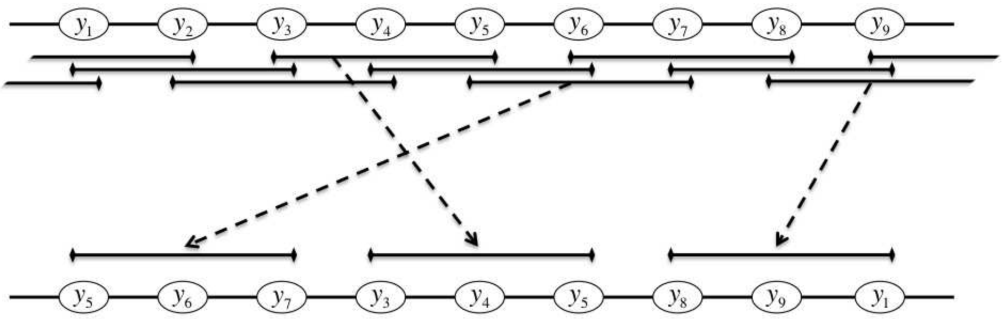

Figure 4.1: Schematic of the moving blocks bootstrap for time series methodology. The upper line corresponds to the original time series. This is split into overlapping blocks of fixed length B, which are sampled with replacement until a new data set is created with the same length as the original one. The lower line corresponds to a bootstrap realisation generated by choosing a block length $B = 3$.

bootstrap for time series [123] and it is illustrated in Figure 4.1. By sampling the blocks, the correlation present in observations less than B units apart is preserved. This methodology is less "model dependent" than the bootstrapping of the residuals approach [123]. It is important to notice that the choice of B is quite important. If it is too small, the correlation within the observations may be lost. If it is too big, there would be no distinction between the original data set and the bootstrap realisations. Although B can be selected empirically by running several simulations and picking the value that produces the best results, effective methods for choosing B are still been investigated as the computation time can vary substantially depending on the size of the data set and the number of lags to consider. In the remaining of this chapter, it is assumed that B is known beforehand.

The bootstrap technique has been extended to a very popular approach nowadays. Assume that a total of R bootstrap realisations have been carried out and each of them has been used in a specific algorithm to duplicate a result of its own. Therefore, R outputs are generated and all of them can be used to predict a numerical value via averaging (for regression problems) or via voting (for classification problems). There is no clear choice for the value of R. In [123], it is suggested that 25 or 200 bootstrap realisations produce decent results, although nowadays it is common to use 1000 or more as a rule of thumb. This procedure is known as bagging (that stands for *b*ootstrap *agg*regat*ing*) and was proposed by Leo Breiman [125].

4.3 Distance Correlation

The distance correlation was developed by Székely, *et.al.* [119]. It is a measure that belongs to a new class of functions of distances between statistical observations [120]. Distance correlation, denoted as $dCor\left(\mathbf{x},\mathbf{y}\right)$, provides a new approach to measure all types of nonlinear or non-monotone dependencies between two random vectors with finite first moment, but not necessarily with equal dimension [119, 120].

The distance correlation requires the computation of the distance covariance. Considering an observed random sample $(\mathbf{x},\mathbf{y}) = \{(x_k, y_k) : k = 1, \ldots, N\}$, the sample distance covariance is calculated as follows [119, 120]:

1. Compute all the pairwise distances between sample observations of the $\mathbf{x}$ sample to get a distance matrix.

2. Similarly, compute the distance matrix for the $\mathbf{y}$ sample.

3. Centralize the entries of the distance matrices. For the $\mathbf{x}$ distance matrix, this can be achieved by using the following formulas:

$$a_{kl} = \|x_k - x_l\|_2 \qquad \bar{a}_{k\cdot} = \tfrac{1}{N}\textstyle\sum_{l=1}^{N} a_{kl} \qquad \bar{a}_{\cdot l} = \tfrac{1}{N}\textstyle\sum_{k=1}^{N} a_{kl}$$

$$\bar{a}_{\cdot\cdot} = \tfrac{1}{N^2}\textstyle\sum_{k,l=1}^{N} a_{kl} \qquad A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$$

4. Repeat for the $\mathbf{y}$ distance matrix, using $b_{kl} = \|y_k - y_l\|_2$ and $B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}$ for $k, l = 1, \ldots, N$.

5. Using the centered distance matrices A_{kl} and B_{kl}, the sample distance covariance $\mathcal{V}_N\left(\mathbf{x},\mathbf{y}\right)$ is the nonnegative number defined as the square root of

$$\mathcal{V}_N^2\left(\mathbf{x},\mathbf{y}\right) = \frac{1}{N^2}\sum_{k,l=1}^{N} A_{kl}B_{kl} \tag{4.1}$$

From Eq. (4.1), the sample distance variance $\mathcal{V}_N\left(\mathbf{X}\right)$ can be defined as the square root of

$$\mathcal{V}_N^2\left(\mathbf{x}\right) = \mathcal{V}_N^2\left(\mathbf{x},\mathbf{x}\right) = \frac{1}{N^2}\sum_{k,l=1}^{N} A_{kl}^2 \tag{4.2}$$

The sample distance correlation $dCor\left(\mathbf{X},\mathbf{Y}\right)$ is defined as the square root of

$$dCor^2\left(\mathbf{x},\mathbf{y}\right) = \begin{cases} \frac{\mathcal{V}_n^2(\mathbf{x},\mathbf{y})}{\sqrt{\mathcal{V}_n^2(\mathbf{x})\mathcal{V}_n^2(\mathbf{y})}} & \mathcal{V}_n^2\left(\mathbf{x}\right)\mathcal{V}_n^2\left(\mathbf{y}\right) > 0 \\ 0 & \mathcal{V}_n^2\left(\mathbf{x}\right)\mathcal{V}_n^2\left(\mathbf{y}\right) = 0 \end{cases} \tag{4.3}$$

The sample distance correlation has the following properties [119, 120]:

i) $0 \leq dCor\left(\mathbf{x},\mathbf{y}\right) \leq 1$

ii) If $dCor\left(\mathbf{x},\mathbf{y}\right) = 0$, then the random vectors $\mathbf{x}$ and $\mathbf{y}$ are independent.

iii) If $dCor\left(\mathbf{x},\mathbf{y}\right) = 1$, then the dimensions of the linear subspaces spanned by $\mathbf{x}$ and $\mathbf{y}$ are almost surely equal. Therefore, there exists a vector $\mathbf{a}$, a nonzero real number b and an orthogonal matrix $\mathbf{C}$ such that $\mathbf{y} = \mathbf{a} + b\mathbf{C}\mathbf{x}$. This means that the random vectors $\mathbf{x}$ and $\mathbf{y}$ are statistically dependent.

The sample distance correlation is analogous to Pearson product-moment correlation coefficient ρ. However, Pearson's coefficient only characterises linear dependency between two variables while distance correlation is a more general measure that characterizes independence of random variables [120].

As a simple comparison, Figure 4.2 displays three distinct noisy data sets. These have been created using a linear ($y = x$), sinusoidal ($y = \sin\left(x + \frac{\pi}{2}\right)$), and circular ($x^2 + y^2 = 1$) relationship with additive white noise. Each of the figures shows the respective values for the Pearson product-moment correlation coefficient, mutual information and distance correlation. The Pearson coefficient is able to detect a linear dependency in the first data set, but finds no such dependency in the other cases, as expected. The mutual information provides a better insight in each of the data sets, but its value is difficult to interpret because the maximum value of the mutual information is not fixed and depends on the entropy of each of the variables involved. Finally, the distance correlation is able to detect dependencies in all cases. Also, the distance correlation is not as strict as the Pearson coefficient, and the fixed range between 0 and 1 for possible values of the distance correlation is an important characteristic that plays a key role in the new algorithm when determining significant terms. If the distance correlation is equal to 0, then it can be assumed that the variables are independent. Similarly, if the distance correlation is equal to 1, then the variables involved are statistically dependent. It is important to mention that one drawback of the distance correlation metric is its computation time, since it can take three times longer to compute it compared with the Pearson coefficient

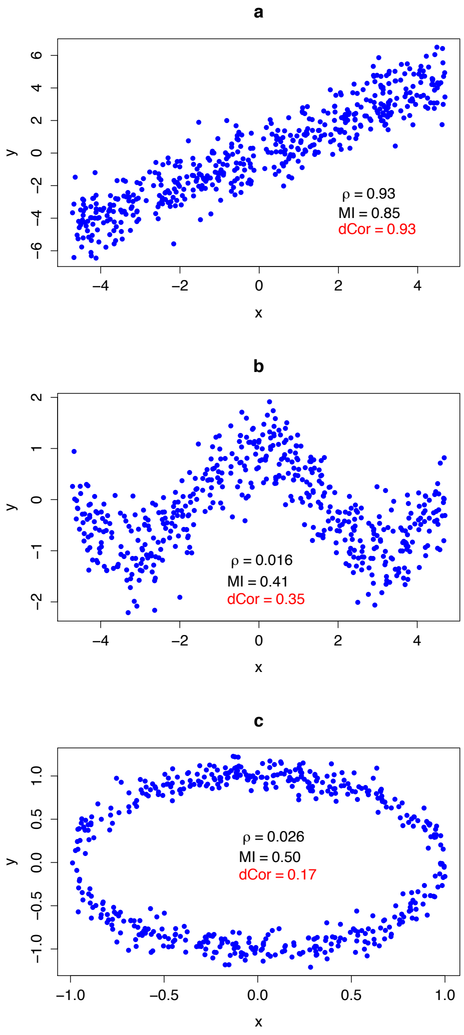

Figure 4.2: Three distinct noisy data sets displaying a a) linear, b) sinusoidal, and c) circular dependency. In each case the Pearson product-moment correlation coefficient (ρ), mutual information (MI) and distance correlation (dCor) are computed. The Pearson coefficient is able to detect a linear dependency in case a), but finds no such dependency in the other cases. The mutual information provides a better insight in all cases, but its value is difficult to interpret. Finally, the distance correlation is able to detect dependencies in all cases.

or the mutual information. This metric has been implemented by its authors in the *energy* package within the R programming language using efficient coding to reduce the computation time issue.

4.4 The BFOR-dCor algorithm

The bagging method and distance correlation are combined with the OFR algorithm to produce the Bagging Forward Orthogonal Regression using distance Correlation (BFOR-dCor) algorithm. This is the first time that the distance correlation metric is introduced and incorporated to the well-known Orthogonal Forward Regression [121]. This algorithm is divided into two parts. In Algorithm 4.1, the OFR algorithm using the distance correlation dependency metric is described. It is important to mention that in contrast with the original algorithm developed by Billings, *et.al.* [5], that requires a threshold in the ESR, the user needs to specify the maximum number of terms m_{max} that the algorithm will look for, as proposed in Section 3.4.3. In this algorithm, lines 1-4 search for the candidate term that has the most significant influence on the system output based on the distance correlation metric. Once found, lines 5-9 create an orthogonal projection of $\mathbf{y}$ with respect to $\mathbf{q}_1$ using the modified Gram-Schmidt process. This orthogonalisation sequence is repeated in lines 11-24 until the maximum number of models m_{max} specified by the user is achieved. To avoid redundant candidate terms, lines 14-16 are introduced, which check the squared norm-2 of a candidate term, and if it is less than 10^{-10}, it is simply removed. Following [47], the concept of Leave-One-Out Cross Validation (LOOCV) is introduced in order to prevent under- and overfitting. Every time a new model term is added, the LOOCV statistic is computed with its standard error (SE) using the following equations:

$$LOOCV = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{e_i}{1-h_i}\right)^2 \tag{4.4}$$

Algorithm 4.1 Orthogonal Forward Regression using distance Correlation

```
Input: Dictionary $D = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\}$, output signal $\mathbf{y}$, maximum number of terms $m_{max}$
Output: NARX model with significant terms selected from $D$ and corresponding parameters $\boldsymbol{\theta}$ estimated
for all $\boldsymbol{\phi}_i$ in $D$ do
    Define $\mathbf{w}_i = \phi_i/\|\phi_i\|_2$
    Compute $dCor^{(i)}\left(\mathbf{w}_i, \mathbf{y}\right)$
Find $j = \underset{1 \leq i \leq M}{max}\left\{dCor^{(i)}\left(\mathbf{w}_i, \mathbf{y}\right)\right\}$
Define $\mathbf{q}_1 = \mathbf{w}_j$
Define $\mathbf{p}_1 = \phi_j$
Train a linear regression model using $\mathbf{y}$ and $\mathbf{p}_1$
Define $g_1 = \mathbf{q}_1^T\mathbf{y}$
Define $\mathbf{y}_{new}^{(1)} = \mathbf{y} - g_1\mathbf{q}_1$
Compute LOOCV with standard error and store them
Remove $\boldsymbol{\phi}_j$ from $D$
for $s = 2$ to $m_{\max}$ do
    for all $\boldsymbol{\phi}_i$ in $D$ do
        Orthonormalize $\boldsymbol{\phi}_i$ with respect to $[\mathbf{q}_1, \ldots, \mathbf{q}_{s-1}]$ to obtain $\mathbf{w}_i$
        if $\mathbf{w}_i^T\mathbf{w}_i < 10^{-10}$ then
            Remove $\boldsymbol{\phi}_j$ from $D$
            Go to next iteration
        Compute $dCor^{(i)}\left(\mathbf{w}_i, \mathbf{y}_{new}^{(s-1)}\right)$
    Find $j = \underset{1 \leq i \leq M-s+1}{max}\left\{dCor^{(i)}\left(\mathbf{w}_i, \mathbf{y}\right)\right\}$
    Define $\mathbf{q}_s = \mathbf{w}_j$
    Define $\mathbf{p}_s = \phi_j$
    Train a linear regression model using $\mathbf{y}$ and $\mathbf{p}_1, \ldots, \mathbf{p}_s$
    Define $g_s = \mathbf{q}_s^T\mathbf{y}_{new}^{(s-1)}$
    Define $\mathbf{y}_{new}^{(s)} = \mathbf{y}_{new}^{(s-1)} - g_s\mathbf{q}_s$
    Compute LOOCV with standard error and store them
    Remove $\boldsymbol{\phi}_j$ from $D$
Using the stored LOOCVs, select the most parsimonious model with $m \leq m_{max}$ terms that satisfies the one standard deviation rule
Solve $\mathbf{A}_{m\times m}\boldsymbol{\theta}_{m\times 1} = \mathbf{g}_{m\times 1}$
Return matrix of terms selected $\boldsymbol{\Phi} = \left[\begin{array}{cccc}\boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \ldots & \boldsymbol{\phi}_m\end{array}\right]$ and vector of coefficients $\boldsymbol{\theta} = \left[\begin{array}{cccc}\theta_1 & \theta_2 & \ldots & \theta_m\end{array}\right]^T$
```

$$SE = \sqrt{\frac{1}{N}\underset{i\epsilon\{1,\ldots,N\}}{Var}\left[\left(\frac{e_i}{1-h_i}\right)^2\right]} \tag{4.5}$$

where e_i is the residual obtained from fitting the model to all N observations using the selected candidate terms at each iteration s, and h_i are the diagonal values of the influence matrix for the fitted model [82]. Once the maximum number of terms m_{max}

is achieved, the most parsimonious model with $m \leq m_{max}$ terms is selected in line 27 using the one standard deviation rule [54], i.e. select the simplest model for which the LOOCV is within one standard error from the minimum LOOCV. Finally, the parameters $\boldsymbol{\theta}$ are computed in line 28, and the algorithm returns them together with the significant terms selected. The parameter m_{max} can be selected heuristically, by running Algorithm 4.1 a couple of times and inspecting the resulting LOOCV curve. In case the best model contains exactly m_{max} model terms, this means that the appropriate number of model terms may be beyond this value; therefore it could be increased to find a better model.

The proposed BFOR-dCor algorithm is described in Algorithm 4.2. Here, Algorithm 4.1 is repeated R times, each with a different bootstrap realisation taken from the original input and output signals. Every time a bootstrap realisation is used, the identified model is recorded in a table. After all the R bootstrap realisations are taken, the table is summarised to identify the different models that are found, and each of them is assigned a value that is equal to the number of times it is selected within the R bootstrap realisations.

Algorithm 4.2 Bagging Forward Orthogonal Regression using Distance Correlation

```
Input: Number of bootstrap realisations $R$, block length $B$, dictionary $D = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\}$, output signal $\mathbf{y}$, maximum number of terms $m_{max}$
Output: Table with $R$ models
for all $i \in \{1, \ldots, R\}$ do
    Obtain a bootstrap realisation by applying the moving blocks bootstrap method to $D$ and $\mathbf{y}$ using a block length $B$
    Apply Algorithm 4.1 to the bootstrap realisation
    Record the identified model in a table
Summarise the table to identify the different models
Rank each model with respect to the number of votes
Return table with ranking
```

The BFOR-dCor algorithm is a new method that has been applied for the first time to nonlinear model selection. The proposed algorithm outperforms the conventional OFR algorithm in that the new method aims to find correct model terms within noisy data by introducing a voting mechanism in the algorithm. The algorithm will be demonstrated in the following section.

4.5 Case studies

In this section, several examples are provided to illustrate the effectiveness of the BFOR-dCor algorithm. First, a comparison of the new method with both the traditional OFR-ERR algorithm and the recent Forward Orthogonal Regression using Mutual Information (FOR-MI) [64] algorithms is performed. Second, the BFOR-dCor technique is applied to a testing model in [58] where the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm is applied. Finally, the BFOR-dCor algorithm is applied to the sunspot data provided by the World Data Center for the Sunspot Index and Long-term Solar Observations at the Royal Observatory of Belgium in Brussels [126]. The data consists of annual number of sunspots from 1700 to 2013.

4.5.1 Comparison of the BFOR-dCor with OFR-ERR and FOR-MI

The following model is taken from [42]:

$$\begin{aligned} y\left(k\right) = &-0.5y\left(k-2\right) + 0.7y\left(k-1\right)u\left(k-1\right) + 0.6u^2\left(k-2\right) \\ &+ 0.2y^3\left(k-1\right) - 0.7y\left(k-2\right)u^2\left(k-2\right) + e\left(k\right) \end{aligned} \tag{4.6}$$

where the input $u\left(k\right) \sim \mathcal{U}\left(-1,1\right)$, that is $u\left(k\right)$ is evenly distributed over $[-1,1]$, and the error $e\left(k\right) \sim \mathcal{N}\left(0, 0.02^2\right)$. Following [42], the maximum lags for the input and output are chosen to be $n_u = n_y = 4$ and the nonlinear degree is $\ell = 3$. The stop criterion for the OFR-ERR and FOR-MI algorithms is when the ESR is less than 0.05, i.e. $\eta = 5\%$. A total of 500 input-output data points are generated and the same random seed is used to ensure a fair comparison. The results for the OFR-

ERR algorithm are shown in Table 4.1 and Figure 4.3. It can be seen that all the model terms selected are correct except for the first one. Likewise, the results for the FOR-MI algorithm are displayed in Table 4.2 and Figure 4.4. The four model terms selected are correct, still the algorithm failed to select the term $y^3(k-1)$. From Tables 4.1 and 4.2, both models failed to select all the true model terms in Eq. (4.6). It is interesting to notice that, except by the spurious term found by the OFR-ERR algorithm, the union set of the model terms found by the OFR-ERR and FOR-MI algorithms is equivalent to the true model terms set. As explained in [42], both the OFR-ERR and FOR-MI algorithms can be used at the same time to select the model terms based on the t-tests, however this example shows that the selection is still hard to perform as all the terms selected by both methods are statistically significant.

Term	**Parameter**		**ERR (%)**	**t-test**
	True	**Estimate**		
$y(k-4)\,u^2(k-2)$	0	0.30537	48.64	8.96
$u^2(k-2)$	0.6	0.49517	12.20	42.81
$y(k-2)$	-0.5	-0.64684	22.33	-43.03
$y(k-1)\,u(k-1)$	0.7	0.68973	9.46	29.90
$y^3(k-1)$	0.2	0.18835	4.50	17.50

Table 4.1: Identified NARX model for Eq. (4.6) using the OFR-ERR algorithm. Five model terms were identified with their corresponding estimated parameter values. The true parameter values are included for reference. The ERR for each model term is shown together with the corresponding t-test value. Notice that the first model term identified by the algorithm is spurious.

Term	**Parameter**		**Mutual Information**	**t-test**
	True	**Estimate**		
$y(k-2)$	-0.5	-0.487327	0.7948	-37.10
$u^2(k-2)$	0.6	0.618496	0.9245	83.38
$y(k-1)\,u(k-1)$	0.7	0.616732	1.0218	38.93
$y(k-2)\,u^2(k-2)$	-0.7	-0.639457	0.9498	-20.21

Table 4.2: Identified NARX model for Eq. (4.6) using the FOR-MI algorithm. Four model terms were identified with their corresponding estimated parameter values. The true parameter values are included for reference. The MI for each model term is shown together with the corresponding t-test value. Notice that the algorithm failed to select the term $y^3(k-1)$.

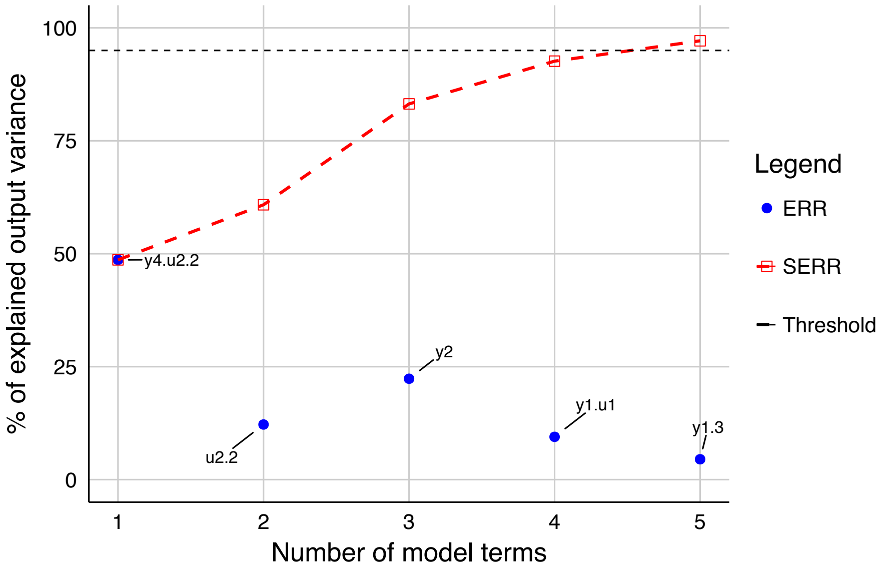

Figure 4.3: Model terms selected for Eq. (4.6) by the OFR-ERR algorithm with their corresponding ERR in blue dots, and the updated sum of ERR (SERR) represented in a red dashed line. The total SERR is 97.13%, which satisfies the ESR threshold of 5% shown as a horizontal black dashed line.

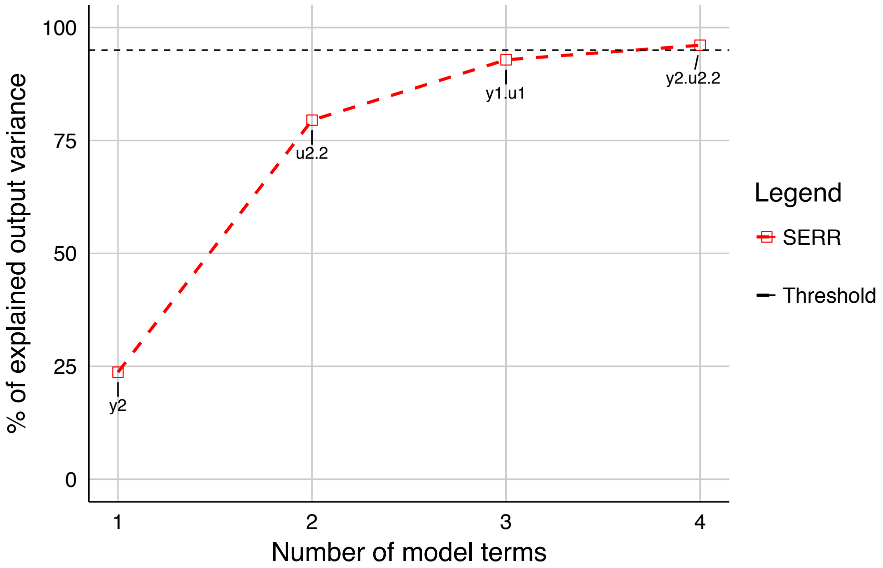

Figure 4.4: Model terms selected for Eq. (4.6) by the FOR-MI algorithm with the updated sum of ERR (SERR) represented in a red dashed line. The total SERR is 96.08%, which satisfies the ESR threshold of 5% shown as a horizontal black dashed line.

The BFOR-dCor algorithm is applied to Eq. (4.6) using a total of $R = 1000$ bootstrap realisations and a block length $B = 5$. The maximum number of terms to look for is $m_{max} = 10$. On Table 4.3, the 3 top model structures obtained by the BFOR-dCor algorithm are shown. These 3 model structures correspond to 96.5%

of the bootstrap realisations. The most-voted model structure has a structure that coincides with the true model in Eq. (4.6), something that is not obtained with the OFR-ERR and FOR-MI algorithms.

Model 1		**Model 2**	
Structure	# of votes	Structure	# of votes
$y\left(k-2\right)$	924	$y\left(k-4\right)u^{2}\left(k-2\right)$	30
$u^{2}\left(k-2\right)$		$u^{2}\left(k-2\right)$	
$y\left(k-1\right)u\left(k-1\right)$		$y\left(k-2\right)$	
$y\left(k-2\right)u^{2}\left(k-2\right)$		$y\left(k-1\right)u\left(k-1\right)$	
$y^{3}\left(k-1\right)$		$y\left(k-2\right)u^{2}\left(k-2\right)$	
		$y^{3}\left(k-1\right)$	

Model 3		**All Other Models**
Structure	# of votes	# of votes
$y\left(k-2\right)$	11	35
$u^{2}\left(k-2\right)$		
$y\left(k-1\right)u\left(k-1\right)$		
$y\left(k-2\right)u^{2}\left(k-2\right)$		
$y^{3}\left(k-1\right)$		
$y\left(k-3\right)u\left(k-3\right)$		

Table 4.3: Three top model structures identified for Eq. (4.6) using the BFOR-dCor algorithm. These model structures correspond to 96.5% of the bootstrap realisations.

For the 924 realisations that have the most-voted model structure, Figure 4.5 shows the beanplots [127] for each of the parameter estimates, which clearly suggest that each parameter bootstrap distribution is not Gaussian. Furthermore, Table 4.4 shows a statistical summary of the parameter estimates. It is interesting to notice that all but one of the true values are within two standard deviations from the mean. The exception is the $y^{3}\left(k-1\right)$ term. A frequency analysis may reveal an insightful understanding of the contribution of this term.

The results presented here show that the BFOR-dCor algorithm is able to identify 924 realisations with the true model structure together with a bootstrap distribution of the parameter estimates. Furthermore, having different equal-structure models is beneficial for the forecasting task since all the models or a sample from them can be used to compute an average prediction with the corresponding standard deviation.

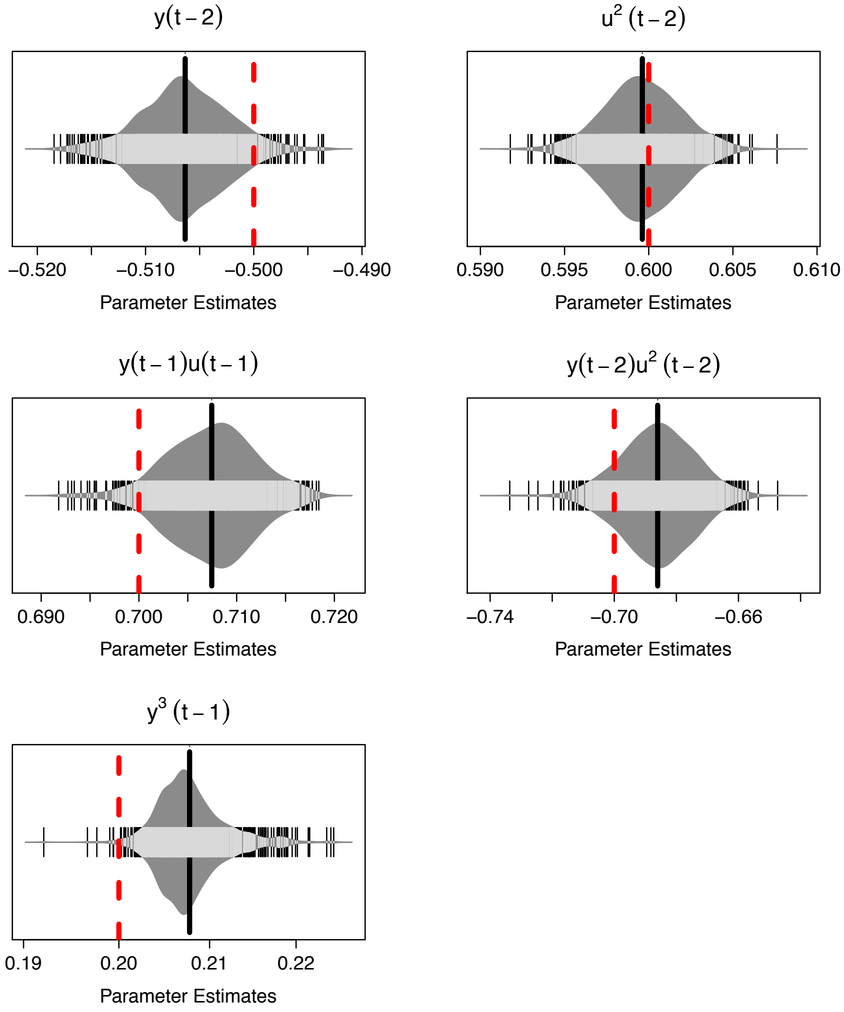

Figure 4.5: Beanplots for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for identification of Eq. (4.6). The vertical red dashed line represents the parameter's true value while the vertical black solid line represents the parameter's mean estimated value.

Term	Parameter		
	True	Mean	Standard Deviation
$y\,(k-2)$	-0.5	-0.5063	0.0039
$u^2\,(k-2)$	0.6	0.5996	0.0022
$y\,(k-1)\,u\,(k-1)$	0.7	0.7074	0.0044
$y\,(k-2)\,u^2\,(k-2)$	-0.7	-0.6860	0.0117
$y^3\,(k-1)$	0.2	0.2078	0.0037

Table 4.4: Statistical summary for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for identification of Eq. (4.6).

4.5.2 Comparison of the BFOR-dCor with RJMCMC algorithm

The following model is taken from [58]:

$$\begin{aligned} y\left(t\right) = & -0.5y\left(k-2\right)+0.7y\left(k-1\right)u\left(k-1\right) \\ & +0.6u^{2}\left(k-2\right)-0.7y\left(k-2\right)u^{2}\left(k-2\right)+e\left(k\right) \end{aligned} \tag{4.7}$$

In [58], the authors developed a computational Bayesian identification framework for NARMAX models that uses the RJMCMC algorithm to perform structure detection and parameter estimation together with a characterisation of the probability distribution over models. The algorithm is stochastic in nature, which encourages a global search over the model term space while at the same time ensuring that the identified model is parsimonious [58, 118]. In their work, the algorithm is executed 10 times on the same input-output data. From the 10 runs, the algorithm is able to get the true model structure 7 times. The main drawbacks of this method are that it is computationally expensive, and it needs to define different probability distributions for the parameters involved. Most of these distributions are chosen to be conjugate prior to ease the computations, but of course this does not mean that such distributions are faithful to the real unknown distributions. Because of the stochasticity of the RJMCMC algorithm, and some implementation issues, the results in [58] are difficult to reproduce. The reader is referred to this paper for further details.

The BFOR-dCor algorithm requires no assumptions about probability distributions and it can work extremely well once the basic parameters are defined. Here again the maximum lags for the input and output are $n_u = n_y = 4$ and the nonlinear degree is $\ell = 3$, exactly the same values as in [58]. A total of 500 input-output data points are generated. The BFOR-dCor algorithm is applied to Eq. (4.7) using a total of $R = 1000$ bootstrap realisations, a block length $B = 5$, and the maximum number of terms is $m_{max} = 10$. On Table 4.5, the 3 top model structures obtained by the BFOR-dCor algorithm are shown. These 3 model structures correspond to 88.1% of the bootstrap realisations. The most-voted model structure has a structure that coincides with the true model in Eq. (4.7).

Model 1			**Model 2**	
Structure	# of votes		Structure	# of votes
$y(k-2)$	839		$y(k-2)$	26
$u^2(k-2)$			$u^2(k-2)$	
$y(k-1)\,u(k-1)$			$y(k-1)\,u(k-1)$	
$y(k-2)\,u^2(k-2)$			$y(k-2)\,u^2(k-2)$	
			$y^2(k-2)\,y(k-4)$	

Model 3			**All Other Models**
Structure	# of votes		# of votes
$y(k-2)$	16		119
$u^2(k-2)$			
$y(k-1)\,u(k-1)$			
$y(k-2)\,u^2(k-2)$			
$y(k-3)\,u(k-3)$			

Table 4.5: Three top model structures identified for Eq. (4.7) using the BFOR-dCor algorithm. These model structures correspond to 88.1% of the bootstrap realisations.

Figure 4.6 shows the beanplots for each of the parameter estimates, which suggest that each parameter may be treated as a Gaussian random variable. Likewise, Table 4.6 shows a statistical summary of the parameter estimates. It is interesting to notice that all the true values are within two standard deviations from the mean.

Term	**Parameter**		
	True	**Mean**	**Standard Deviation**
$y(k-2)$	-0.5	-0.5046	0.0041
$u^2(k-2)$	0.6	0.6000	0.0023
$y(k-1)\,u(k-1)$	0.7	0.7067	0.0045
$y(k-2)\,u^2(k-2)$	-0.7	-0.6839	0.0118

Table 4.6: Statistical summary for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for identification of Eq. (4.7).

These results show that the BFOR-dCor algorithm is efficient and works well without the need of assumptions of probability distributions.

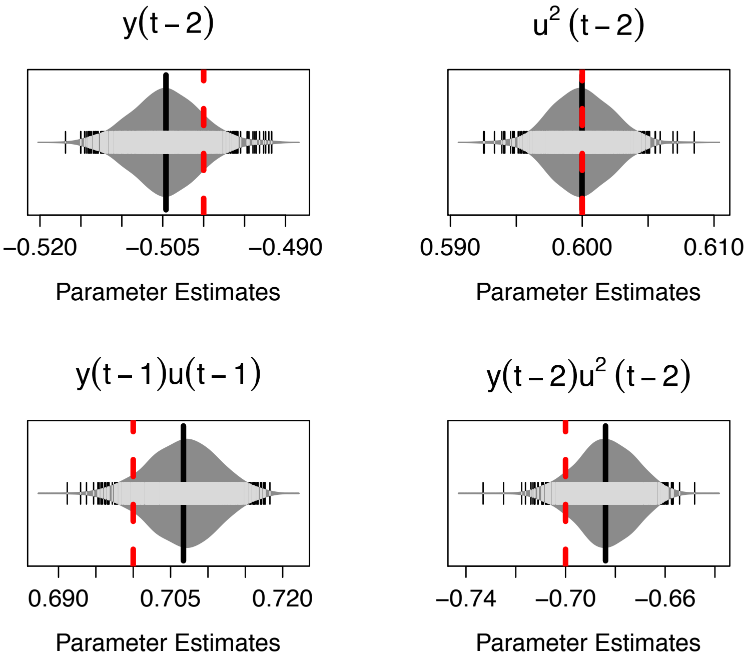

Figure 4.6: Beanplots for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for identification of Eq. (4.7). The vertical red dashed line represents the parameter's true value while the vertical black solid line represents the parameter's mean estimated value.

4.5.3 Forecasting the annual sunspot number

The sunspot time series provided by the World Data Center for the Sunspot Index and Long-term Solar Observations at the Royal Observatory of Belgium in Brussels [126] consists of 314 observations of the annual number of sunspots from 1700 to 2013. The data from 1700 to 1950 is used for structure detection and parameter estimation while the data from 1951 to 2013 is used for model performance testing and validation. It is assumed that the annual number of sunspots depends only on previous annual observations, i.e. $n_u = 0$. Furthermore, it is well-known that the Sun's north and south poles reverse around every 11 years which corresponds to a period of great solar activity known as the solar max [128]. Therefore, $n_y = 12$ is chosen, and a Nonlinear AutoRegressive (NAR) model with nonlinear degree $\ell = 3$ is employed to test the performance of the proposed BFOR-dCor algorithm.

The BFOR-dCor algorithm is applied using a total of $R = 1000$ bootstrap realisations, a block length $B = 15$, and the maximum number of terms is $m_{max} = 15$.

The 5 top model structures obtained by the BFOR-dCor algorithm are shown in Table 4.7, which correspond to 7.2% of the bootstrap realisations.

Model 1		**Model 2**	
Structure	# of votes	Structure	# of votes
$y\left(k-1\right)y\left(k-10\right)$	30	$y\left(k-1\right)y\left(k-10\right)$	19
$y\left(k-2\right)y^2\left(k-10\right)$		$y\left(k-2\right)y^2\left(k-10\right)$	
constant		constant	
$y\left(k-1\right)$		$y^2\left(k-1\right)y\left(k-10\right)$	
$y\left(k-2\right)$		$y\left(k-1\right)$	
$y^2\left(k-1\right)y\left(k-10\right)$		$y\left(k-3\right)$	

Model 3		**Model 4**	
Structure	# of votes	Structure	# of votes
$y\left(k-1\right)y\left(k-9\right)$	12	$y\left(k-1\right)y\left(k-9\right)$	6
$y\left(k-2\right)y^2\left(k-9\right)$		$y\left(k-2\right)y^2\left(k-9\right)$	
$y\left(k-1\right)$		constant	
$y^2\left(k-1\right)y\left(k-9\right)$		$y\left(k-1\right)$	
$y\left(k-2\right)$		$y^2\left(k-1\right)y\left(k-9\right)$	
constant		$y\left(k-3\right)$	

Model 5		**All Other Models**
Structure	# of votes	# of votes
$y\left(k-1\right)y\left(k-10\right)$	5	823
$y\left(k-2\right)y^2\left(k-10\right)$		
constant		
$y\left(k-1\right)$		
$y\left(k-2\right)$		
$y^3\left(k-1\right)$		
$y^3\left(k-2\right)$		

Table 4.7: Five top model structures from a total of 875 different models identified for the sunspot time series using the BFOR-dCor algorithm.

For the 30 realisations that have the most-voted model structure, Figure 4.7 shows the beanplots for each of the parameter estimates, which clearly suggest that most of the bootstrap parameter distributions are not Gaussian. Furthermore, Table 4.8 shows a statistical summary of the parameter estimates. Figures 4.8 and 4.9 show the OSA output and MPO together with the two standard deviation region, respectively. In both cases, from these two graphs it can be seen that a simple NAR model has successfully captured the general trend of the sunspots behaviour. The RMSE for the OSA output is 19.39716 while the RMSE for the MPO is 28.77858.

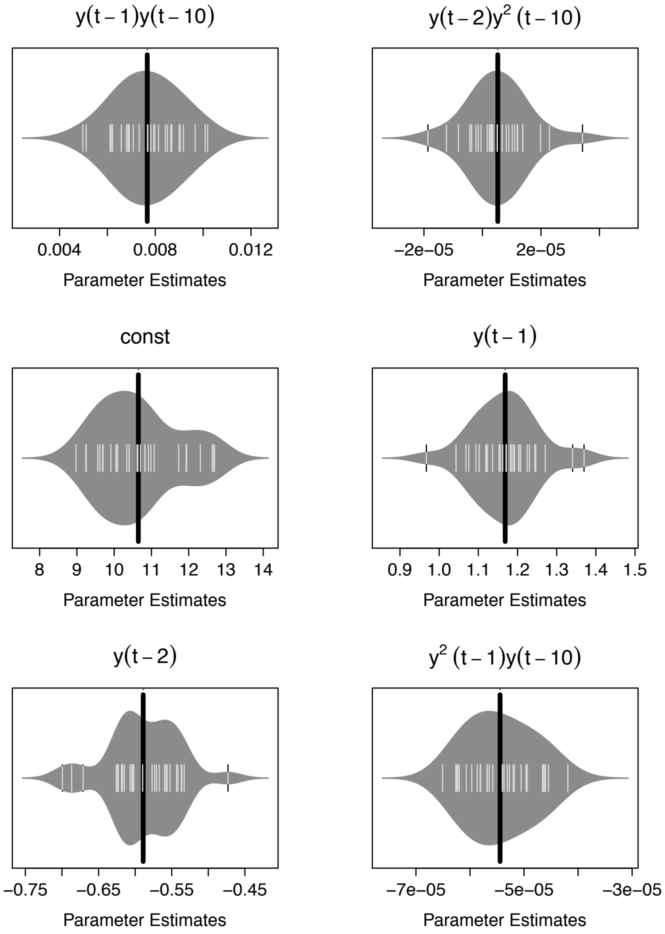

Figure 4.7: Beanplots for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for forecasting the annual sunspot number. The black solid line represents the parameter mean estimated value.

Term	**Parameter**	
	Mean	**Standard Deviation**
$y\,(k-1)\,y\,(k-10)$	0.007665	1.3371e-03
$y\,(k-2)\,y^2\,(k-10)$	5.239e-06	1.0521e-05
constant	10.649	1.0723
$y\,(k-1)$	1.1685	8.3556e-02
$y\,(k-2)$	-0.5890	4.8301e-02
$y^2\,(k-1)\,y\,(k-10)$	-5.443e-05	6.1261e-06

Table 4.8: Statistical summary for the parameter estimates of the model terms identified in the most-voted model structure using the BFOR-dCor algorithm for forecasting the annual sunspot number.

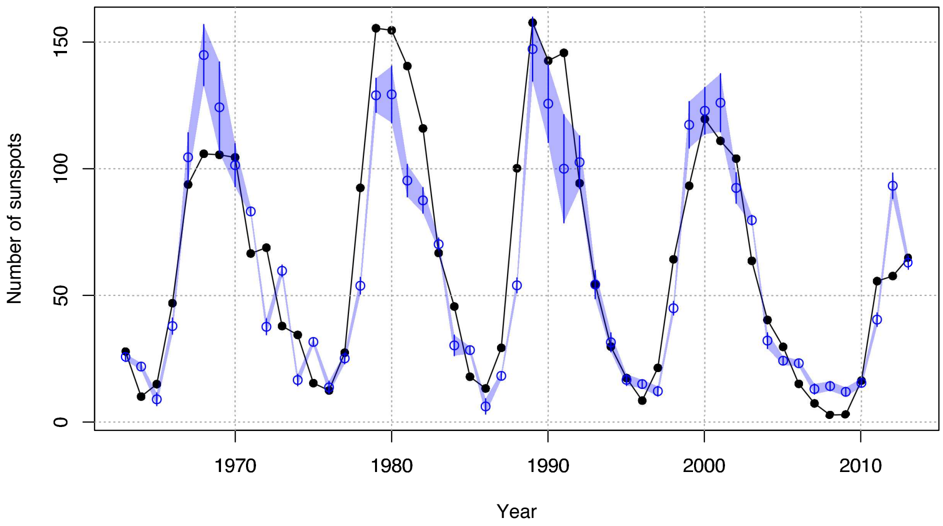

Figure 4.8: OSA predicted output for the sunspot time series using the most-voted model structure identified by the BFOR-dCor algorithm. The black solid line with circles indicates the true measurements. The empty blue circles represent the OSA predicted output. The blue shadow represents the two standard deviation region.

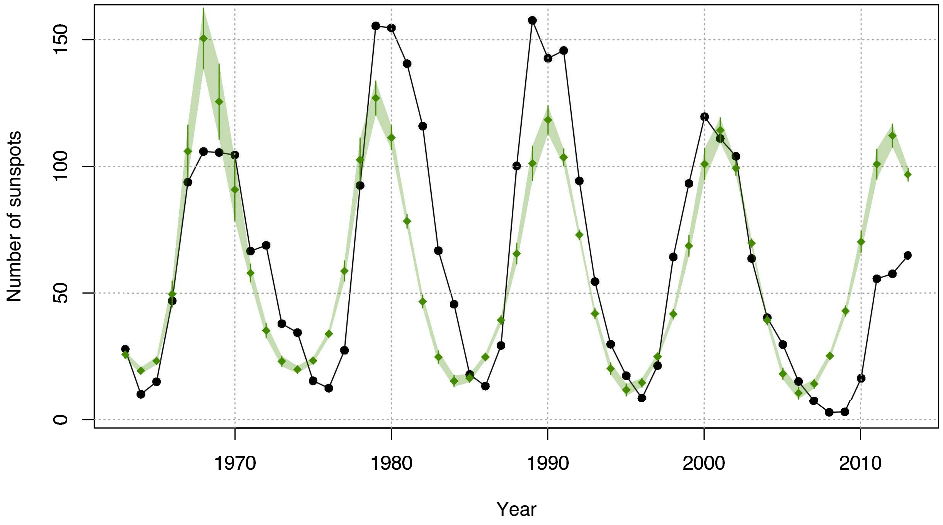

Figure 4.9: MPO for the sunspot time series using the most-voted model structure identified by the BFOR-dCor algorithm. The black solid line with circles indicates the true measurements. The green diamonds represent the MPO. The green shadow represents the two standard deviation region.

Figure 4.10 shows the statistical correlation tests given by Eq. (2.17) for the OSA output of the most-voted NAR model identified by the BFOR-dCor algorithm. It can be seen that the second and third tests, i.e. $\phi_{\xi'(\xi^2)'}(\tau) = 0$ and $\phi_{(\xi^2)'(\xi^2)'}(\tau) = \delta(\tau)\ \forall\tau$, are not ideally satisfied, suggesting that autoregressive models may not be sufficient to fully characterise the entire dynamics of the process. Nevertheless, the results obtained by the BFOR-dCor algorithm are still remarkable given the complexity of the system.

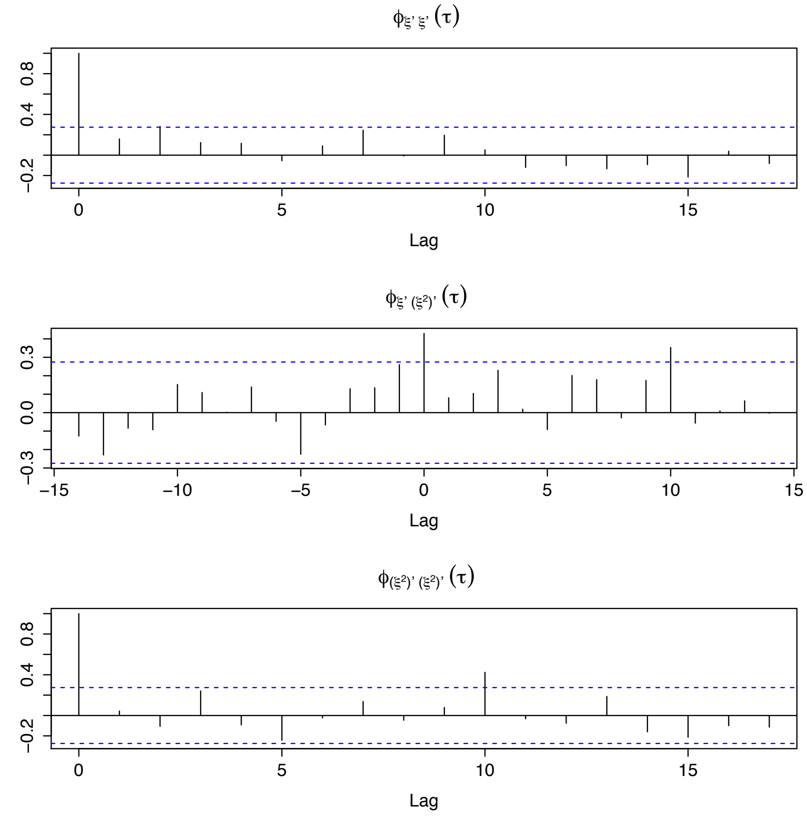

Figure 4.10: Statistical correlation tests given by Eq. (2.17), with 95% confidence limits, for the OSA predicted output of the most-voted NAR model identified for the sunspot time series using the BFOR-dCor algorithm. The second and third tests, i.e. $\phi_{\xi'(\xi^2)'}(\tau) = 0$ and $\phi_{(\xi^2)'(\xi^2)'}(\tau) = \delta(\tau)\ \forall\tau$, are not ideally satisfied because the correlations have several values well outside the 95% confidence bands, which means that certain nonlinearities are not captured by the model

4.6 Discussion

The BFOR-dCor algorithm provides an alternative to the deterministic notion of the OFR algorithm. From the results obtained, it can be seen that training several models by resampling the data offers more information about the uncertainty in the model structure together with the parameter estimates. The main advantage from this alternative comes from the fact that prior distribtions are not required, which can be difficult to define. Also, by checking the most common model structures, it is possible to determine the most important model terms, which can be analysed separately with other tools like frequency analysis [5] or empirical mode decomposition [129]. However, the main disadvantage of this approach is that by resampling the data set, there is no guarantee that a dominant model structure will be identified. Even if it is found, there is little variability in the trained models, which increases the bias in the predictions. An alternative for this would be to consider a similar approach like in the tree-based models, i.e. random forests, where the inputs are chosen randomly and many poor-performance models are trained. This reduces the bias while increasing the variability in the forecasts, however the latter is overcomed when the average of the predictions is computed.

4.7 Summary

A new algorithm for model structure detection and parameter estimation has been developed. This new algorithm combines two different concepts that enhance the performance of the original OFR algorithm. First, the distance correlation metric is used, which measures all types of nonlinear or non-monotone dependencies between random vectors. Second, the bagging method is implemented, which produces different models for each resample from the original data set. Identified models, or a subset of them, can be used together to generate improved predictions via averaging (for regression problems) or via voting (for classification problems). A main advantage of these concepts in the new BFOR-dCor algorithm is that it provides the interpretability of nonlinear dependencies and an insightful uncertainty analysis. The algorithm can be slow since the distance correlation is a complex computation compared with other metrics; nevertheless it produces results that outperform its counterparts and requires no assumptions of probability distributions like the RJM-

CMC algorithm. All these have been demonstrated through numerical case studies. The results of this chapter were published in [30].

Chapter 5

Modelling of the Atlantic Meridional Overturning Circulation

5.1 Introduction

The Atlantic Meridional Overturning Circulation (AMOC) is an important component of the Atlantic Ocean that is composed of a flow of warm water in the surface, and a returning flow of cold water beneath it. The AMOC plays a key role in the Earth's climate system given that it transports heat from the Tropics and Southern Hemisphere toward the North Atlantic. This has an important influence in the global climate system, particularly in the weather of Western Europe and North America.

The study of the AMOC has been more easily accesible due to the deployment of the RAPID array at 26ºN in the Atlantic Ocean in spring 2004. This has enable the acquisition of measurements that can be used to understand its dynamics.

In this chapter the use of the NARX methodology is applied for the first time to describe the dynamics of the AMOC anomaly. Different sets of variables are taken into account, which involve the North Atlantic Oscillation (N) index and density variables from the Gulf of Mexico, the Labrador Sea and the Norwegian Sea. The task is challenging given that the sample size is small. Based on what is discussed in Chapter 3, two dependency metrics are used to find significant model terms, together with penalty and performance metrics to identify the model structure. The best model identified is used not only for forecasting but also for hindcasting the AMOC back to January of 1980.

5.2 Data set description

The AMOC is an important component of the Atlantic Ocean's circulation. The AMOC is composed of a poleward flow of surface warm water that is above a deep flow of colder water from the North Atlantic. It is believed that the AMOC influences several meteorological phenomena including the weather of western Europe and the sea-level on the eastern seaboard of the USA [25, 130].

In spring 2004, the RAPID array was deployed at 26ºN in the Atlantic Ocean. This array consists of moored instruments that estimate the meridional flow on a daily basis through continuous measurements of temperature and salinity [131, 132]. Figure 5.1 shows the AMOC time series, with the most recent acquisition ocurring in September 2015. Such measurements have provided valuable information and there is a great interest not only on predicting future behaviour, but also in investigating past natural variability in the AMOC. Climate or ocean models [133], and ocean reanalysis reconstruction [134–137] have been used for such purposes. Although there is not a general agreement on the dynamics of the AMOC, several climate model simulations suggest that the it will decline in strength during the 21st century due to increasing greenhouse gases [132, 138].

The data set that is used consists of the variables shown in Table 5.1. These were selected and gathered by Prof. Grant Bigg from the Department of Geography at the University of Sheffield. The variable of interest consists of mean AMOC values from the RAPID array that were measured monthly from April 2004 to September 2015. The input variables used for the model are composed of two major types, an atmospheric one and three ocean density variables.

The AMOC strength has an upper ocean component directly related to the

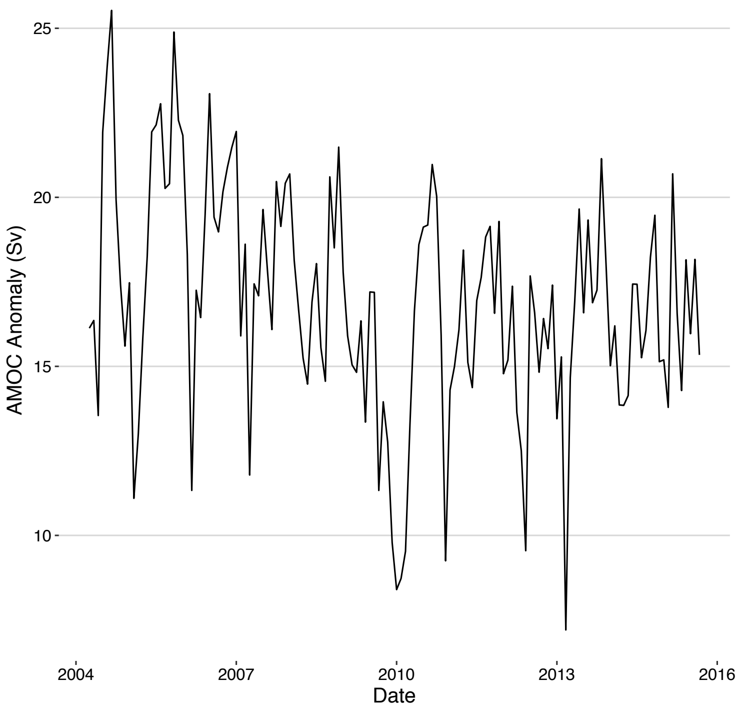

Figure 5.1: Monthly observed values of the AMOC over April 2004 - September 2015. The units are Sverdrups (Sv), equivalent to $10^6\ {}^{m^3}\!/_{s}$.

Variable	**Symbol**	**Description**
Input	N	Standardized North Atlantic Oscillation index
	GM	Density over the Gulf of Mexico $[{}^{kg}\!/_{m^3}]$
	LS	Density over the Labrador Sea $[{}^{kg}\!/_{m^3}]$
	NS	Density over the Norwegian Sea $[{}^{kg}\!/_{m^3}]$
Output	AMOC	AMOC strength [Sv] (variable of interest)

Table 5.1: Data set variables for modelling the Atlantic Meridional Overturning Circulation. The units of the AMOC are Sverdrups (Sv), equivalent to $10^6\ {}^{m^3}\!/_{s}$. These variables were selected and gathered by Prof. Grant Bigg from the Department of Geography at the University of Sheffield

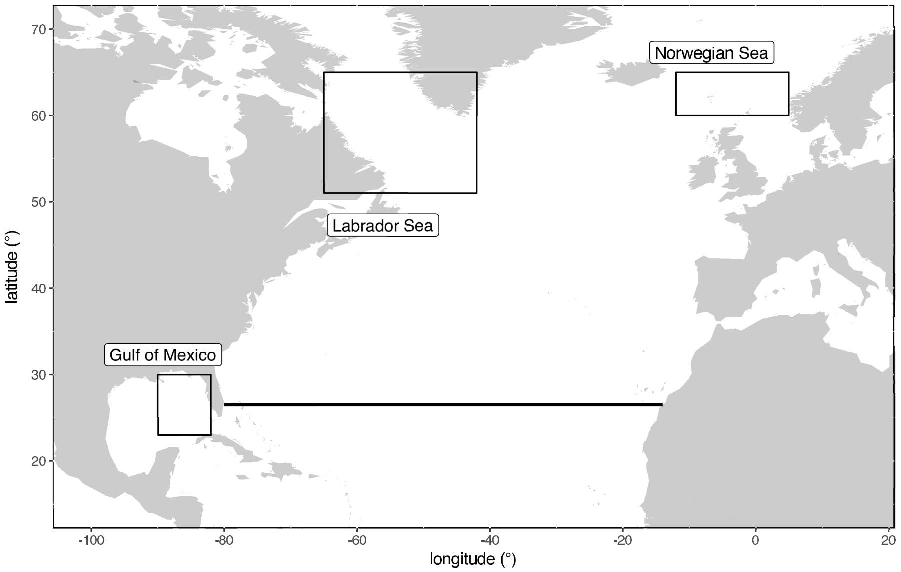

Figure 5.2: Map showing the position of the RAPID line (in black) deployed at 26ºN in the Atlantic Ocean. The regions chose to calculate regional surface density variables are highlighted in squares. This choice involves a mix of northern regions, where winter convection and deep water formation occurs, and a southern region in the Gulf of Mexico, from which the upper ocean waters feeding the main northward flow of the Gulf Stream derive.

wind strength. To represent this at basin-scale, the North Atlantic Oscillation (N) index is chosen because of its strong links to the relative strength and locations of the central North Atlantic atmospheric pressure systems [139]. Monthly values of the N index were taken from the Climate Prediction Center, which is part of the National Oceanic and Atmospheric Administration's National Weather Service in the USA [140]. With respect to the ocean density variables, these are the surface density averaged over the region in the Gulf of Mexico (GM) 23-30ºN, 82-90ºW, the surface density averaged over the Labrador Sea (LS) region 51-65ºN, 42-65ºW, and the surface density over the southern Norwegian Sea (NS) area 60-65ºN, 5ºE-12ºW (Figure 5.2).

This choice of ocean density variables involves a mix of northern regions, where winter convection and deep water formation occurs, and a southern region in the Gulf of Mexico, from which the upper ocean waters feeding the main northward flow of the Gulf Stream derive. Computation of the three density variables involved downloading surface potential temperature and salinity data over the respective

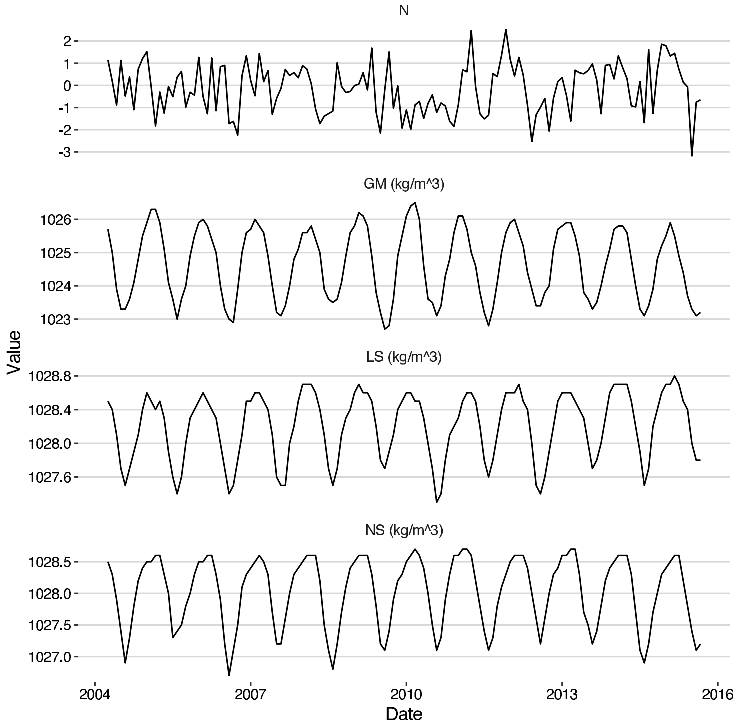

Figure 5.3: Monthly observed values of the N (standardized NAO index), density variables GM (Gulf of Mexico), LS (Labrador Sea) and NS (Norwegian Sea) over April 2004 - September 2015.

areas, and the use of the density formula given by [141] at zero pressure. Time series of the input variables involved in the data set are shown in Figure 5.3.

5.3 Methodology

The NARX methodology is employed for the analysis of the AMOC data set. As discussed in chapter 2, the OFR algorithm uses the ERR index to identify the most significant predictors that explain the output variable's variance. However, the ERR is only able to detect linear dependencies. Another concern is related to the stop criterion when building a model. Originally, when the sum of the ERR values of the predictors selected is above a given threshold η, the model training process is stopped. This of course requires a careful selection of the threshold. If it is too small, the identified model cannot capture the dynamics of the system completely. However, if the threshold is too large, it can lead to an overfitted model that does not generalize well to new observations. Based on the improvements discussed to the OFR algorithm in Section 3.4.3, in this chapter two model selection approaches are complementarily used to select the most appropriate number of model terms. The first one makes use of penalty metrics, which provide a relative quality of the model, penalising those models that are too complex. The second approach makes use of performance metrics, where the data is separated into training and test sets.

Based on the data description given in Section 5.2, three sets of variables are used to build three different NARX models of the AMOC. These are summarised below:

- Case 1: given the similarity of the three density variables as shown in Figure 5.3, they are combined in a single variable by taking the mean value. This case detects the relative contributions of the atmosphere and ocean mean states. The variables to consider are:

  - AMOC strength (output variable)
  - NAO index (input variable)
  - Mean of the density variables (input variable) defined as

  $$U = \frac{GM + LS + NS}{3} \tag{5.1}$$

- Case 2: given that the AMOC is a current between the water of the Gulf of Mexico and the northern seas, the difference in the densities of these regions is taken into account. This cases focuses on the relative contributions of the atmosphere and the meridional density difference between surface and deep water source waters. The variables to consider are:

  - AMOC strength (output variable)
  - NAO index (input variable)
  - Difference of the density variables (input variable) defined as

  $$V = \frac{LS + NS}{2} - GM \tag{5.2}$$

- Case 3: this cases combines the two previous cases to detect the relative contributions of the atmosphere and the contrasting mean and meridional differences in ocean density. The variables to consider are:
  - AMOC strength (output variable)
  - NAO index (input variable)
  - Mean of the density variables U (input variable)
  - Difference of the density variables V (input variable)

Note that the main objectives of this study are twofold: a) to investigate which input variables are the most important, and how the change of AMOC depends on the interactions of these important input variables; and b) to investigate the predictive power of these important variables for forecasting the AMOC. We therefore do

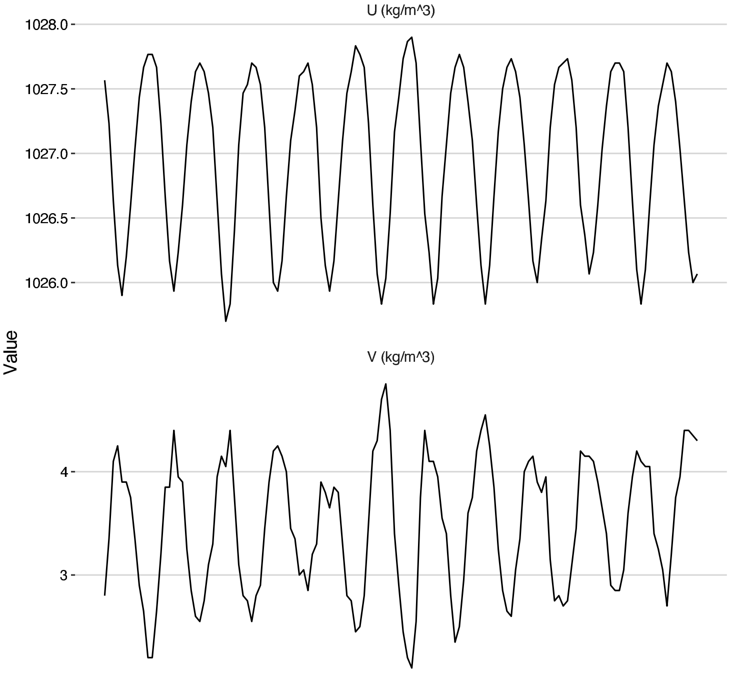

Figure 5.4: Monthly observed values of U (mean of density variables given by Eq. 5.1) and V (difference of density variables given by Eq. 5.2) over April 2004 - September 2015. It is noteworthy that these variables, while retaining the annual cycle, have opposite extremes, namely, the highest value of the mean density (U) is during the winter, while the largest difference in density (V) occurs during the summer.

not consider autoregressive model terms (i.e. lagged AMOC terms are not included in the models).

The time series for the new variables U and V are shown in Figure 5.4. From the time series, it is noteworthy that these variables, while retaining the annual cycle, have opposite extremes, namely, the highest value of the mean density (U) is during the winter, while the largest difference in density (V) occurs during the summer.

The data set is divided in three parts. The first part contains data from April 2004 to March 2013, which is used for training several models using the ERR and MI indices, together with penalty and performance metrics. The second part uses data from April 2013 to March 2014 for model validation/comparison and model evaluation. The last part contains data from April 2014 to September 2015, which is used to test models' predictive performance on data that are not used in the model identification and selection phase. Furthermore, the fitting performance over the training set and the prediction performance over the validation set are treated equally. This allows the computation of an average evaluation metric that helps to build a model that captures efficiently the system dynamics without under- or overfitting the data.

It is important to mention that for each of the three variables, AMOC, U, and V, the corresponding mean value is removed prior to the model building procedure. The mean values of the three variables, estimated based on the training data (i.e. data from April 2004 to March 2013), are 16.97 Sv for AMOC, 1026.98 $^{kg}/_{m^3}$ for U, and 3.4 $^{kg}/_{m^3}$ for V, respectively. This is done partly because the magnitudes of the density variables are much larger than the N index and the AMOC strength. Removing the mean value ensures that the density variables do not dominate the training and validation phases, and that the resulting models are more robust.

A nonlinear model term and variable selection procedure proposed in [46] is applied, and numerical experimental results suggest that $n_u = 8$ is an appropriate choice. For convenience, polynomials of nonlinear degree $\ell = 2$ are employed.

5.4 Results

5.4.1 Model Training and Validation

For all three cases, the models with the best performance are those selected by means of the ERR metric using cross-validation. These are shown in Tables 5.2-5.4. Note that the variables reported in Tables 5.2-5.4 are mean-removed. So, for example, the model for Case 3 should be read as:

$$y\left(k\right) = -2.5\left[V\left(k-7\right) - 3.41\right] + 1.207N\left(k\right) - 1.240N\left(k\right)\left[U\left(k-6\right) - 1026.98\right]$$

Accordingly, the model predicted AMOC strength is:

$$AMOC\left(k\right) = y\left(k\right) + 16.97$$

Models for Cases 1 and 2 should be used in the same manner.

Each trained model is evaluated using the training and validation data sets up to March 2014. The average performance metrics for the three models are shown in Table 5.5. From these, it can be argued that the Case 2 model performs best overall. This suggests that the difference in density between the deep-water formation areas and the upstream Gulf Stream source region seven months ago provide the best indication of variation in the AMOC strength. Furthermore, an important observation is that all three cases agree that the current NAO index plays a discernible role in the AMOC strength.

The best model (i.e. the Case 2 model) is applied to the test data of April 2014 to September 2015, and its performance is shown in Table 5.6. Figure 5.5 shows a comparison between the model simulation output and the actual measurements. These show that the model captures the main dynamics of the AMOC process, although it is worth noting that the reduced anual cycle component of the AMOC in 2014/15 decreased the metric scores for the test period (Table 5.6) compared

Model Term	**Parameter**	**ERR (%)**
$U(k-7)$	2.221	17.95
$N(k)$	1.307	13.88
$N(k)U(k-6)$	-1.363	6.63
$N(k-8)U(k-3)$	1.096	4.42

Table 5.2: Identified NARX model for the AMOC strength using Case 1 scenario. Four model terms were identified with their corresponding estimated parameter values. The ERR for each model term is shown in the last column.

Model Term	**Parameter**	**ERR (%)**
$V(k-7)$	-2.449	20.60
$N(k)$	1.316	14.46
$N(k)V(k-6)$	1.237	5.27
$N(k-8)V(k-3)$	-1.065	5.10
$N(k-3)V(k-3)$	1.018	4.61

Table 5.3: Identified NARX model for the AMOC strength using Case 2 scenario. Five model terms were identified with their corresponding estimated parameter values. The ERR for each model term is shown in the last column.

Model Term	**Parameter**	**ERR (%)**
$V(k-7)$	-2.500	20.60
$N(k)$	1.207	14.46
$N(k)U(k-6)$	-1.240	5.90

Table 5.4: Identified NARX model for the AMOC strength using Case 3 scenario. Three model terms were identified with their corresponding estimated parameter values. The ERR for each model term is shown in the last column.

Case	**ME**	**RMSE**	**MAE**
1	-0.6376 Sv	2.3282 Sv	1.8603 Sv
2	**-0.2123 Sv**	**2.0761 Sv**	**1.6908 Sv**
3	-0.3479 Sv	2.2852 Sv	1.8940 Sv

Table 5.5: Average performance metrics on the training and validation data sets for each of the three Model Cases. The Case 2 model performs best overall suggesting that the difference in density between the deep-water formation areas and the upstream Gulf Stream source region seven months ago provide the best indication of variation in the AMOC strength.

ME	RMSE	MAE
-0.3573 Sv	2.6477 Sv	2.2210 Sv

Table 5.6: Performance metrics on the test set using the best model found (model from Case 2). The reduced anual cycle component of the AMOC in 2014/15 decreased the metric scores for the test period.

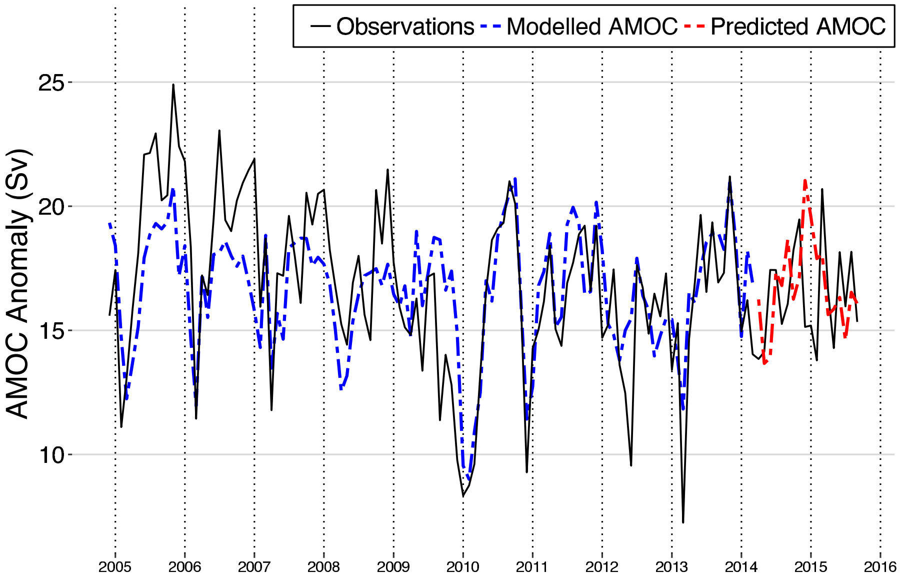

Figure 5.5: Modelled and predicted AMOC anomaly obtained using the best model found (model from Case 2). The blue line corresponds to the training and validation set, while the red line corresponds to the testing set. The model captures the main dynamics of the AMOC process, although it is worth noting that the reduced anual cycle component of the AMOC in 2014/15 decreased the metric scores for the test period compared to the training period.

to the training period. This phenomenon was observed at the RAPID Challenge (www.rapid.ac.uk/challenge), where many of the predictions also experienced difficulty in predicting this feature.

It is noteworthy that over the whole period of the RAPID data set, the correlation between the model simulation output (from the Case 2 model) and the observations, as shown in Figure 5.5, is 0.66, which is statistically significant well beyond the 1% level.

5.4.2 Nonlinear versus linear models

It is interesting to notice that the two leading terms of all three Model Cases shown in Tables 5.2-5.4 are linear. To test whether use of a nonlinear model has a statisti-

Model Term	Parameter	ERR (%)
$V(k-7)$	-3.132	20.60
$N(k)$	1.268	14.46
$U(k-4)$	-5.322	5.23
$U(k)$	-4.571	4.50
$V(k-3)$	-4.145	2.40
$V(k-1)$	-1.960	2.20
$U(k-7)$	-0.1045	1.71

Table 5.7: Identified NARX model of maximum degree 1 for the AMOC strength. Seven model terms were identified with their corresponding estimated parameter values. The ERR for each model term is shown in the last column.

ME	RMSE	MAE
0.5393 Sv	2.5678 Sv,	1.9481 Sv

Table 5.8: Average performance metrics on the training and validation data sets for NARX model of maximum degree 1. These metrics are significantly larger than the metrics for the Case 2 model suggesting that the purely linear model has an inferior performance.

cally significant improvement over use of a purely linear model, a NARX model of maximum degree 1 is developed for the training period. This is shown in Table 5.7. Such a model is applied to predict the AMOC strength; the average performance metrics of the linear model on the training and validation data sets are shown in Table 5.8, all of which are significantly larger than the metrics for the Case 2 model (in Table 5.5) and so clearly suggesting that the purely linear model is inferior to the Case 2 model.

The above statement can be confirmed by means of the Ramsey Regression Equation Specification Error Test (RESET) [142]. This test was designed to examine the null hypothesis that a linear model is enough to explain the output signal, whereas the alternative hypothesis suggests that the model has missed important nonlinearities. Mathematically, this test fits the linear part of Eq. (2.3), i.e.

$$y(k) = \theta_0 + \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + e(k) \tag{5.3}$$

and compares it with the model

$$y\left(k\right)=\theta_{0}+\sum_{i_{1}=1}^{n}\theta_{i_{1}}x_{i_{1}}\left(k\right)+\gamma_{1}\hat{y}^{2}\left(k\right)+\ldots+\gamma_{d-1}\hat{y}^{d}\left(k\right)+e\left(k\right) \tag{5.4}$$

where the polynomial degree $d \geq 2$, and the $\hat{y}\left(k\right)$ corresponds to the fitted values of Eq. (5.3). Under the null hypothesis of a correct linear specification in Eq. (5.4), then $\gamma_1 = \ldots = \gamma_{d-1} = 0$, which can be tested by the F-test, i.e. $F\left(d-1, N-\left(n+1\right)-d+1\right)$, on the joint significance of the parameters $\gamma_1, \ldots, \gamma_{d-1}$.

The results from the RESET test are shown in Table 5.9. These suggest that there is enough evidence to use a nonlinear model whose nonlinearity degree is $d = 2$ (i.e. the polynomial power is 2), to represent the preprocessed data, while not enough evidence is available to choose a model of power 3 (i.e. nonlinearity degree $d = 3$) at the 5% significance level.

Polynomial Degree	**P-value**
2	**4.591e-05**
3	0.9573

Table 5.9: P-values obtained from the Ramsey Regression Equation Specification Error Test (RESET) to determine the appropriate degree of the model. These suggest that there is enough evidence to use a nonlinear model whose nonlinearity degree is $d = 2$ to represent the preprocessed data, while not enough evidence is available to choose a model of power 3 at the 5% significance level.

5.4.3 Hindcasting

The Case 2 model is used to hindcast the AMOC strength back to January 1980. The hindcast and predicted AMOC values are shown in Figure 5.6. It is clear that the mean of the recovered AMOC from the model has changed little since 1980. The mean before the establishment of the RAPID array is 16.8±1.9 Sv, while since April 2004 it has become 16.8±2.3 Sv, showing no statistical difference in either the mean or variance. The tendency for an irregular annual cycle, with a winter minimum, a spring maximum, and a typical range of 2-3 Sv, also extends throughout the data set, although this has occasionally broken down in the past (e.g. around 1989) as during the RAPID program (e.g. around 2009).

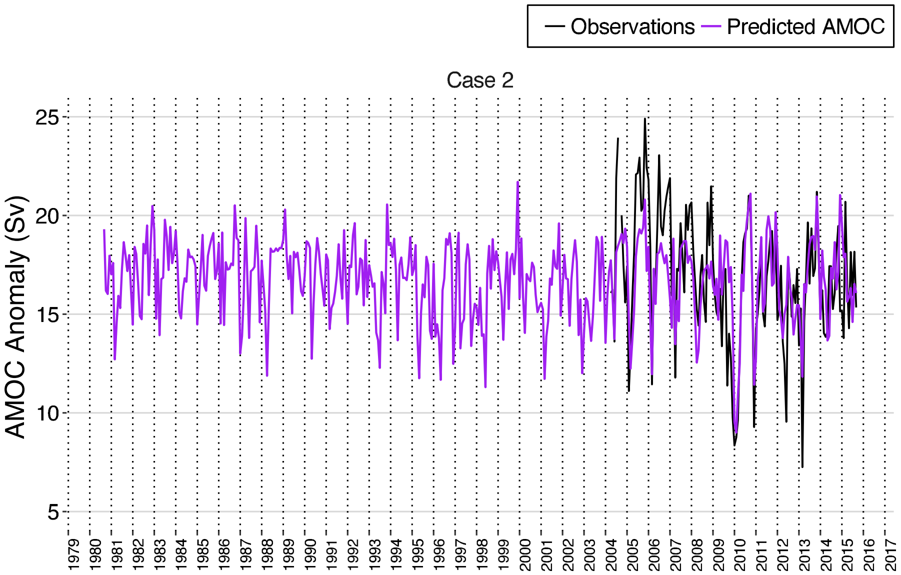

Figure 5.6: Hindcast and predicted AMOC obtained using the Case 2 model. It is clear that the mean of the recovered AMOC from the model has changed little since 1980.

From the results above, it can be seen that the NARX model from Case 2 matches reasonably well the AMOC dynamics. This gives confidence on the hindcast back to 1980.

5.5 Modelling of the AMOC with the BFOR-dCor algorithm

The model from Case 2 (Table 5.3) suggests the possibility of some predictive ability for the AMOC, because the dominant term contains a time lag of 7 months, through the density difference driving the variability. To confirm this, the BFOR-dCor algorithm is applied, which can provide insightful information about the most common model term given the limited data set size.

The BFOR-dCor algorithm is applied using a total of $R = 500$ bootstrap realisations, a block length $B = 10$, and the maximum number of terms is $m_{max} = 1$. The 3 top model terms are shown in Table 5.10. The most voted model term is $V(t-7)$ with 176 votes, which correspond to 35.2% of the bootstrap realisations. This confirms the results from the Case 2 model, where the density difference between the

Model Term	# votes
$V(t-7)$	176
$V(t-8)$	144
$N(t)V(t-6)$	98

Table 5.10: Three top model terms identified for the AMOC anomaly using the BFOR-dCor algorithm. The most voted model term is $V(t-7)$, confirming that the density difference between the northern sinking waters and the Gulf of Mexico source waters of the main overturning current with a dominant lag time of 7 months have a significant contribution in the prediction of the AMOC.

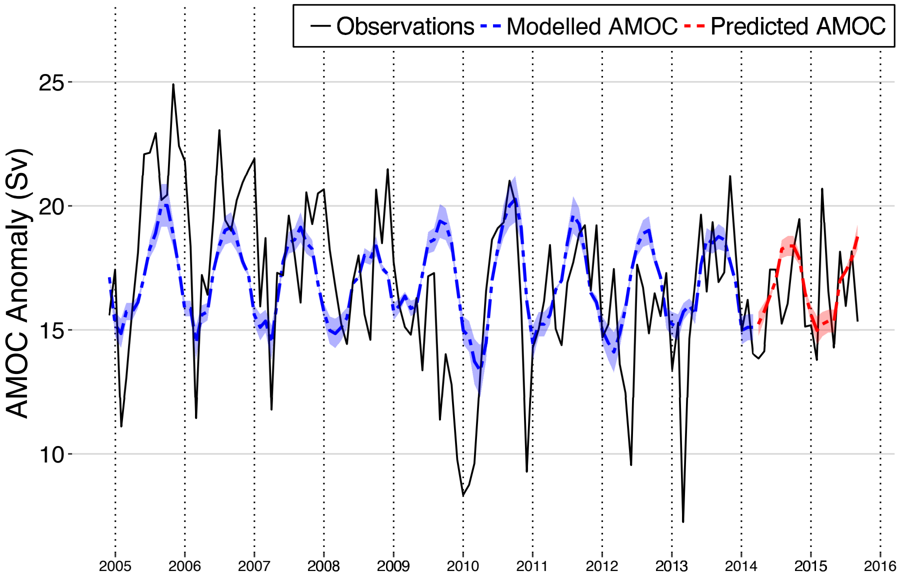

Figure 5.7: Modelled and predicted AMOC anomaly obtained using the average of the 176 models identified by the BFOR-dCor algorithm. The blue line corresponds to the training and validation set, while the red line corresponds to the testing set. The blue and red shadows represent the two standard deviation region.

northern sinking waters and the Gulf of Mexico source waters of the main overturning current with a dominant lag time of 7 months have a significant contribution in the prediction of the AMOC.

For comparison purposes, the 176 simpler models with the $V(t-7)$ term are used to forecast the AMOC. This is shown in Figure 5.7. Much less of the variability of the AMOC signal is captured when just using this term, but the correlation with the RAPID series from 2004 to 2015 is still a statistically significant 0.44. While such a simple model is clearly not of significant predictive usefulness in itself, it suggests that the AMOC may be predictable at least 6 months in advance.

5.6 Discussion

The identified models show that in many terms a dominant lag time tends to be around 6-8 months, particularly in V, the density difference between the convection regions and the Gulf Stream source. Previous studies agree with this timescale, where it is suggested that the AMOC variation is linked to boundary waves generated by density fluctuations in the Labrador Sea and then travelling south along the American shelf [137, 143].

This analysis also shows that, while the leading terms of each model are linear, the best model has distinct nonlinear components, involving a modulation of the wind and density difference variables. This nonlinearity is important in providing the best reproduction of the observed AMOC variation, and its inclusion is statistically robust. This nonlinearity is consistent with the nonlinear nature of many density-driven wave processes [141].

Nevertheless, details of the variation in the AMOC are not always well captured. The extrema during the training and test period are often under- or overestimated, although there are periods when these are captured well. In particular, it is notable that the extended reduction in observed AMOC strength around the beginning of 2010 is well predicted by the model (Figure 5.5). This is related to an extreme variation in the mean density difference V, between a peak maximum in 2009 and a peak minimum in 2010, associated with the prolonged negative excursion of the NAO index around this period (Figure 5.3), which led to the coldest winter in the UK since 1979 [144].

Looking at the longer model reconstruction, back to 1980, an element of decadal-scale change is visible (Figure 5.6). While there is essentially no trend over the whole record (-0.02 Sv/yr), the 1980s tended to have a higher modelled AMOC (17.2±1.7 Sv) than the late 1990s (16.2±2.1 Sv over 1995-1999). Furthermore, it is also notable that the hindcasted AMOC varies in a range approximately between 13 and 20 Sv. Rapid and significant change in the strength of the AMOC within this range is a characteristic of the longer term pattern, and recent changes since 2010 are not unprecedented.

5.7 Summary

In this chapter, the NARX modelling methodology is used to forecast and hindcast the Atlantic Meridional Overturning Circulation. Three cases are considered, each involving a different set of variables that include the relative contributions of the atmosphere and the contrasting mean and/or meridional differences in ocean density. Several models are trained using the ERR and MI indices, together with penalty and performance metrics as discussed in Chapter 3. The best models in each case are compared in order to select the most appropriate one for hindcasting based on three evaluation metrics. For this purpose, the model that is built using the meridional differences in ocean density is chosen and used to predict the AMOC back to 1980. In general, the NARX model captures reasonably well the inner dynamics even though the sample size is not large enough. Furthermore, it is found that the difference in ocean density has a significant contribution with a dominant lag time around 7 months. This was confirmed by the BFOR-dCor algorithm. This case study serves as an example of how the NARX R package can be used for modelling, as well as the opportunity to show the predictive power of NARX models to predict the strength of the AMOC in the subtropical North Atlantic.

Chapter 6

Logistic NARX model

6.1 Introduction

Many real-life systems involve a mixed combination of continuous and discrete variables. Binary responses are commonly studied in many situations, such as the presence or absence of a disease, granting a loan, or detecting the failure of a process, system or product [54, 117]. However, the use of traditional regression techniques to deal with systems with a dichotomous response variable may not be appropriate given that they are sensitive to outliers and the distribution of the classes [54]. In fact, the different versions of the NARX methodology have been designed under the assumption that the variables involved are continuous.

In this chapter, a *novel approach* is proposed that *combines logistic regression with the NARX methodology* focusing on systems with binary responses that depend on continuous predictors. The main motivation comes from the fact that logistic regression models are more suitable for binary classification problems given that they provide probabilities of belonging or not to a particular class. One important consideration when constructing a logistic regression model is multicollinearity, i.e. checking for high inter-correlations among the predictor variables. In the ideal scenario, the predictor variables will have a strong relationship to the dependent variable but should not be strongly related to each other [145]. However, it is not straightforward to select the predictor variables that satisfy this requirement. This problem is adequately solved using the NARX approach, since the model terms selected are orthogonal (uncorrelated) to each other. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models, something that is not achievable using other popular classification techniques like random forests [146], support vector machines [36] and k-nearest neighbors [17].

6.2 Logistic NARX Modelling Approach

Classification problems appear in several disciplines like, among others, finance, healthcare, and engineering, where the aim is to identify a model that is able to classify observations or measurements into different categories or classes. Many methods and algorithms are available which include logistic regression [54,117], random forest [146], support vector machines [36] and k-nearest neighbors [17]. The latter three are very popular but their major drawback is that they remain as black boxes for which the interpretation of the models may not be straightforward. Although it is possible to obtain an importance index for the predictors in the model, this does not help in understanding the possible inner dynamics of a system. On the other hand, logistic regression is an approach that produces a model to predict categorical outcomes. The predicted values are probabilities and are therefore restricted to values between 0 and 1 [145]. Logistic regression uses the logistic function defined as,

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{6.1}$$

where x has an unlimited range, i.e. $x \in \mathbb{R}$, and $f(x)$ is restricted to range from 0 to 1 [117]. One issue with logistic regression models is that they require the model terms and the interactions between them to be specified beforehand. This is problematic since it is important to always check for high inter-correlations among the predictor variables. In the ideal scenario, the predictor variables will be strongly related to the dependent variable but not strongly related to each other in order to avoid the multicollinearity problem [145].

The new approach combines the *logistic function* with the *NARX representation* in order to obtain a probability model

$$p\left(k\right)=\frac{1}{1+\exp\left[-\sum_{m=1}^{M}\theta_{m}\phi_{m}\Big(\boldsymbol{\varphi}\left(k\right)\Big)\right]} \tag{6.2}$$

The new algorithm is described in Algorithm 6.1 based on the original OFR algorithm. For convenience, let us assume that the output sequence $y\left(k\right)$ can be either $y\left(k\right)=1$ or $y\left(k\right)=0$ for $k=1,2,\ldots,N$, where $y\left(k\right)=1$ denotes the occurrence of the event of interest. Similar to chapter 4, instead of a threshold for the total of ERR, the user needs to specify the maximum number of terms m_{max} that the algorithm will look for [47]. Furthermore, traditionally the OFR algorithm relies on the ERR index given by Eq. (2.7) to determine the significance of a model term with respect to the output sequence. However, this metric is no longer useful given that the output is a binary sequence and the information from the class denoted as 0 would be lost. To overcome this issue, the biserial correlation coefficient is used, which measures the strength of the association between a continuous variable and a dichotomous variable [145]. The biserial correlation coefficient is defined as

$$r\left(\mathbf{x},\mathbf{y}\right)=\frac{\overline{X}_{1}-\overline{X}_{0}}{\sigma_{X}}\sqrt{\frac{n_{1}n_{0}}{N^{2}}} \tag{6.3}$$

where $\overline{X}_0$ is the mean value on the continuous variable X for all the observations that belong to class 0, $\overline{X}_1$ is the mean value of variable X for all the observations that belong to class 1, σ_X is the standard deviation of variable X, n_0 is the number of observations that belong to class 0, n_1 is the number of observations that belong to class 1, and N is the total number of data points.

In Algorithm 6.1, lines from 1 to 4 aim to find the candidate model term that makes the most significant contribution in explaining the variation of the system output measured by the biserial correlation coefficient. Once found, lines 5-8 create

Algorithm 6.1 Orthogonal Forward Regression for Logistic NARX models

Input: Dictionary of regressor vectors $D = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\}$, output signal $\mathbf{y}$, maximum number of terms m_{max}

Output: Logistic NARX model with significant terms selected from D and corresponding parameters $\boldsymbol{\theta}$ estimated

```
 1: for all $\boldsymbol{\phi}_i$ in $D$ do
 2:     Define $\mathbf{w}_i = {}^{\phi_i}/_{\|\phi_i\|_2}$
 3:     Compute $r^{(i)}(\mathbf{w}_i, \mathbf{y})$
 4: Find $j = \underset{1\leq i\leq M}{max} \left\{r^{(i)}(\mathbf{w}_i, \mathbf{y})\right\}$
 5: Define $\mathbf{q}_1 = \mathbf{w}_j$
 6: Define $\mathbf{p}_1 = \boldsymbol{\phi}_j$
 7: Train a logistic regression model using $\mathbf{y}$ and $\mathbf{p}_1$
 8: Compute the $k$-fold cross validation accuracy and store it
 9: Remove $\boldsymbol{\phi}_j$ from $D$
10: for $s = 2$ to $m_{\max}$ do
11:     for all $\boldsymbol{\phi}_i$ in $D$ do
12:         Orthonormalize $\boldsymbol{\phi}_i$ with respect to $[\mathbf{q}_1, \ldots, \mathbf{q}_{s-1}]$ to obtain $\mathbf{w}_i$
13:         if $\mathbf{w}_i^T \mathbf{w}_i < 10^{-10}$ then
14:             Remove $\boldsymbol{\phi}_j$ from $D$
15:             Go to next iteration
16:         Compute $r^{(i)}(\mathbf{w}_i, \mathbf{y})$
17:     Find $j = \underset{1\leq i\leq M-s+1}{max} \left\{r^{(i)}(\mathbf{w}_i, \mathbf{y})\right\}$
18:     Define $\mathbf{q}_s = \mathbf{w}_j$
19:     Define $\mathbf{p}_s = \boldsymbol{\phi}_j$
20:     Train a logistic regression model using $\mathbf{y}$ and $\mathbf{p}_1, \ldots, \mathbf{p}_s$
21:     Compute the $k$-fold cross validation accuracy and store it
22:     Remove $\boldsymbol{\phi}_j$ from $D$
23: Using the stored $k$-fold cross validation accuracies, select the most parsimonious
    model with $m \leq m_{max}$ terms with the best accuracy performance
24: Return matrix of terms selected $\boldsymbol{\Phi} = \left[\begin{array}{cccc} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \ldots & \boldsymbol{\phi}_m \end{array}\right]$ and vector of coef-
    ficients $\boldsymbol{\theta} = \left[\begin{array}{cccc} \theta_1 & \theta_2 & \ldots & \theta_m \end{array}\right]^T$
```

a simple logistic model using the maximum likelihood estimation method, and assess its performance using a k-fold CV accuracy. New candidate terms are orthogonalised with respect to the model terms already chosen using the modified Gram-Schmidt method, and evaluated using the biserial correlation coefficient. This process is repeated in lines 10-22 until it reaches a pre-specified maximum number m_{max} of model terms to be included in the final model, where $m \leq m_{max}$. Lines 13-15 are used to calculate the squared norm-2 of each candidate model term, based on which it decides if a candidate term should be excluded to avoid any potential ill-conditional issue. When a new model term is included, a logistic regression model is trained and the k-fold CV accuracy is computed in lines 20-21. When the iteration reaches the specified number m_{max}, a parsimonious model consisting of a total of up to m_{max} model terms is then selected in line 23 based on the best CV accuracy obtained. Finally, the algorithm returns the parameters $\boldsymbol{\theta}$ together with the selected model terms. Given that the optimal number of model terms is not known in advance, the parameter m_{max} can be selected heuristically, by running Algorithm 6.1 several times, and checking the resulted CV accuracy curve. In case the best model contains exactly m_{max} model terms, this means that the appropriate number of model terms may be beyond this value; therefore it could be increased to find a better model.

The proposed algorithm combines the transparency and efficiency of the NARX models with logistic regression to deal with classification problems. This combination is advantageous since the NARX methodology helps to deal with the multicollinearity problem because of the orthogonalisation process that takes places. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models, something that is not achievable using random forests, support vector machines and k-nearest neighbors.

The time complexity of the logistic NARX method is determined by three main parts: the assessment of feature relevancy to the class label, the computation of the logistic regression model, and the orthogonalisation operations. Feature relevancy assessment has a linear time complexity of $O\left(NM\right)$, where N is the number of observations and M is the number of candidate features. The computation of the regression model has a worst-case time complexity of $O\left(M^3 + NM\right)$ [147], while the orthogonalisation procedure has a complexity of $O\left(N\left(M-1\right)\right)$ [148]. As a result, the overall time complexity takes the order of $O\left(M^3 + NM\right)$.

6.3 Case studies

In this section, three simulation examples are provided to illustrate the effectiveness of the new Logistic NARX methodology. In the first two cases, data is created from a lagged polynomial model, while in the third case a lagged non-polynomial model is used. In both cases, the performance of the algorithm with traditional classification techniques is compared. For simplicity, the analysis is restricted to polynomial NARX models as described in Eq. (2.2), although the algorithm can be applied to other NARX models using wavelets [78, 149] or radial basis functions [150, 151]. Furthermore, two real scenarios are presented where the methodology is applied to the detection of cancerous cells in a breast cancer data set [152–154], and the detection of human eye blinking using an electroencephalogram data set [154].

6.3.1 Example 1

Consider the following input-output system:

$$y\left[k\right] = \begin{cases} 1 & \text{if } u^2\left(k\right) + 2v^2\left(k\right) - 0.8u^2\left(k\right)v\left(k\right) + e\left(k\right) < 1 \\ \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

where the inputs $u\left[k\right]$ and $v\left[k\right]$ are uniformly distributed between $\left[-1, 1\right]$, i.e. $u\left[k\right]$,$v\left[k\right] \sim \mathcal{U}\left(-1, 1\right)$, and $e\left[k\right] \sim \mathcal{N}\left(0, 0.3^2\right)$. A total of 1000 input-output data points are collected. Plotting such points produces the figure shown in Figure 6.1.

Most classification techniques are able to perform static binary classification with high accuracy. The new algorithm is applied to this data set. The data is separated in a training set (700 points) and a testing set (300 points). Given that this is a static problem, no lags are used, and the nonlinear degree is chosen as $\ell = 3$, which results in a search space with 10 model terms. Therefore, the maximum number of terms is selected as $m_{max} = 10$, and 10 folds are used to compute the CV accuracy.

Fig. 6.2 shows the CV accuracy plot obtained after applying Algorithm 6.1 and it suggests that no significant improvement in accuracy is obtained with models that have more than 4 models terms. Therefore, a model with 4 terms is chosen and these are shown in Table 6.1. Such results show that the algorithm is able to identify correctly all model terms involved in the decision boundary for Eq. (6.4). The parameters obtained are log odds ratios, therefore they do not necessarily need

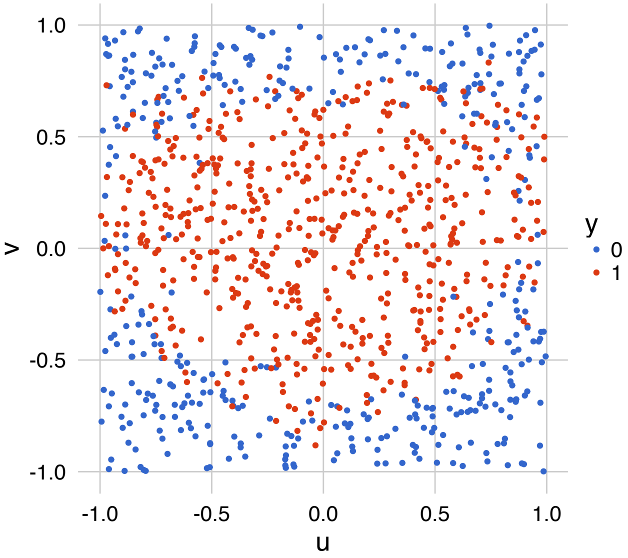

Figure 6.1: Data points obtained from the input-output system given in Eq. (6.4). The variables u and v are uniformly distributed random variables. Points in blue correspond to class $y = 0$ and points in red correspond to class $y = 1$. It is possible to find a model that separates the two classes with high accuracy.

to resemble the ones in the decision boundary function.

For comparison purposes, a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with

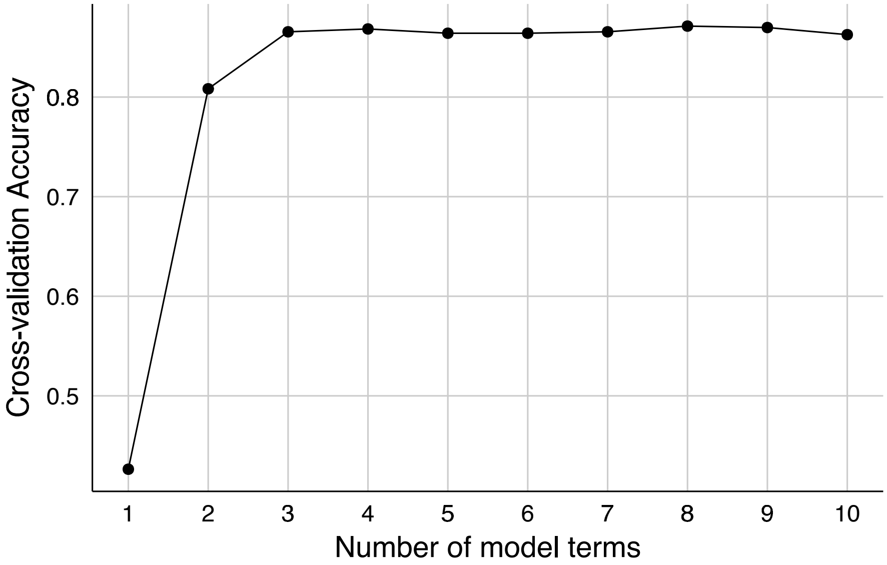

Figure 6.2: Cross-validation accuracy plot obtained for Eq. (6.4) using Algorithm 6.1. No significant improvement in accuracy is obtained with models that have more than 4 models terms.

Model Term	**Parameter**
$v^2(k)$	-12.297
constant	6.459
$u^2(k)$	-6.632
$u^2(k)\, v(k)$	4.470

Table 6.1: Identified logistic NARX model for Eq. (6.4) using Algorithm 6.1. Four model terms were identified with their corresponding estimated parameter values. The parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function. All model terms selected correspond to the true model.

Method	**Classification accuracy**
Logistic NARX	**0.8829**
Regression NARX	0.8763
Random Forest	0.8729
Support Vector Machine	0.8796
K-Nearest Neighbors	0.8428

Table 6.2: Accuracy performance between different methods for modelling of Eq. (6.4). The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The logistic NARX model has a comparable performance with the rest of the techniques.

a radial basis kernel, and a k-nearest neighbors model are trained with the same training set. All models are compared using the testing set and the classification accuracy. The results are shown in Table 6.2. It can be seen that the new method has a comparable performance with the rest of the techniques, making it a feasible alternative for static binary classification problems.

6.3.2 Example 2

Let us consider a slightly different version of Eq. (6.4) as follows:

$$y[k] = \begin{cases} 1 & \text{if } u^2(k-1) + 2v^2(k-2) - 0.8u^2(k-2)\, v(k-1) + e(k) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

Plotting again the data points results in the figure shown in Figure 6.3. As it can be observed, there is not a clear boundary between the two classes as in Figure 6.1. This is a problem as it can be wrongly suggested that the two classes cannot be separated.

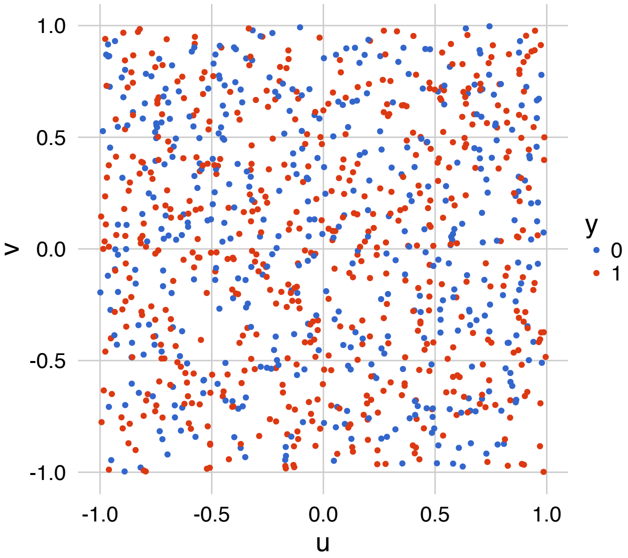

Figure 6.3: Data points obtained from the input-output system given in Eq. (6.5). The variables u and v are uniformly distributed random variables. Points in blue correspond to class $y = 0$ and points in red correspond to class $y = 1$. Although the points can be separated by means of Eq. (6.5), there is not a clear boundary between the two classes.

The new algorithm is applied to this data set. The data is separated in a training set (the first 700 points) and a testing set (the last 300 points). The maximum lags for the inputs and output are chosen to be $n_u = n_y = 4$, and the nonlinear degree is $\ell = 3$, which results in a search space with 165 model terms. The maximum number of terms is selected as $m_{max} = 10$, and 10 folds are used to compute the CV accuracy. Figure 6.4 shows the CV accuracy plot obtained after applying Algorithm 6.1 and it suggests that the most parsimonious model with the best accuracy has 4 models terms. These are shown in Table 6.3. Such results show that the algorithm is able to identify correctly all model terms involved in the decision boundary for Eq. (6.5). The parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function.

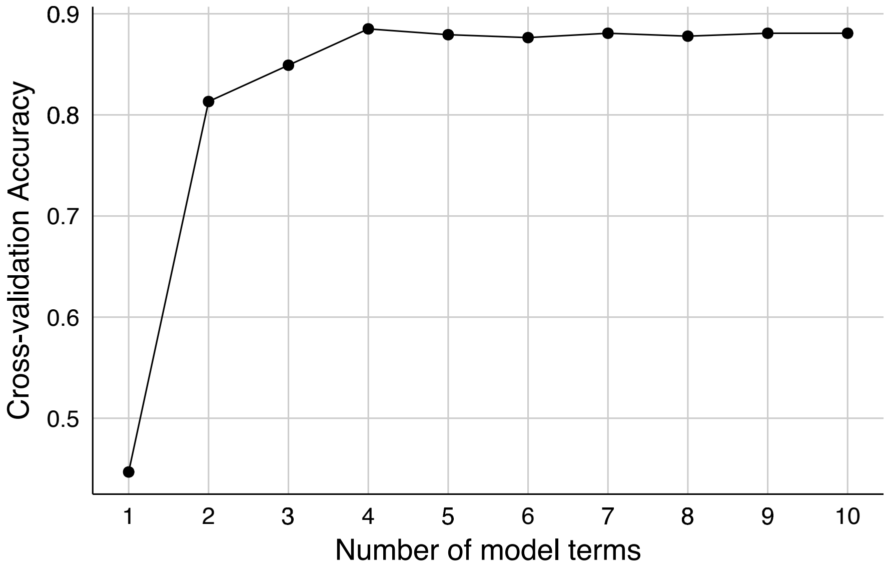

Figure 6.4: Cross-validation accuracy plot obtained for Eq. (6.5) using Algorithm 6.1. No significant improvement in accuracy is obtained with models that have more than 4 models terms.

Model Term	Parameter
$v^2(k-2)$	-12.508
constant	6.155
$u^2(k-1)$	-6.086
$u^2(k-2)\,v(k-1)$	4.582

Table 6.3: Identified logistic NARX model for Eq. (6.5) using Algorithm 6.1. Four model terms were identified with their corresponding estimated parameter values. The parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function. All model terms selected correspond to the true model.

For comparison purposes, a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the same training set. In general, traditional classification techniques do not consider lagged variables unless these are explicitly included, therefore two cases are considered: the first case assumes that no autoregressive terms are available, therefore only $u(k)$ and $v(k)$ are used. In the second one, the same lagged input and output variables that are considered for the logistic NARX model are used with the maximum lags chosen to be $n_u = n_y = 4$ (the regression-like NARX model only considers the second case). All models are compared using the testing set and the OSA accuracy. The results

Method	**Classification accuracy**
Logistic NARX	**0.8581**
Regression NARX	**0.8581**
Random Forest (without autoregressive inputs)	0.5034
Support Vector Machine (without autoregressive inputs)	0.5574
K-Nearest Neighbors (without autoregressive inputs)	0.5267
Random Forest (with autoregressive inputs)	0.8514
Support Vector Machine (with autoregressive inputs)	0.777
K-Nearest Neighbors (with autoregressive inputs)	0.6284

Table 6.4: Accuracy performance between different methods for modelling of Eq. (6.5). The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The exclusion/inclusion of autoregressive inputs is also taken into account. The logistic NARX model has a comparable performance with the rest of the techniques. The NARX-type models have the best accuracy. This is expected given the NARX-like structure that generates the data.

are shown in Table 6.4. It can be seen that the new method has the best accuracy performance together with the regression-like NARX model. This is expected given the NARX-like structure that generates the data. Nevertheless, the regression-like NARX model produces real-valued outputs, which make them difficult to interpret for classification. On the other hand, the logistic NARX model is preferred because its outputs are restricted to range from 0 to 1, and they can be used as classification probabilities. Furthermore, the random forest, support vector machine and k-nearest neighbors models are not able to generate reliable results if lagged variables (i.e. values observed in some previous time instants) are not taken into account when defining the feature vector, however their performance is increased when the autoregressive input variables are included. Although it may be argued that the method is just slightly better than the random forest with autoregressive inputs, it must be taken into consideration that the logistic NARX model is transparent and the role or contribution of individual regressors can be known.

6.3.3 Example 3

Consider the following input-output system:

$$y\left[k\right]=\begin{cases}1 & \text{if } -u\left(k-1\right)\sqrt{\left|v\left(k-1\right)\right|}+0.5u^{3}\left(k-1\right)+\sin\left(v\left(k-2\right)\right)+e\left(k\right)<0\\ 0 & \text{otherwise}\end{cases} \tag{6.6}$$

where the inputs $u\left(k\right)$ and $v\left(k\right)$ are uniformly distributed between $[-1,1]$, i.e. $u\left(k\right),$ $v\left(k\right)\sim\mathcal{U}\left(-1,1\right)$, the error sequence is given by $e\left(k\right)=w\left(k\right)+0.3w\left(k-1\right)+0.6w\left(k-2\right)$ and $w\left(k\right)$ is normally distributed with zero mean and variance of 0.01, i.e. $w\left(k\right)\sim\mathcal{N}\left(0,0.1^{2}\right)$. A total of 1000 input-output data points are collected.

The new algorithm is applied to this data set. The data is separated in a training set (the first 700 points) and a testing set (the last 300 points). The maximum lags for the inputs and output are chosen to be $n_u=n_y=4$, and the nonlinear degree is $\ell=3$, which results in a search space with 165 model terms. The maximum number of terms is selected as $m_{max}=10$, and 10 folds are used to compute the CV accuracy. Figure 6.5 shows the CV accuracy plot obtained after applying Algorithm 6.1 and it suggests that the most parsimonious model with the best accuracy has 8 models terms. These are shown in Table 6.5.

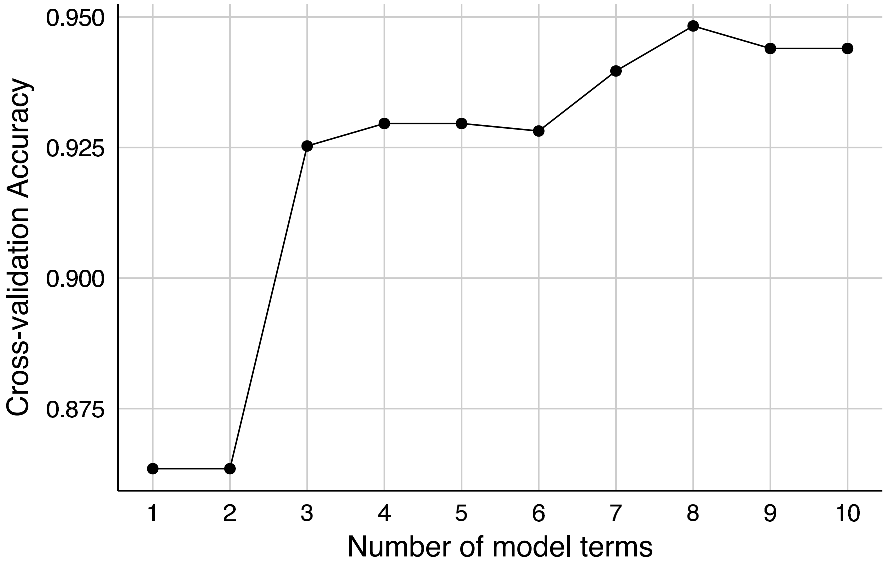

Figure 6.5: Cross-validation accuracy plot obtained for Eq. (6.6) using Algorithm 6.1. No significant improvement in accuracy is obtained with models that have more than 8 models terms.

Model Term	**Parameter**
$v\left(k-2\right)$	-12.755
constant	0.224
$u\left(k-1\right)v^2\left(k-1\right)$	8.488
$v^3\left(k-2\right)$	-15.323
$u\left(k-1\right)v^2\left(k-2\right)$	10.066
$u\left(k-1\right)$	9.047
$u^3\left(k-1\right)$	-8.715
$u\left(k-1\right)u^2\left(k-4\right)$	-3.285

Table 6.5: Identified logistic NARX model for Eq. (6.6) using Algorithm 6.1. Eight model terms were identified with their corresponding estimated parameter values. The parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function.

Similarly to the previous case study, a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the same training set. Again, two cases are considered: the first case assumes that no autoregressive terms are available, therefore only $u\left(k\right)$ and $v\left(k\right)$ are used. In the second one, the same lagged input and output variables that are considered for the logistic NARX model are used with the maximum lags chosen to be $n_u = n_y = 4$, and the nonlinear degree is $\ell = 3$, which results in a search space with 165 model terms. All models are compared using the testing set and the OSA accuracy. The results are shown in Table 6.6. It can be seen that the new method has the best accuracy performance, with a very similar result to the regression-like NARX model and the random forest with autoregressive inputs. Once more, the advantage over the random forest models is the transparency and interpretability about the role or contribution of individual regressors. Also, the advantage over the regression-like NARX model is a more interpretable output that is easily related to a classification probability.

Method	Classification accuracy
Logistic NARX	**0.9392**
Regression NARX	0.9358
Random Forest (without autoregressive inputs)	0.527
Support Vector Machine (without autoregressive inputs)	0.4932
K-Nearest Neighbors (without autoregressive inputs)	0.47
Random Forest (with autoregressive inputs)	0.9223
Support Vector Machine (with autoregressive inputs)	0.8986
K-Nearest Neighbors (with autoregressive inputs)	0.7973

Table 6.6: Accuracy performance between different methods for modelling of Eq. (6.6). The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The exclusion/inclusion of autoregressive inputs is also taken into account. The logistic NARX model has a slightly better performance than the regression-like NARX model and the random forest with autoregressive inputs. However, the logistic NARX model provides transparency and interpretability about the contribution of individual regressors together with a classification probability for the predictions.

6.3.4 Breast Cancer Classification

Breast cancer is the most common cancer in women worldwide [153]. Among the different prevention and control techniques, early detection is still the best method in order to improve breast cancer outcome and survival [155]. For this case study, it is employed the breast cancer data set from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [154]. This data set contains 699 instances with the following 10 attributes:

- ID number
- Clump thickness (integer value between 1 and 10)
- Uniformity of cell size (integer value between 1 and 10)
- Uniformity of cell shape (integer value between 1 and 10)
- Marginal adhesion (integer value between 1 and 10)
- Single epithelial cell size (integer value between 1 and 10)
- Bare nuclei (integer value between 1 and 10)

- Bland chromatin (integer value between 1 and 10)
- Normal nucleoli (integer value between 1 and 10)
- Mitoses (integer value between 1 and 10)
- Class (2 for benign, 4 for malignant)

The bare nuclei attribute contains 16 missing values. Such instances are removed from the analysis. Also, the ID number attribute does not provide any meaningful information for the classification task, so it is removed from the data set. The class attribute is recoded with ’0’ for a benign case and ’1’ for a malignant. The rest of the attributes are divided by 10 in order to have feature values ranging from 0.1 to 1.

The data is separated in a training set (400 instances with 200 samples from each class) and a testing set (283 instances). The frequency of the class for each set is shown in Figure 6.6 where it can be noticed that each cancer type has the same frequency in the training set, however, this is not the case in the testing set.

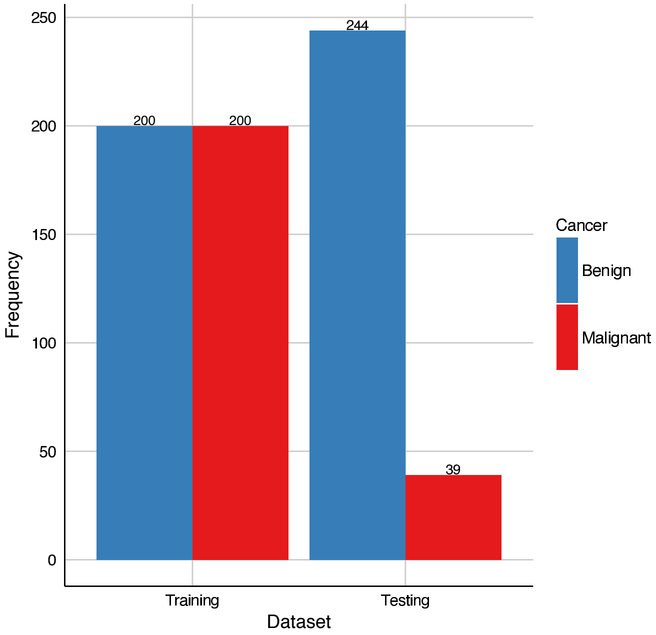

Figure 6.6: Frequency of each cancer type for the training and testing sets. Each cancer type has the same frequency in the training set to facilitate the identification of the two classes. The imbalanced testing set can be used to check the performance of the trained model.

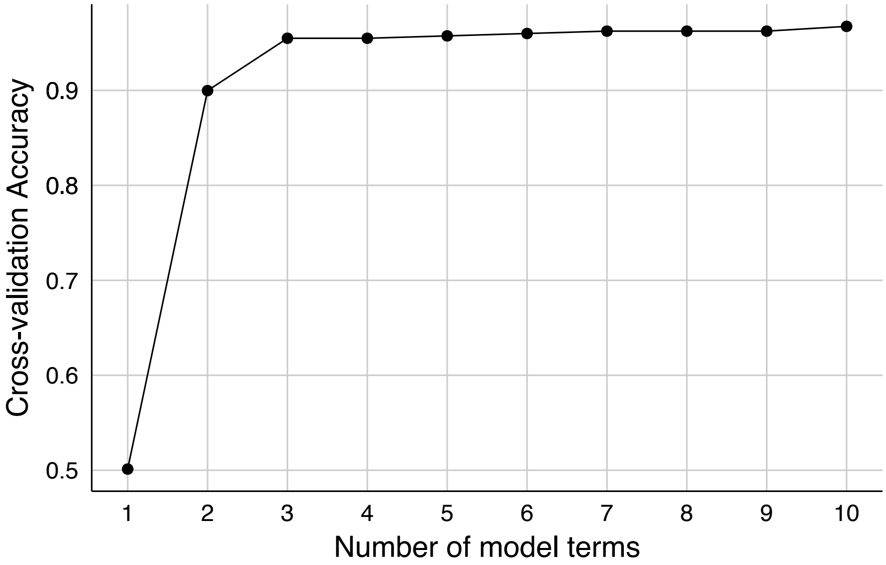

Figure 6.7: Cross-validation accuracy plot obtained for the Breast Cancer data set using Algorithm 6.1. No significant improvement in accuracy is obtained with models that have more than 3 models terms.

Nevertheless, this is not a significant issue as the training phase has access to a good balance of the two classes that need to be identified, while the imbalanced testing set can be used to check the performance of the trained model.

Given that this is a static problem, no lags are used, and the nonlinear degree is chosen as $\ell = 2$ based on [46]. This results in a search space with 55 model terms. Therefore, the maximum number of terms to search is selected as $m_{max} = 10$, and 10 folds are used to compute the CV accuracy. Fig. 6.7 shows the CV accuracy plot obtained after applying Algorithm 6.1 and it suggests that no significant improvement is obtained in accuracy with models that have more than 3 models terms. Therefore, a model with 3 terms is chosen and these are shown in Table 6.7.

Model Term	**Parameter**
Bare nuclei	6.430
constant	-5.774
Uniformity of cell size	11.338

Table 6.7: Identified model terms for the Breast Cancer data set using Algorithm 6.1. Three model terms were identified with their corresponding estimated parameter values.

Once more, a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the same training set. All models are compared using the testing set and the classification accuracy. The results are shown in Table 6.8. All the methods are able to obtain a good classification accuracy. Although the logistic NARX does not have the best accuracy, the difference with the best ones is negligible. This makes the logistic NARX model a competitive alternative to other classification techniques.

Method	**Classification accuracy**
Logistic NARX	0.9716
Regression NARX	**0.9787**
Random Forest	**0.9787**
Support Vector Machine	0.9681
K-Nearest Neighbors	0.9716

Table 6.8: Accuracy performance between different methods for modelling of the Cancer Breast data set. The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The results suggest that the logistic NARX model is a competitive alternative to other classification techniques.

6.3.5 Electroencephalography Eye State Identification

Recently, electroencephalography (EEG) eye state classification has become a popular research topic with several applications in areas like stress features identification, epileptic seizure detection, human eye blinking detection, among others [156]. For this case study, the EEG Eye State data set found at the UCI Machine Learning Repository is used [154]. This data set contains 14,980 EEG measurements from 14 different sensors taken with the Emotiv EEG neuroheadset during 117 seconds (Figure 6.8). The eye state of the patient was detected with the aid of a camera during the experiment. If the eye is closed, it is coded as a '1', otherwise it is coded as '0'.

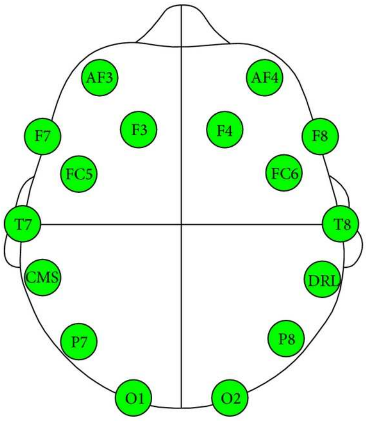

Figure 6.8: Emotiv EEG neuroheadset sensor position. A total of 14,980 EEG measurements were taken from each sensor during 117 seconds. Original source: [156].

For this analysis, the first 80% of the data set is used for training, while the rest is used for testing. The frequency of the eye state for each data set is shown in Figure 6.9. Similar to the breast cancer scenario, the two eye states have roughly the same frequency in the training set, however, this is not the case in the testing set. Once more, this is not a significant issue as the training phase can be performed

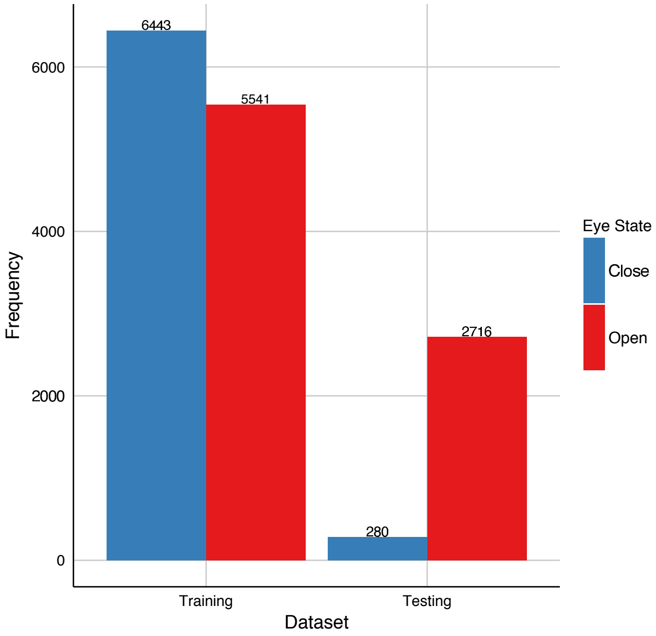

Figure 6.9: Frequency of the eye state for the training and testing sets. Each eye state has the roughly the same frequency in the training set to facilitate the identification of the two classes. The imbalanced testing set can be used to check the performance of the trained model.

with enough information from both eye states, while the imbalanced testing set can be used to check the performance of the trained model.

Furthermore, two preprocessing steps are performed in the training set. First of all, several outliers are detected using data visualisation techniques (i.e. boxplots, histograms and line plots) and summary statistics on the data of each of the 14 sensors (input variables). The outliers are replaced with the mean value of the remaining measurements for each variable. The eye state time series, together with the 14 cleaned variables, are shown in Figure 6.10. Second, an attempt to train a model using the original data set is done. However, given the high variability and dependency between the variables measured, the model does not perform well enough. Because of this, a principal component analysis (PCA) is performed in order to reduce the dimensionality of both the data and model space, and in this

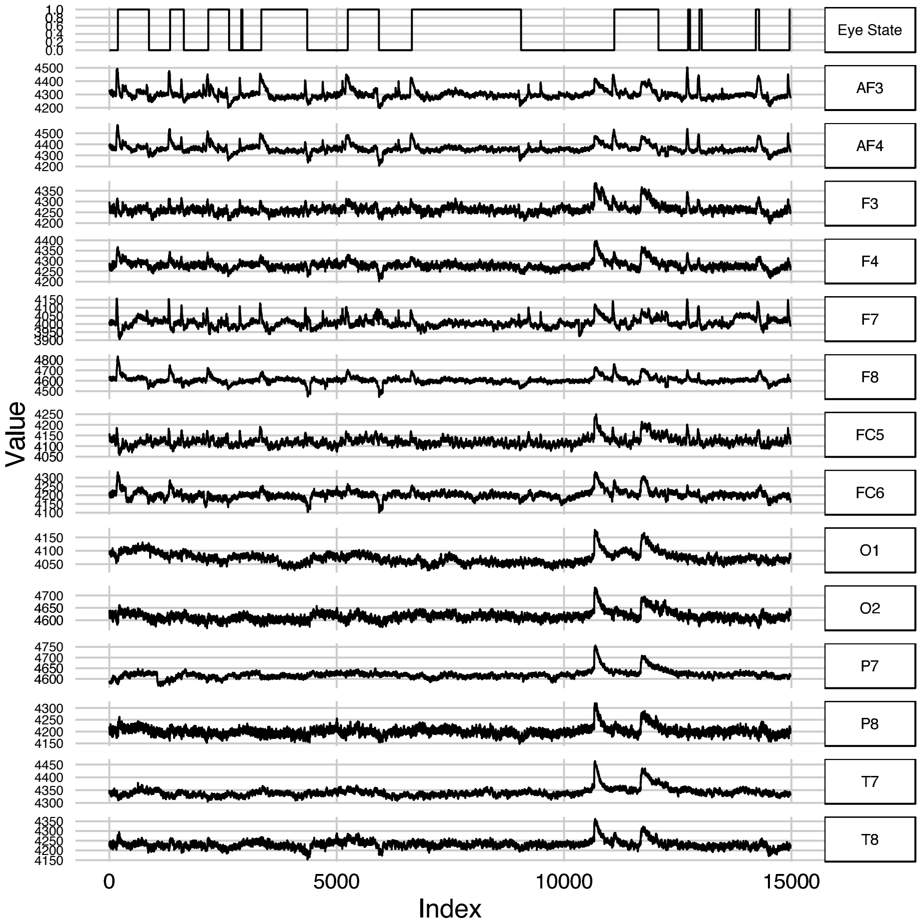

Figure 6.10: Time series of all variables in the EEG Eye State data set found at the UCI Machine Learning Repository [154].

analysis the 5 most important principal components (PCs) are used to represent the features of the original data. The PC time series are shown in Figure 6.11. Each PC is treated to be a new input variable; lagged PC variables are then used to built a logistic NARX model. For this analysis, the variables are transformed using scaling, centering and Box-Cox transformations. Therefore, the PCs summarise the main variability of the data set and simplify the identification process. The preprocessing parameters obtained during the training phases are directly used on the testing set in order to avoid the data snooping problem.

The logistic NARX modelling approach is applied to this data set. The output variable is the eye state signal, and the input variables are the 5 PCs computed in the preprocessing phase. For this scenario, no lagged variables of the output signal are used in order to ensure that the model captures a pattern with the exogenous

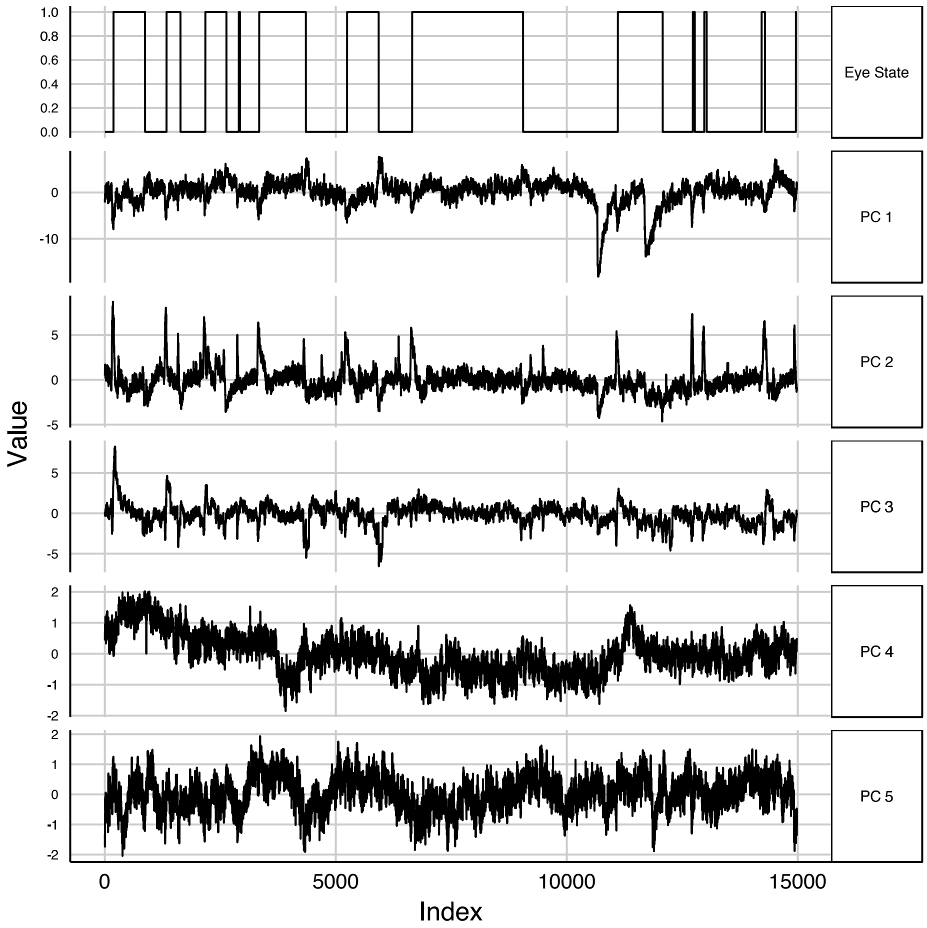

Figure 6.11: Time series of the 5 most important principal components of the EEG Eye State data set.

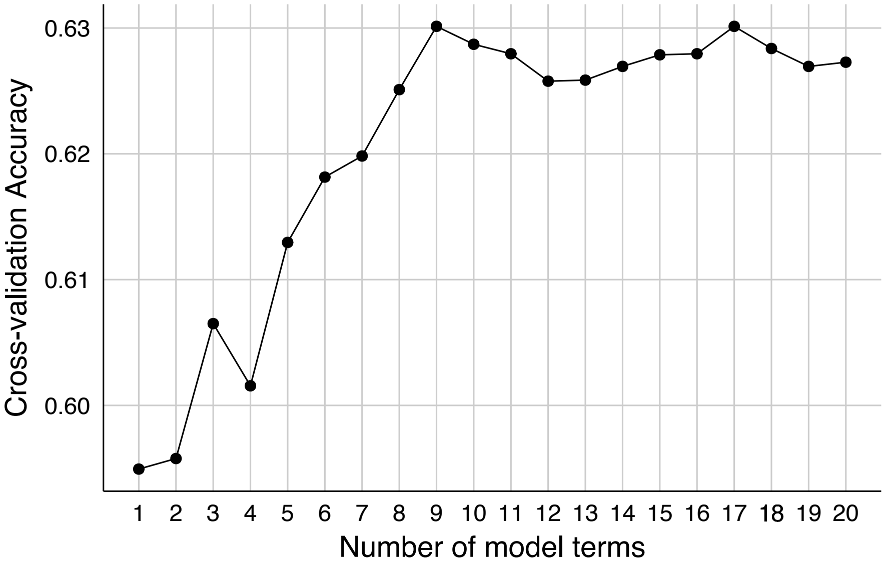

Figure 6.12: Cross-validation accuracy plot obtained for the EEG Eye State data set using Algorithm 6.1. No significant improvement in accuracy is obtained with models that have more than 9 models terms.

inputs only. The maximum lag for the inputs is chosen to be $n_u = 50$, and the nonlinear degree is $\ell = 1$ based on the results of previous works in [46, 156]. The search space is made up of 251 model terms. The maximum number of terms to look for is chosen as $m_{max} = 30$, and 10 folds are used to compute the CV accuracy. Fig. 6.12 shows the CV accuracy plot obtained after applying Algorithm 6.1 and it suggests that the most parsimonious model with the best accuracy has 9 models terms. These are shown in Table 6.9.

Model Term	**Parameter**
$PC_2\,(k-43)$	0.1545
constant	0.2123
$PC_3\,(k-50)$	0.5776
$PC_1\,(k-43)$	-0.1384
$PC_2\,(k-1)$	-0.2593
$PC_2\,(k-38)$	0.1766
$PC_2\,(k-50)$	0.3606
$PC_3\,(k-1)$	-0.1214
$PC_2\,(k-32)$	0.1536

Table 6.9: Identified model terms for the EEG Eye State data set using Algorithm 6.1. Nine model terms were identified with their corresponding estimated parameter values.

In order to assess the performance of the resultant logistic NARX model, a

Method	Classification accuracy
Logistic NARX	**0.7199**
Regression NARX	0.6643
Random Forest (without autoregressive inputs)	0.5475
Support Vector Machine (without autoregressive inputs)	0.6029
K-Nearest Neighbors (without autoregressive inputs)	0.5041
Random Forest (with autoregressive inputs)	0.6365
Support Vector Machine (with autoregressive inputs)	0.6473
K-Nearest Neighbors (with autoregressive inputs)	0.5662

Table 6.10: Accuracy performance between different methods for modelling of the EEG Eye State data set. The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The exclusion/inclusion of autoregressive inputs is also taken into account. The logistic NARX model has the best accuracy performance and identifies the most significant lagged PCs that contribute to the classification of the eye state.

regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the same training set. Similar to the previous examples, two cases are considered. One where the current input values are used, i.e. without lags, and another where the same lagged variables that are used for the logistic NARX model are employed. All models are compared using the testing set and the OSA accuracy. The results are shown in Table 6.10. In this case, the new method has the best accuracy performance and identifies the most significant lagged PCs that contribute to the classification of the eye state. The models that are trained without autoregressive inputs have a poor classification accuracy. This is improved when autoregressive information is included. However, they do not achieve a classification accuracy like the one obtained by the logistic NARX model.

6.4 Discussion

The proposed logistic NARX algorithm shows a new approach to deal with classification problems. The new method has a similar performance with other classification techniques when dealing with static data, but it outperforms other methods when there is a dynamic component, and lagged versions of input and output variables are required. The algorithm is able to produce interpretable models where the contribution of each model term can be analysed. In comparison with random forests, support vector machines and k-nearest neighbors approaches, the new method can generate better or comparative performance as illustrated in the case studies. Additionally, when using random forests, it is possible to get the variable importance, which may provide some insight about which variables are contributing the most to explain the output. However, this only ranks the variables, and does not explain how the variables are contributing to the modelling process. The logistic NARX model overcomes this by providing variable importance and interpretability about how the variables are interacting.

Nevertheless, there are some limitations to the proposed algorithm. First of all, this work focuses on polynomial-like structures, therefore, severe nonlinearities may not be modelled properly. To overcome this, other structures can be considered (e.g. radial basis functions, wavelets), and this will be considered in a future extension of this research. Another issue, is the selection of the maximum lags for the output and input sequences (n_y and n_u). This is an open research problem where several interesting approaches have been proposed to tackle it [46,50]. It would become more difficult when the lags become large, as the model search space has a factorial growth which makes it intractable. Also, the logistic NARX approach may be affected by severe correlation between the inputs, which results in poor performance models. Some alternatives to overcome this include the iterative OFR [85] and the ultra OFR [86]. Finally, the performance of the logistic NARX model can be affected if the data are not balanced (especially when the output data are imbalanced). The scenario of imbalanced data is typical in many real applications where the minority class is dominated or buried by the majority class. Several approaches are available for dealing with imbalanced data problem, readers are referred to [17,157] for details.

6.5 Summary

In this chapter a novel algorithm that combines logistic regression with the NARX methodology is developed. This allows to tackle classification problems where the output signal is a binary sequence and the regressors are continuous lagged variables. The new approach can deal with the multicollinearity problem while producing models that predicts binary outcomes. From the five case studies, the performance of the proposed logistic NARX models is preferable to that of the other compared methods when dealing with binary-label prediction, where it is sometimes highly desirable to know which input variables play an important role individually and/or interactively in the classification process. The results obtained are promising, and future research may extend this method to multi-class problems. The results of this chapter were published in [29].

Chapter 7

Modelling Global Magnetic Disturbances in Near-Earth Space

7.1 Introduction

The operation of many modern technological systems is vulnerable to space weather disturbances. Severe geomagnetic disturbances, such as magnetic storms, can have serious adverse effects on power grids, navigation systems and affect satellite drag. Forecasts of space weather hazards can assist reliable operation of these technological systems. However, a physical model of the solar-terrestrial system that can be used to forecast the evolution of the magnetosphere has not been developed yet, because of the complexity of the dynamical processes involved.

The Kp index is one of the most widely used indices for quantifying geomagnetic activity. It stands for *planetarische Kennziffer,* which means planetary index in German. In [26], it was concluded that the Kp index is a good measure of the strength of magnetospheric convection because of its dependence on the latitude of the auroral current region. This index is computed by taking the weighted average of K indices at 13 ground magnetic field observatories. The values of Kp range from 0 (very quiet) to 9 (very disturbed) in 28 discrete steps, resulting in values of 0, 0+, 1-, 1, 1+, 2-, 2, 2+, ..., 9 [158].

The Kp index is known to be correlated with solar wind observations [159, 160]. This has enabled the development of models that attempt to forecast Kp. The most popular models are based on artificial neural networks, which are considered black-box models [161, 162]. For instance, in [158], three neural networks were trained with solar wind data and are now used to nowcast the Kp index, producing hourly and 4-hourly forecasts of the Kp, updated every 15 minutes. In [163], an improved neural network was trained using the Boyle index in order to generate 1-, 3- and 6-hour ahead predictions. The Liu Kp model consists of a neural network trained with autoregressive values of Kp and solar wind data, and is able to predict Kp values up to 3.5 hours in advance [164]. A comparative study between neural networks and support vector machines was done in [165]. These authors found that the best model is a neural network trained with the same inputs as the Liu Kp model. A probabilistic approach was taken in [166] where the Kp range is divided in 4 groups and 1268 models were compared in terms of accuracy, reliability, discrimination capability, and forecast skill.

In general, there are two approaches for the modelling of magnetic disturbances: first-principles modelling and data-based modelling. The latter has been previously used to model space weather phenomena. For example, the NARX approach was used to model the evolution of energetic electrons fluxes at geostationary orbit [167], to obtain the most influential coupling functions that affect the evolution of the magnetosphere [168], to predict the Dst index using multiresolution wavelet models [169], and to build a multiscale radial basis function network to forecast the geomagnetic activity of the Dst index [150], among others. Furthermore, NARX models can be used to compute the generalised frequency response functions (GFRFs) in order to perform frequency domain analysis [5]. This technique has been used previously to study the spectral properties of the Dst index dynamics [170] and to identify types of nonlinearities involved in the energy storage process in the magnetosphere [171].

In this chapter, the use of NARX models to forecast the Kp index is investigated. In particular, there is interest in forecasts at four different horizons: 3, 6, 12 and

24 hours ahead. To do so, two approaches are explored. The first one consists of a recursive sliding window scheme in which a window period of 6 months is employed to train a model and used to forecast future values based on previous predictions. The second approach involves the identification of a specific model for each horizon of interest using a fixed data set of 6 months. In addition, given that the output variable is categorical, the use of the logistic NARX approach, described in Chapter 6, is explored.

7.2 Data set description

Every three hours throughout the day, 13 ground-based magnetic field observatories located at geomagnetic latitudes between 48º and 63º around the world, record the largest magnetic change that their instruments measure. This change is denoted as the K index, which is given on a quasi-logarithmic scale from 0 ($< 5\,nT$) to 9 ($> 500\,nT$) [162]. The average of these observations is known as the Kp index. This determines how disturbed the Earth's magnetosphere is on a scale that goes from 0 (very quiet) to 9 (very disturbed) in 28 discrete steps, resulting in values of 0, 0+, 1-, 1, 1+, 2-, 2, 2+, ..., 9 [158, 162]. In this analysis, these values are rescaled to be represented by the numbers 0, 0.3, 0.7, 1, ..., 9. In general, large Kp values can indicate a more active terrestrial magnetosphere due to a solar storm, or a sudden rearrangement of the Earth's magnetosphere due to the solar wind [26, 158].

The data sets that are used consist of the variables shown in Table 7.1, which have been selected and gathered by researchers in the Department of Automatic Control and Systems Engineering at the University of Sheffield. These variables were measured during the year 2000. The inputs are taken from the low resolution OMNI data set, which consist of hourly average near-Earth solar wind magnetic field and plasma data from several spacecraft in geocentric or L1 (Lagrange point) orbits. The data period used for this study employed four spacecrafts: IMP 8, WIND, Geotail and ACE. The output is the Kp index which, as mentioned before, is measured every three hours. In order to match the time resolutions between

Variable	Symbol	Description
Input	V	Solar wind speed [km/s]
	Bs	Southward interplanetary magnetic field [nT]
	VBs	Southward interplanetary magnetic field [$VBs = V \cdot Bs/1000$]
	p	Solar wind pressure [nPa]
	$\sqrt{p}$	Square root of solar wind pressure
Output	Kp	Kp index (variable of interest)

Table 7.1: Data set variables for modelling global magnetic disturbances in near-Earth space. The inclusion of the $\sqrt{p}$ variable is to allow the algorithm to identify fractional exponents of the solar wind pressure p.

the input and output signals, the observed Kp values are interpolated to 1-hour resolution by simply repeating the last measured value during the next two hours.

Given that the variable of interest is the Kp index, its distribution for year 2000 is shown in Figure 7.1. This highlights that high values of Kp are rare, which makes their prediction a challenging task.

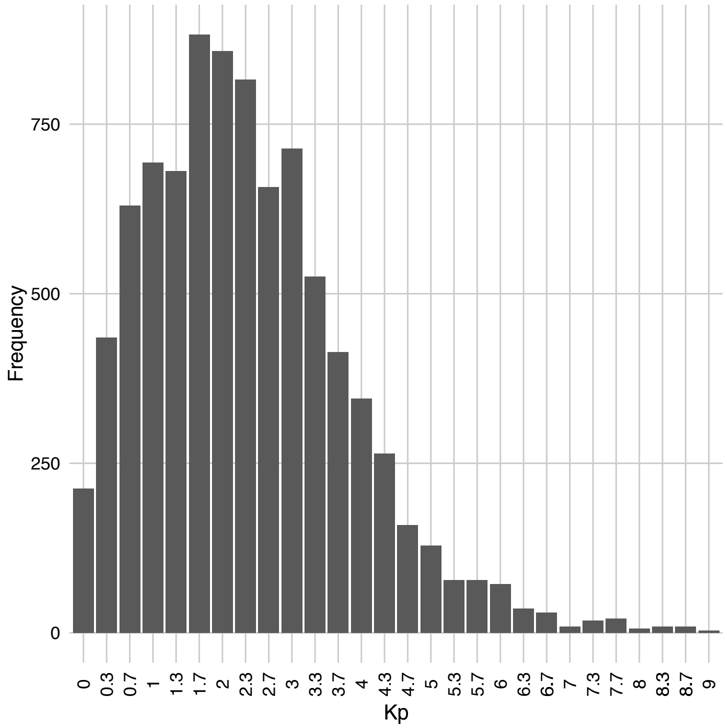

Figure 7.1: Histogram of the Kp index for year 2000. High Kp values of 5 to 9 are relatively rare.

7.3 Recursive approach

This approach uses a window of a fixed length to build a single model using the data within the window frame as the training set. The remaining future data outside the window is used as validation set to make predictions 3, 6, 12 and 24 hours ahead based on the simulated values (MPOs) as shown in Eq. (2.19). Once this is done, the window is moved forward by one time step, a new model is built and subsequently used to forecast the next Kp values at 3, 6, 12 and 24 hours ahead. This way, the training and validation sets are mutually exclusive.

Every time the window frame moves forward, a new NARX model is trained. The training process uses the adaptive orthogonal search algorithm described in [47] together with the PRESS metric. A nonlinear model term and variable selection procedure proposed in [46] is applied, and numerical experimental results suggest that $n_y = 4$ and $n_u = 2$ is an appropriate choice. Accordingly, the NARX model structure is given by

$$\begin{aligned}\widehat{Kp}\left(k\right) =f\Big(&Kp\left(k-1\right),\ldots,Kp\left(k-4\right),\\ &V\left(k-1\right),V\left(k-2\right),Bs\left(k-1\right),Bs\left(k-2\right),\\ &VBs\left(k-1\right),VBs\left(k-2\right),p\left(k-1\right),p\left(k-2\right),\\ &\sqrt{p\left(k-1\right)},\sqrt{p\left(k-2\right)}\Big) \end{aligned} \tag{7.1}$$

where $f\left(\cdot\right)$ is chosen to be a polynomial of nonlinear degree $\ell = 2$, $Kp\left(k\right)$ is the measured Kp index at time k, and $\widehat{Kp}\left(k\right)$ is the predicted Kp index at time k. The window length is of 6 months, therefore the initial training and validation sets correspond to the first and second half of year 2000, respectively. As the window frame moves forward, the validation set size decreases. The reason to choose a window length of 6 months is because for the NARX methodology typically just a few hundred data samples are required to estimate a model, which can be important in many applications where it is unrealistic to perform long experiments [5].

The results for this approach are shown in Figure 7.2. Here it can be seen that there is a bias for low and high magnetic disturbances. Furthermore, for high values of Kp ($Kp \geq 8$) the error bars become odd and difficult to interprete. This is due to the fact that there are very few occurrences of high-value Kp indexes, so few predictions are made in such cases and hence they tend to be underpredicted. Such characteristics have been previously reported in [161, 162], where it is argued that a model will perform well for the most common training values, while predictions for rare values will be poor.

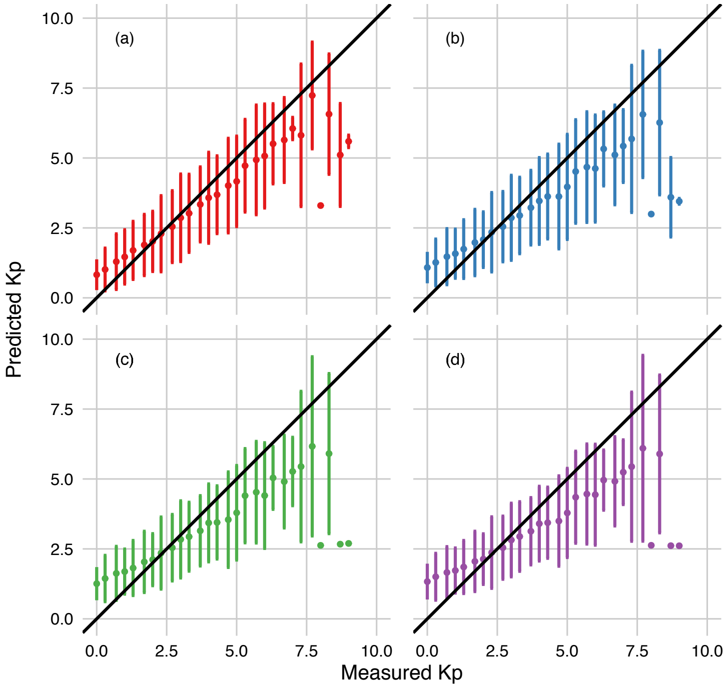

Figure 7.2: Comparison between the measured Kp index and predictions made for (a) 3, (b) 6, (c) 12, and (d) 24 hours ahead using the sliding window approach. The black line represents the ideal case when the prediction is equal to the measured Kp index. The points and bars correspond to the means and one standard deviations of the predictions made for each of the 28 Kp values.

To quantify the results, the root mean squared error (RMSE), correlation coefficient (ρ) and prediction efficiency (PE) are computed. The latter is defined as

$$PE = 1 - \frac{\sigma^2_{error}}{\sigma^2_{measured}} \tag{7.2}$$

where $\sigma^2_{measured}$ is the variance of the measured Kp values, and σ^2_{error} is the variance of the error between the measured Kp values and the predicted ones. These metrics are shown in Table 7.2.

Horizon	RMSE	ρ	PE
3	0.7935	0.8590	0.7359
6	0.9014	0.8159	0.6598
12	0.9513	0.7991	0.6225
24	0.9624	0.7972	0.6149

Table 7.2: Performance metrics for each of the four horizons of interest obtained with the sliding window approach.

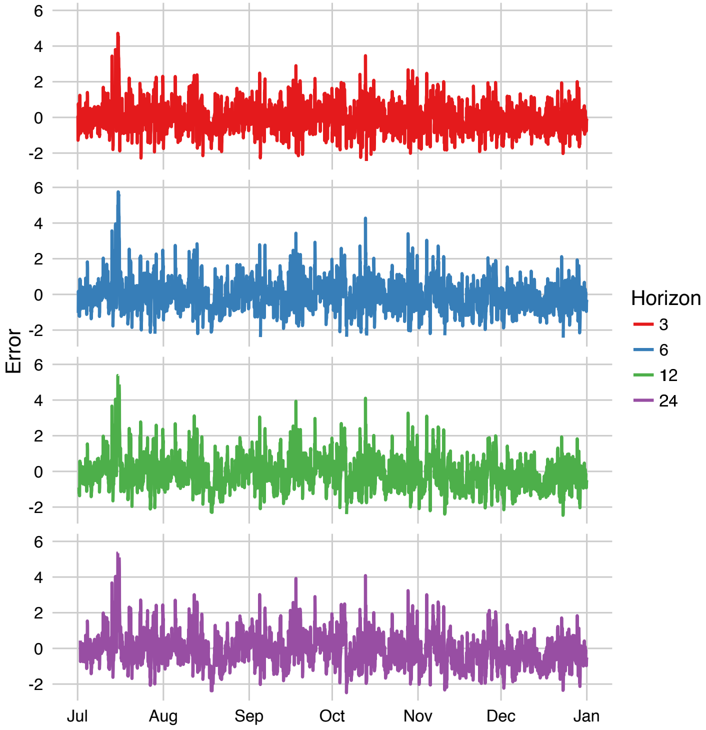

Figure 7.3: Error time series for the four horizons of interest obtained with the sliding window approach.

The error time series for each of the four horizons of interest are shown in Figure 7.3. It can be seen that the error is notoriously high at the middle of July. Figure 7.4 shows a glimpse of this period where it can be seen that high activity of the terrestrial magnetosphere was recorded between July 13th-17th, 2000. Such an activity is not properly forecasted by this approach. In addition, Table 7.3 shows a statistical summary of the error time series in Figure 7.3. In general, it can be concluded that this approach tends to overpredict the Kp index given that both the median and the mean are negative. Furthermore, as the number of hours to predict ahead increases, the forecasts are less accurate because the interquartile range (1st quartile - 3rd quartile) increases.

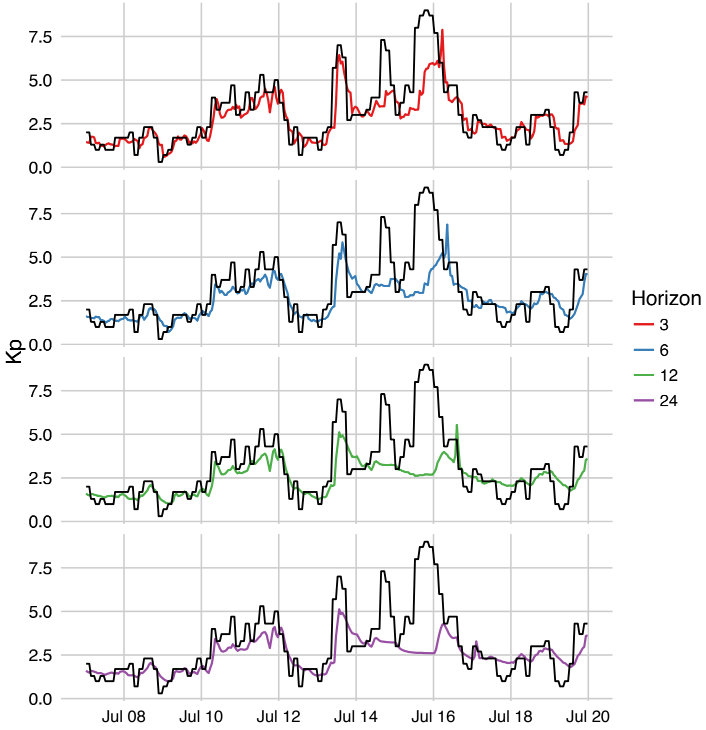

Figure 7.4: Predictions of the *Kp* index for the four horizons of interest during the middle of July 2000 using the sliding window approach. The black line corresponds to the measured *Kp* values.

Statistic	Forecast			
	3	6	12	24
Minimum	-3.1980	-2.7180	-2.5570	-2.4860
1st Quartile	-0.5394	-0.6409	-0.7106	-0.7270
Median	-0.0843	-0.1032	-0.1391	-0.1497
Mean	-0.0303	-0.0491	-0.0711	-0.0830
3rd Quartile	0.4084	0.4440	0.4454	0.4447
Maximum	4.7170	5.7520	6.3050	6.3900

Table 7.3: Statistical summary for the error time series shown in Figure 7.3.

7.4 Direct approach

The second modelling technique investigated involves use of what is termed the direct approach. Instead of training a model many times and using it recursively to calculate forecasts, the direct approach obtains a separate model for a horizon h of interest. In such a case, Eq. (2.1) becomes

$$\begin{aligned} y\left(k\right) = & f\Big(y\left(k-h\right), y\left(k-h-1\right), \ldots, y\left(k-h-n_y+1\right), \\ & u\left(k-1\right), u\left(k-2\right), \ldots, u\left(k-n_u\right)\Big) + e\left(k\right) \end{aligned}$$

The main advantage of the direct approach is that it only requires the computation of h-step ahead predictions. This means that the output at the present time k, $y\left(k\right)$, is predicted using the past values $y\left(k-h\right), y\left(k-h-1\right), \ldots, y\left(k-h-n_y+1\right)$, $u\left(k-1\right), u\left(k-2\right), \ldots, u\left(k-n_u\right)$, where it is assumed that these are known [150]. As mentioned in section 2.4.3, the h-step ahead prediction is defined with respect to the system output; it is actually still one-step ahead prediction with respect to the system input.

In similarity to the sliding window approach, $n_y = 4$ and $n_u = 2$ are chosen, and the training process uses the adaptive orthogonal search algorithm described in [47] together with the PRESS metric. Accordingly, the NARX model structure is given by

$$\begin{aligned} \widehat{Kp}\left(k\right) = & f\Big(Kp\left(k-h\right), \ldots, Kp\left(k-h-3\right), \\ & V\left(k-1\right), V\left(k-2\right), Bs\left(k-1\right), Bs\left(k-2\right), \\ & VBs\left(k-1\right), VBs\left(k-2\right), p\left(k-1\right), p\left(k-2\right), \\ & \sqrt{p\left(k-1\right)}, \sqrt{p\left(k-2\right)}\Big) \end{aligned} \tag{7.3}$$

where $f\left(\cdot\right)$ is chosen to be a polynomial of nonlinear degree $\ell = 2$, $Kp\left(k\right)$ is the measured Kp index at time k, and $\widehat{Kp}\left(k\right)$ is the predicted Kp index at time k. In this analysis, the first six months of year 2000 are used for training while the second half of the year is used for validation.

The models identified by the NARX methodology for each horizon are listed in Tables 7.4-7.7.

Model Term	**Parameter**
$Kp\,(k-3)$	0.325543
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	-0.000043
$\mathbf{Bs\,(k-1)}$	0.673034
$\mathbf{Bs\,(k-1)\,\sqrt{p\,(k-1)}}$	-0.164093
$V(k-1)^2$	-0.000003
$V(k-1)\cdot Bs(k-2)$	0.000217
$Bs(k-1)\cdot Bs(k-2)$	-0.006701
$Bs(k-1)\cdot p(k-2)$	-0.005810
constant	-2.179360
$\sqrt{p\,(k-1)}$	0.753122
$V(k-1)$	0.006105
$VBs(k-1)$	-0.387292
$VBs(k-1)\sqrt{p\,(k-1)}$	0.136271

Table 7.4: Identified NARX model for 3-hour ahead predictions of the Kp index. Thirteen model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	-0.000191
$\mathbf{Bs\,(k-1)}$	0.852464
$Kp\,(k-6)$	0.158716
$\mathbf{Bs\,(k-1)\,\sqrt{p\,(k-1)}}$	-0.172607
$V(k-1)\cdot Bs(k-2)$	0.000340
$V(k-1)^2$	-0.000003
$Bs(k-1)\cdot Bs(k-2)$	-0.058229
$Bs(k-1)\cdot p(k-2)$	-0.007989
$\sqrt{p\,(k-1)}\sqrt{p\,(k-2)}$	0.009495
$p\,(k-1)\cdot p\,(k-2)$	0.000962
constant	-2.749889
$V(k-1)$	0.007744
$\sqrt{p\,(k-1)}$	0.958020
$VBs\,(k-1)$	-0.514336
$VBs(k-1)\sqrt{p\,(k-1)}$	0.113874
$VBs(k-1)^2$	0.011219
$VBs(k-2)^2$	0.009277
$Bs\,(k-2)\cdot VBs\,(k-1)$	0.032255

Table 7.5: Identified NARX model for 6-hour ahead predictions of the Kp index. Eighteen model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\left(k-1\right)\sqrt{p\left(k-1\right)}}$	0.001618
$\mathbf{Bs\left(k-1\right)}$	0.748665
$\mathbf{Bs\left(k-1\right)\sqrt{p\left(k-1\right)}}$	-0.268901
$Kp\left(k-12\right)\cdot V(k-1)$	-0.000229
$Bs(k-2)$	0.203764
$Kp\left(k-12\right)\cdot p(k-1)$	-0.017656
$Bs(k-1)\cdot Bs(k-2)$	-0.007676
constant	-1.606480
$V(k-1)\cdot p\left(k-1\right)$	-0.000324
$p\left(k-1\right)\sqrt{p\left(k-1\right)}$	-0.003098
$V(k-1)\cdot Bs(k-1)$	0.000312
$Kp(k-12)$	0.265301
$V(k-1)$	0.003683
$p\left(k-1\right)$	0.286045
$Kp\left(k-12\right)\cdot VBs(k-2)$	-0.012219
$VBs(k-1)$	-0.531734
$VBs(k-1)\sqrt{p\left(k-1\right)}$	0.195865

Table 7.6: Identified NARX model for 12-hour ahead predictions of the Kp index. Seventeen model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\left(k-1\right)\sqrt{p\left(k-1\right)}}$	0.000066
$\mathbf{Bs\left(k-1\right)}$	0.838922
$\mathbf{Bs\left(k-1\right)\sqrt{p\left(k-1\right)}}$	-0.213375
$Kp\left(k-24\right)\cdot Bs(k-2)$	0.011558
$V(k-1)^2$	-0.000004
$Bs\left(k-2\right)$	0.269300
$Bs(k-1)\cdot Bs(k-2)$	-0.066312
constant	-3.080364
$\sqrt{p\left(k-1\right)}$	1.023429
$V\left(k-1\right)$	0.008776
$Bs(k-1)^2$	0.014446
$VBs(k-1)$	-0.573961
$VBs(k-1)\sqrt{p\left(k-1\right)}$	0.120880
$Kp(k-25)^2$	0.007968
$Bs(k-2)^2$	0.012127
$VBs(k-1)\cdot VBs(k-2)$	0.034862
$VBs(k-2)$	-0.121102
$V(k-2)\cdot VBs\left(k-1\right)$	0.000240

Table 7.7: Identified NARX model for 24-hour ahead predictions of the Kp index. Eighteen model terms were identified with their corresponding estimated parameter values.

The results of this approach are shown in Figure 7.5. They display a similar pattern to the sliding window approach, i.e. there is a bias for low and high magnetic disturbances, and the error bars for high values of Kp ($Kp \geq 8$) become less meaningful. Once again, these characteristics are due to the uncommon number of cases of high values of the Kp index compared with the most common Kp values related with quiet activity periods of the magnetosphere.

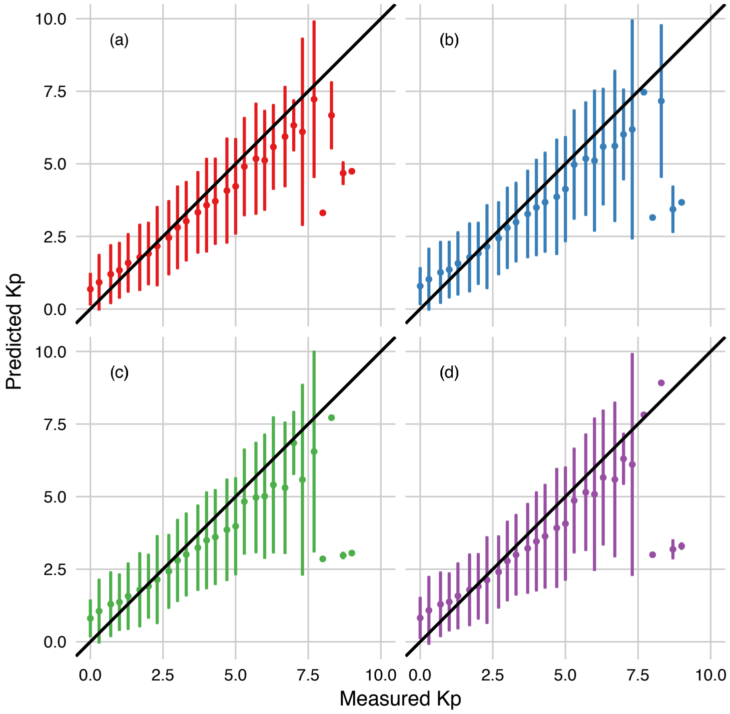

Figure 7.5: Comparison between the measured Kp index and predictions made for (a) 3, (b) 6, (c) 12, and (d) 24 hours ahead using the direct approach. The black line represents the ideal case when the prediction is equal to the measured Kp index. The points and bars correspond to the means and one standard deviations of the predictions made for each of the 28 Kp values.

To quantify the results, the root mean squared error (RMSE), correlation coefficient (ρ) and prediction efficiency (PE) are computed. These metrics are shown in Table 7.8.

The error for each of the four horizons of interest is respectively shown in Figure 7.6. Once again, there is a notoriously high error at the middle of July, corresponding to a period of high geomagnetic activity, as mentioned above. A glimpse of this period is shown in Figure 7.7. In addition, Table 7.9 shows a statistical summary of

Horizon	**RMSE**	ρ	**PE**
3	0.7593	0.8711	0.7585
6	0.8328	0.8424	0.7096
12	0.8623	0.8305	0.6895
24	0.8719	0.8265	0.6824

Table 7.8: Performance metrics for each of the four horizons of interest obtained with the direct approach.

the error time series in Figure 7.6. In general, it can be concluded that on average, this approach tends to slightly underpredict the Kp index given that the means are positive. Furthermore, as the number of hours to predict ahead increases, the forecasts are less accurate because the interquartile range (1st quartile - 3rd quartile) increases, as expected.

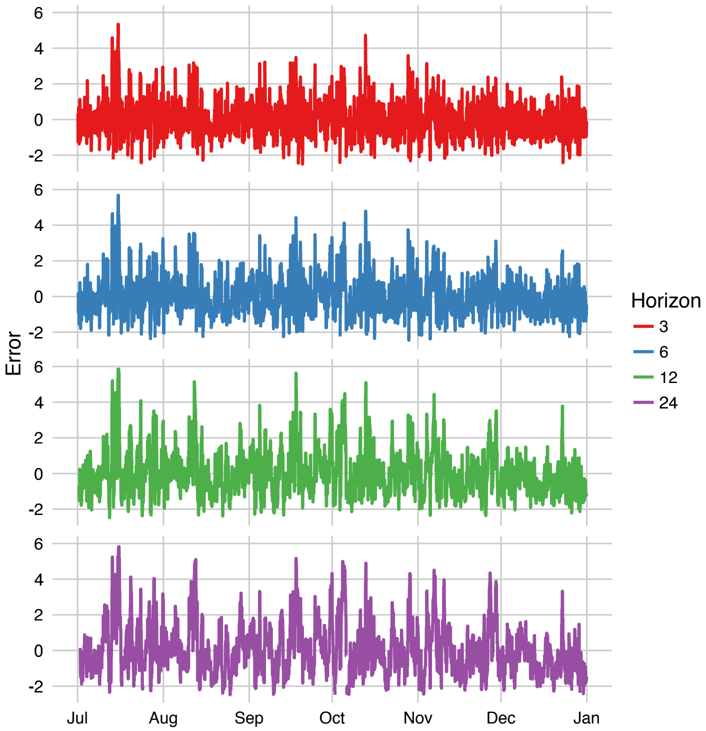

Figure 7.6: Error time series for the four horizons of interest obtained with the direct approach.

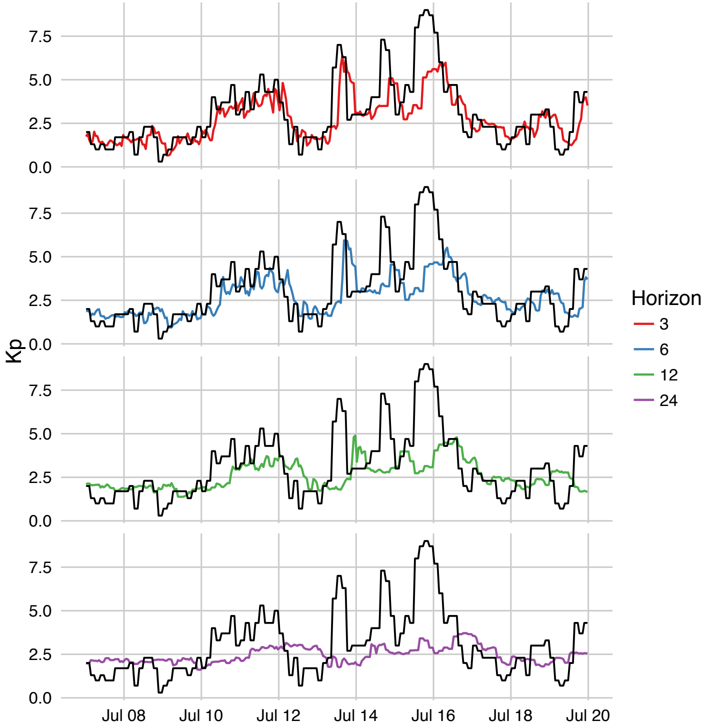

Figure 7.7: Predictions of the Kp index for the four horizons of interest during the middle of July 2000 using the direct approach. The black line corresponds to the measured Kp values.

Statistic	**Forecast**			
	3	**6**	**12**	**24**
Minimum	-2.3890	-2.6040	-2.8780	-3.5550
1st Quartile	-0.4436	-0.4842	-0.4910	-0.5073
Median	-0.0138	-0.0096	0.0068	-0.0107
Mean	0.0373	0.0433	0.0575	0.0446
3rd Quartile	0.4625	0.4950	0.5210	0.5005
Maximum	4.6880	5.6140	5.9440	5.7260

Table 7.9: Statistical summary for the error time series shown in Figure 7.6.

7.5 Discussion

A quick view to Tables 7.3 and 7.9 shows that the direct approach provides better forecasts than the sliding window approach because the means and medians are closer to zero, and the interquartile ranges are smaller. To better visualise this difference, a randomly selected 30-day interval on the second half of year 2000 is taken. The features dynamics are shown in Figure 7.8.

The model forecasts using both approaches during this 30-day interval are shown in Figures 7.9-7.12.

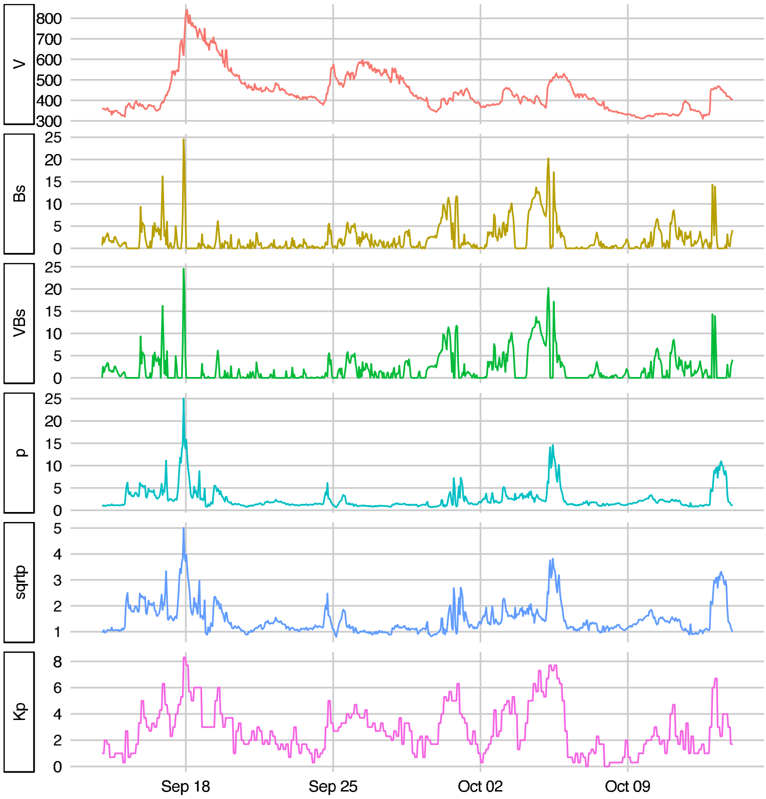

Figure 7.8: Feature dynamics during a 30-day interval on the second half of year 2000. The variable sqrtp corresponds to $\sqrt{p\,(t)}$.

To quantify the results, the root mean squared error (RMSE), correlation coefficient (ρ) and prediction efficiency (PE) are computed. These metrics are shown in Table 7.10.

These results show that better forecast accuracy is obtained by the direct approach. This is an expected result given that the sliding window approach uses model predicted outputs from a single model, and long-term forecasts tend to deviate from true values as time goes on. On the other hand, the direct approach uses a separate model for each horizon and relies on single calculations for h-step ahead predictions. However, both approaches show that predictions for low and high disturbances are slightly biased from the true values. This observation is coincident with previous findings reported in [161] and [162], where a model will perform well for the most common training values, while predictions for others will be poor. Another explanation is that this comes as a trade-off for using a regression model to predict a categorical output variable.

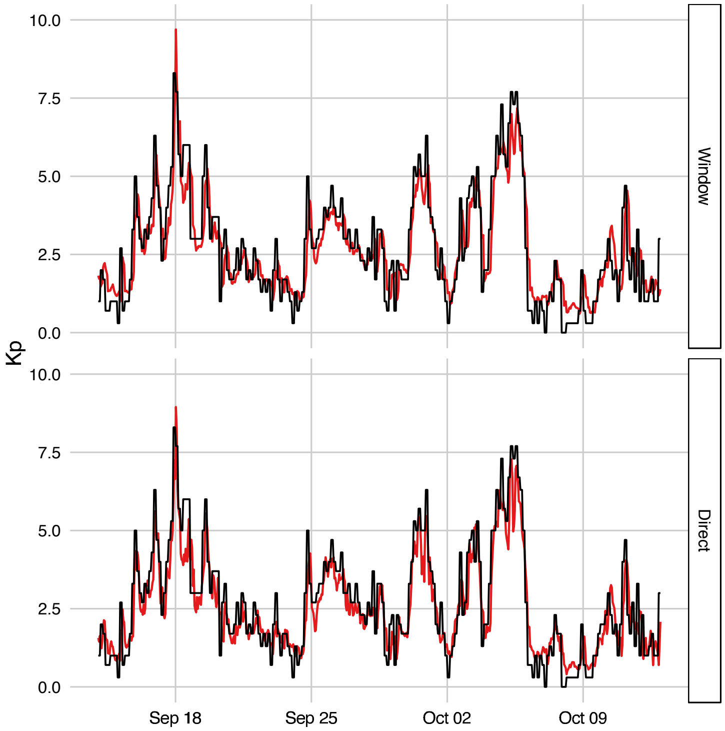

Figure 7.9: Comparison between the sliding window and direct approaches for 3-hour ahead predictions of the Kp index during a 30-day interval between September and October of year 2000. The black line corresponds to the measured Kp values.

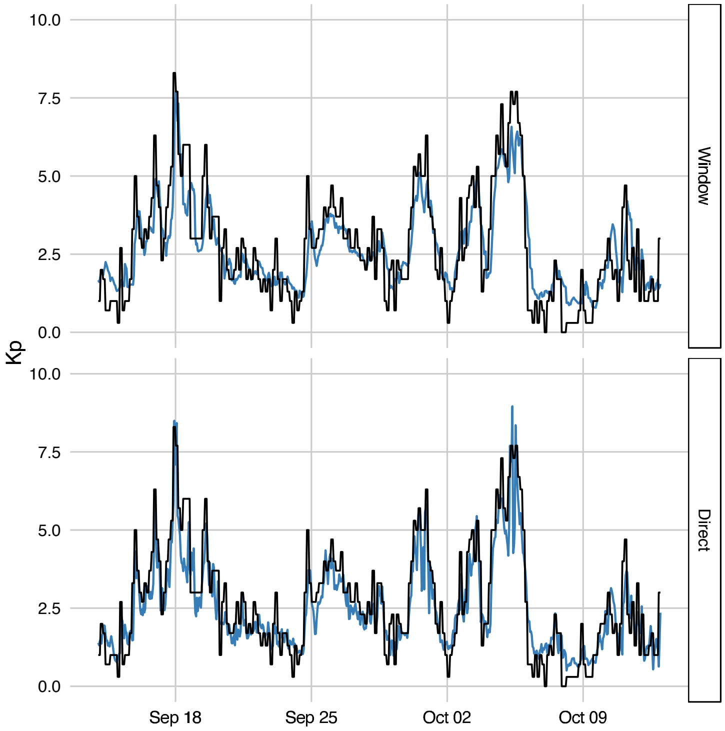

Figure 7.10: Comparison between the sliding window and direct approaches for 6-hour ahead predictions of the Kp index during a 30-day interval between September and October of year 2000. The black line corresponds to the measured Kp values.

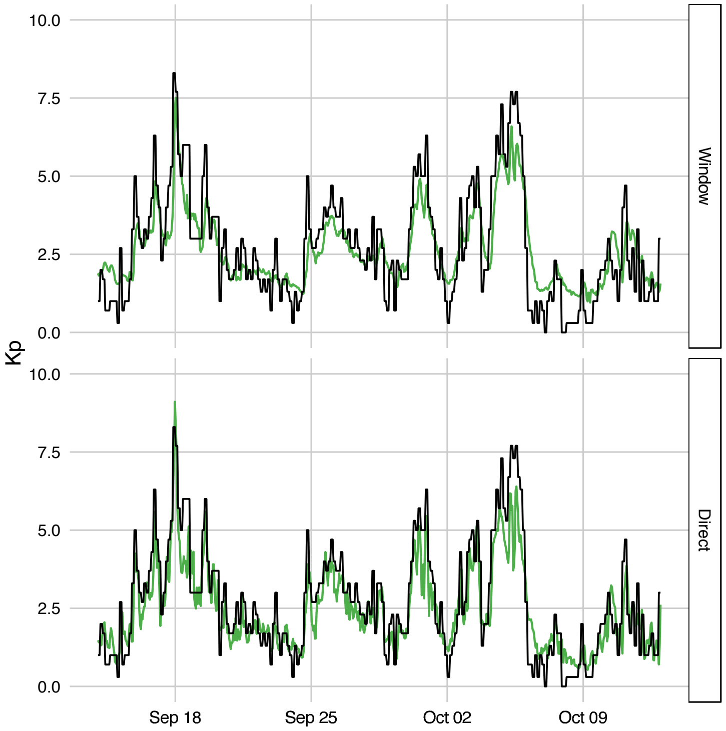

Figure 7.11: Comparison between the sliding window and direct approaches for 12-hour ahead predictions of the Kp index during a 30-day interval between September and October of year 2000. The black line corresponds to the measured Kp values.

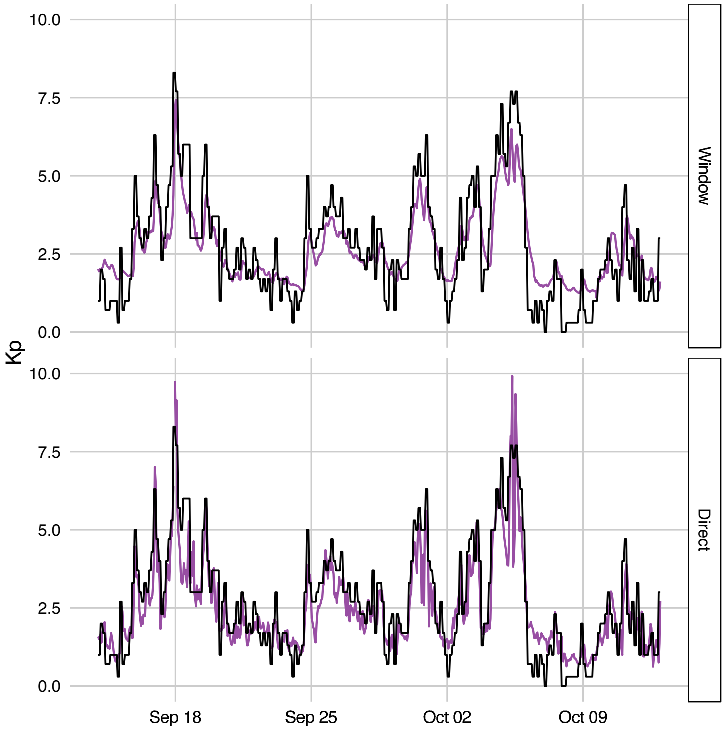

Figure 7.12: Comparison between the sliding window and direct approaches for 24-hour ahead predictions of the Kp index during a 30-day interval between September and October of year 2000. The black line corresponds to the measured Kp values.

Horizon	Approach	RMSE	ρ	PE
3	Window	0.8308	0.8874	0.7828
	Direct	**0.7582**	**0.9156**	**0.8287**
6	Window	0.9298	0.8628	0.7283
	Direct	**0.8053**	**0.9071**	**0.8105**
12	Window	0.9546	0.8728	0.7138
	Direct	**0.8537**	**0.9054**	**0.7919**
24	Window	0.9569	0.8804	0.7125
	Direct	**0.8588**	**0.8875**	**0.7831**

Table 7.10: Performance metrics for each of the four horizons of interest using the sliding window and direct approaches during a 30-day interval between September and October of year 2000.

Comparing the results obtained with those presented in [158], the values of the two model performance metrics (i.e. prediction performance and correlation coefficient) calculated from the results are slightly lower. This may be explained from several factors: i) all the data for all input and output variables used for model estimation in this study are raw data sampled hourly where no pre-processing (e.g. smoothing, interpretation, etc.) is performed; ii) the model input variables used in this work are not exactly the same as those used in previous studies; iii) some coefficients required by the models, for example the maximum lags of the input and output variables, may need to be optimised further. Note that one of the objectives of this analysis is to generate compact transparent models to show how Kp index depends on solar wind parameters and geomagnetic field indices, and then use such models to do further analysis including forecast. As shown in Tables 7.4-7.7, an important contribution obtained from the direct approach is that there are three significant model terms that are shared by all the models. These are shown in bold in Tables 7.4-7.7. The values of the three terms, together with the Kp index, are normalised, and the associated scatter plots are shown in Figure 7.13 (note that the normalisation of the values is just to facilitate the visualisation and comparison of the scatter plots). The importance of the first selected common model term $V(k-1)\sqrt{p(k-1)}$ may be roughly explained by its relevance with Kp when measuring the correlation coefficient ($\rho = 0.6149$). Model terms ranked later would normally not be so important as the top ones, and their correlation with the Kp

signal becomes very weak.

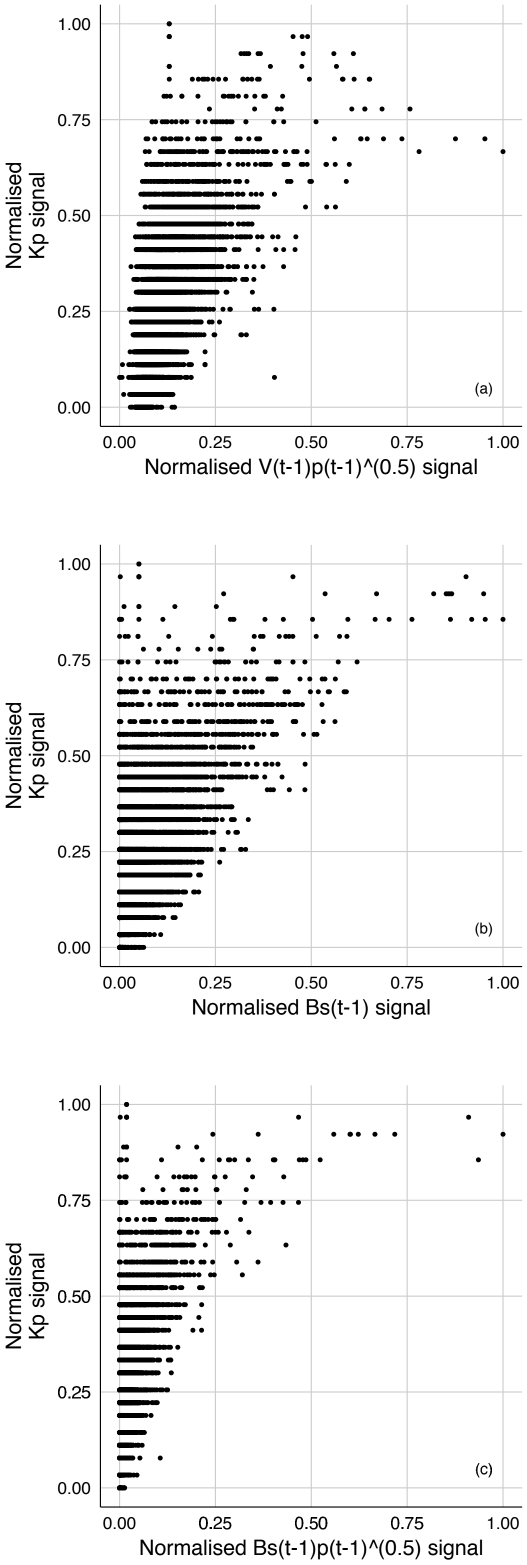

Figure 7.13: Top three significant model terms shared by all models in the direct approach. The correlation coefficients are (a) 0.6149, (b) 0.5571 and (c) 0.5437.

The importance of the model terms selected in Tables 7.4-7.7 is not always measured by the values or amplitude of these model terms. A model term with a high (or low) value does not necessarily mean a high (or low) value in Kp index, as its change is an outcome of combined and weighted interactions of many lagged input variables. Experience shows that top model terms can reflect the major varying trend of the output signals, while model terms ranked later can be useful in revealing local and relatively minor changes. While the role of solar wind speed and dynamic pressure as drivers of the Kp index has been confirmed by previous studies, this analysis provides some further information with an explicit format of these input variables, showing what kind of interactions of these drivers make a contribution to the change of the Kp index. This is important for further understanding and analysis of the dependent relationship of the Kp index on solar wind speed and dynamic pressure, among other drivers.

7.6 Logistic NARX approach

It is interesting to mention that this case study served as a motivation for the development of the Logistic NARX approach. However, this cannot be applied as it is, given that it only works with binary classification problems. For the sake of completion, the logistic NARX approach is applied to check its performance in this case study. This requires the substitution of the 28 classes of the Kp index into only 2. However, there is no reference that supports the partition of the data in 2 classes. In [166], it is mentioned that the Space Weather Prediction Center (in the USA) classifies geomagnetic activity in four levels: quiet to unsettled ($Kp < 4$), active ($Kp = 4$), minor storm ($Kp = 5$), and major to severe storm ($Kp > 5$). On the other hand, the Space Environment Prediction Center (in China) adopted a different classification scheme: quiet to unsettled ($Kp < 4$), active ($Kp = 4$), minor to moderate ($Kp = 5$ or 6), and major to severe storm ($Kp > 6$). For this reason, the 2 classes considered are: quiet to unsettled ($Kp < 4$) and active ($Kp \geq 4$).

Once again, the first six months of year 2000 are used for training while the second half of the year is used for validation. Figure 7.14 shows the corresponding histogram. It is noticed that there is an imbalance problem with the training set as the 'quiet' class has a presence of 86.81%. To overcome this, the Synthetic Minority Over-sampling TEchnique (SMOTE) described in [157] is applied, which in summary takes samples from the training set to create a new one with balanced classes.

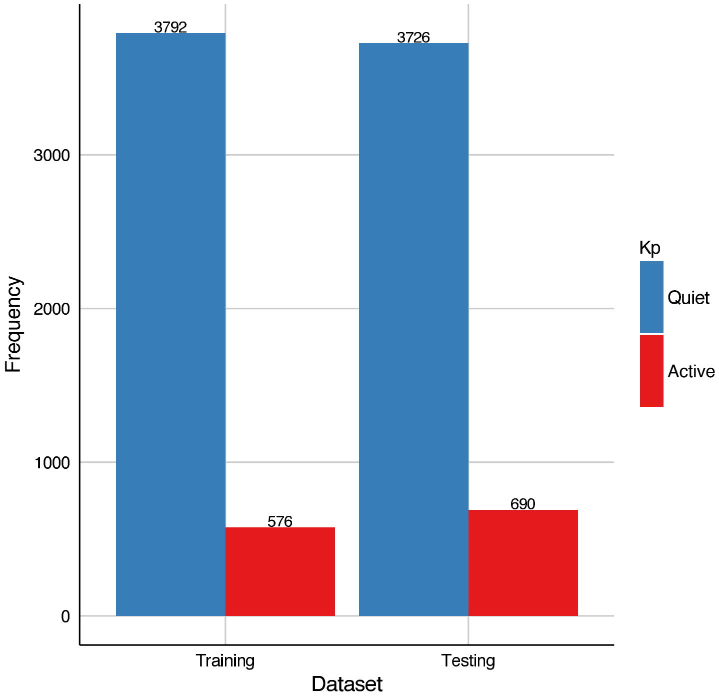

Figure 7.14: Frequency of the two classes of the Kp index for year 2000 on the training and testing sets. There is an imbalance problem with the training set as the 'quiet' class has a presence of 86.81%. To overcome this, the Synthetic Minority Over-sampling TEchnique (SMOTE) described in [157] is applied.

The logistic NARX modelling approach is applied to this new data set. The output variable is the binary Kp index, and the input variables are the same indicated in Table 7.1. Similar to the sliding window and direct approaches, the corresponding lags are $n_y = 4$ and $n_u = 2$, and the nonlinear degree is $\ell = 2$. The maximum number of terms to look for is chosen as $m_{max} = 15$, and 10 folds are used to compute the CV accuracy. These parameters are used for the modelling of the four horizons of interest: 3, 6, 12 and 24 hours ahead. Furthermore, as mentioned in section 2.4.3, the h-step ahead prediction is defined with respect to the system output; it is actually still one-step ahead prediction with respect to the system input. The models for each of the four horizons of interest are shown in Tables 7.11-7.14.

In order to assess the performance of the resultant logistic NARX models, regression-like NARX models based on the approach suggested in [47], random forests with 500 trees, support vector machines with a radial basis kernel, and k-nearest neighbors models are trained. The results are shown in Table 7.15, where it is possible to see that all models have a competitive performance. From these, the logistic NARX models may be preferred given the advantages highlighted in Chapter 6.

Model Term	**Parameter**
$Kp\,(t-3)$	2.0533
constant	-5.7990
$Bs\,(k-1)$	0.6675
$Kp(k-3)\cdot Bs(k-1)$	-0.0826
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	0.0051

Table 7.11: Identified logistic NARX model for 3-hour ahead predictions of the Kp index. Five model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	0.0061
constant	-6.1322
$Bs\,(k-1)$	0.6663

Table 7.12: Identified logistic NARX model for 6-hour ahead predictions of the Kp index. Three model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	0.0059
constant	-6.1678
$V(k-12)\cdot Bs(k-1)$	0.0016

Table 7.13: Identified logistic NARX model for 12-hour ahead predictions of the Kp index. Three model terms were identified with their corresponding estimated parameter values.

Model Term	**Parameter**
$\mathbf{V\,(k-1)\,\sqrt{p\,(k-1)}}$	0.0055
constant	-5.7115
$V(k-12)\cdot Bs(k-1)$	0.0015

Table 7.14: Identified logistic NARX model for 24-hour ahead predictions of the Kp index. Three model terms were identified with their corresponding estimated parameter values.

		Horizon			
		3	**6**	**12**	**24**
Method	**Logistic NARX**	**0.8749**	0.8710	0.8774	**0.8780**
	Regression NARX	0.8704	0.8706	**0.8804**	0.8721
	Random Forest	0.8677	0.8715	0.8724	0.8700
	Support Vector Machine	0.8677	**0.8740**	0.8720	0.8677
	K-Nearest Neighbors	0.8015	0.7667	0.7541	0.7573

Table 7.15: Accuracy performance between different methods for modelling of the binary Kp index. The logistic NARX model is compared against a regression-like NARX model based on the approach suggested in [47], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model. The same autoregressive inputs are used in all cases. Although all models have a competitive performance, the logistic NARX models may be preferred given that they provide transparency and interpretability about the contribution of individual regressors together with a classification probability for the predictions.

Furthermore, although it may be argued that the regression NARX models have a similar performance to the logistic ones, the number of model terms used for the regression-like models are 8 (3-hour ahead), 11 (6-hour ahead), 11 (12-hour ahead), and 16 (24-hour ahead), clearly showing the advantage of the simpler logistic NARX models. Also, it can be seen that the logistic NARX models select the model term $V\left(k-1\right)\sqrt{p\left(k-1\right)}$ again, indicating that this plays an important role in the dynamics of the Kp index.

7.7 Summary

In this chapter, the NARX modelling methodology is applied to the forecasting of the Kp index. A set of models are obtained using two different implementation approaches namely, recursive prediction approach based on sliding windows and a direct approach, which can directly generate h-hour ahead predictions (h = 3, 6, 12 and 24 in the case studies). In general, good forecasts are obtained for both short and long-term prediction using the estimated NARX models, but the direct approach outperforms the recursive approach. Nevertheless, both approaches tend to show that predictions for low and high disturbances are slightly biased from the true values. As previously reported, such a bias is a result of the uneven distribution in the output signal, and the use of a regression model to predict a categorical output variable may also play a role on this matter. An interesting property obtained from the direct approach is a set of significant model terms that are shared by all the models, regardless of the time horizon of interest. While the role of the solar wind speed and dynamic pressure as drivers of the Kp index has been confirmed by previous studies, the present analysis produced some further information showing the relative contributions made by these drivers to the changes in the Kp index. This is useful for further understanding the relationship of the Kp index to solar wind. It was noticed that the values of prediction performance and correlation coefficient relating to the trained models are slightly lower than those reported by [158] and possible reasons are briefly discussed. Finally, the logistic NARX modelling approach is applied to the binary version of the Kp index, showing that it is a competitive alternative to other classification techniques. The results of this chapter were published in [31].

Chapter 8

Conclusions

8.1 Summary and Conclusions

This thesis has focused on adapting and developing a new framework for the NARX methodology. This has been applied in the analysis of several environmental case studies.

For the first time, *a package in the R programming language is developed as a tool to help in the training of NARX models.* This package implements the traditional OFR algorithm together with some improvements to it that include nonlinear dependency metrics and a systematic way of selecting the number of model terms. It also includes a set of features for effectively performing model selection, parameter estimation, model validation, model visualisation and model evaluation.

Furthermore, *two new major components are added to the OFR algorithm. The first one combines the distance correlation metric, which can provide interpretability of nonlinear dependencies, and the bagging method, which can provide an uncertainty analysis, to extend the deterministic notion of the OFR algorithm.* This implementation provides several advantages including the ability to assess the main model terms, the computation of a bootstrap distribution for the forecasts made by the many models trained, and that there is no need to specify prior distributions, which is usually a requirement when performing Bayesian analysis.

The second major component improves the NARX methodology in order to handle *binary outputs*. In this improvement, the logistic regression is combined with the OFR algorithm, which allows to sequentially select continuous lagged regressors that play an important role individually and/or interactively in the classification process. The selection is done by means of the biserial correlation, a metric that is suitable when working with binary variables. The results obtained show that the logistic NARX model is a competitive alternative to other classification techniques.

All these improvements are applied in two case studies. *The first one analyses the Atlantic Meridional Overturning Circulation (AMOC)*, where several models are trained to forecast the AMOC anomaly. Here it is proved that nonlinear models outperformed linear ones, and that only second-order models are of interest, involving a modulation of the wind and density difference variables. The best model trained is then used to hindcast the AMOC signal back to 1980. Here, it is found that the density difference between the northern sinking waters and the Gulf of Mexico source waters of the main overturning current have a significant contribution with a dominant lag time of 7 months. This was confirmed by the BFOR-dCor algorithm.

The second case scenario focuses on the *modelling of global magnetic disturbances in near-Earth space using the Kp index*. Higher values are rare and their prediction becomes challenging. For this task, two different implementation approaches are taken: one based on a recursive approach where a single model is trained using data within a sliding window and then used to make forecasts. The second approach consists on training four different models, each focusing on a horizon h of interest to make h-hour ahead predictions (where $h = 3$, 6, 12 and 24). It is found that the direct approach outperforms the recursive approach, but the predictions for low and high disturbances are slightly biased from the true values. Such a bias is attributed to the uneven distribution in the Kp index, and the use of a regression model to predict a categorical output variable. This motivates the use of the logistic NARX approach by considering a binary version of the Kp index. The whole analysis gives further understanding of the relationship of the Kp index to solar wind parameters.

8.2 Future Work

This section presents current limitations with the research developed, and possible solutions and suggestions:

- The choice of maximum lags for both input and output variables is important given that these define the size of the search space, and the influence of the past in future values. Nevertheless there is still no systematic approach to make such choice. Two common methods are proposed in [46] and [50], but further research is required.

- The BFOR-dCor algorithm offers an alternative by building an ensemble of NARX models to reduce the variance in the predictions. However, given the forward recursion of the OFR algorithm, certain model structures tend to get repeated which may result in an increase in the bias. One way that was investigated to overcome this involves a combination of Random Forests with the NARX methodology to diversify the model selection during the training process. However this did not produce the expected results compared with other traditional algorithms. This is an interesting problem that deserves further investigation.

- The logistic NARX methodology proposed in chapter 6 was developed for binary classification problems. However, many classification problems involve more than two classes, i.e. like the 28 levels of the Kp index (section 7.2). Such scenarios cannot be addressed with the proposed algorithm 6.1. One way to overcome this could be the use of the one-VS-one or one-VS-all approaches [172]. Of course, this would require the training of several models which could be problematic to handle. Another possibility could be the use of neural networks with a softmax layer at the final stage [173,174]. This would produce a single model that could handle several classes, with the inconvenience that interpretability would be lost.

- One scenario that has been of great interest is the combination of the NARX methodology with Bayesian methods. Several studies have been conducted including [2, 57, 58]. However, most of these require the explicit selection of conjugate priors to facilitate the computation of the posterior distributions. It would be of great interest to investigate possible alternatives that do not require the conjugate priors.

- Nowadays there is great interest in the use of huge amounts of data, a.k.a. Big Data. In particular, there is an area that has benefited from those huge data volumes known as Deep Learning. This area mostly uses different architectures of neural networks, which have produced a lot of promising results in different fields. It would be interesting to extend the NARX methodology, and possibly combine it with Deep Learning techniques, to handle big data problems.

Bibliography

[1] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems.* Prentice Hall, 1983.

[2] T. Baldacchino, “Statistical estimation and identification of nonlinear dynamic systems,” Ph.D. dissertation, The University of Sheffield, 2011.

[3] V. Babovic, “Data mining in hydrology,” *Hydrological processes*, vol. 19, no. 7, pp. 1511–1515, 2005.

[4] S. Rogers and M. Girolami, *A First Course in Machine Learning.* CRC Press, 2012.

[5] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains.* Wiley, 2013.

[6] SINTEF, “Big data, for better or worse: 90% of world´s data generated over last two years.” [Online]. Available: http://www.sciencedaily.com/releases/2013/05/130522085217.htm

[7] Y. N. Harari, *Homo Deus: A brief history of tomorrow.* Penguin Random House, 2016.

[8] Y. S. Abu-Mostafa, “Machines that think for themselves,” *Scientific American*, vol. 289, no. 7, pp. 78–81, July 2012.

[9] S. Olafsson, X. Li, and S. Wu, “Operations research and data mining,” *European Journal of Operational Research*, vol. 187, no. 3, pp. 1429–1448, 2008.

[10] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. AMLBook, 2012.

[11] "Observing Dark Worlds," October 2012. [Online]. Available: http://www.kaggle.com/c/DarkWorlds

[12] S. A. Billings and D. Coca, "Identification of NARMAX and related models," Department of Automatic Control and Systems Engineering, The University of Sheffield, UK, Tech. Rep., 2001.

[13] S. M. Weiss and N. Indurkhya, *Predictive Data Mining: a practical guide*. Morgan Kaufmann, 1998.

[14] J. Spate, K. Gibert, M. Sànchez-Marrè, E. Frank, J. Comas, I. Athanasiadis, and R. Letcher, "Data mining as a tool for environmental scientists," in *1st iEMSs Workshop DM-TEST 2006*. International Environmental Modelling and Software Society, 2006.

[15] "What is the difference between data mining, statistics, machine learning and AI?" October 2013. [Online]. Available: http://stats.stackexchange.com/q/5026/56779

[16] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[17] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013.

[18] W. W. Piegorsch and A. J. Bailer, *Analyzing environmental data*. John Wiley & Sons, 2005.

[19] K. Gibert, M. Sanchez-Marre, and I. Rodríguez-Roda, "GESCONDA: An intelligent data analysis system for knowledge discovery and management in environmental databases," *Environmental Modelling & Software*, vol. 21, no. 1, pp. 115–120, 2006.

[20] P. Cortez and A. d. J. R. Morais, “A Data Mining Approach to Predict Forest Fires Using Meteorological Data,” 2007.

[21] F. C. Morabito, “Environmental data interpretation: the next challenge for intelligent systems,” in *NATO Advanced Research Workshop on Systematic Organization of Information in Fuzzy Systems. Vila Real, Portugal*, 2001.

[22] G. Hanrahan, Ed., *Modelling of Pollutants in Complex Environmental Systems*, ser. Advanced Topics in Environmental Science. ILM Publications, 2010, vol. 1.

[23] J.-F. Mas, H. Puig, J. L. Palacio, and A. Sosa-López, “Modelling deforestation using GIS and artificial neural networks,” *Environmental Modelling & Software*, vol. 19, no. 5, pp. 461–471, 2004.

[24] L. A. Belanche, J. J. Valdés, J. Comas, I. R. Roda, and M. Poch, “Towards a model of input-output behaviour of wastewater treatment plants using soft computing techniques,” *Environmental Modelling & Software*, vol. 14, no. 5, pp. 409–419, 1999.

[25] M. W. Buckley and J. Marshall, “Observations, inferences, and mechanisms of the Atlantic Meridional Overturning Circulation: A review,” *Reviews of Geophysics*, 2016.

[26] M. F. Thomsen, “Why Kp is such a good measure of magnetospheric convection,” *Space Weather*, vol. 2, no. 11, pp. n/a–n/a, 2004, s11004. [Online]. Available: http://dx.doi.org/10.1029/2004SW000089

[27] T. G. Dietterich, “Machine Learning for Sequential Data: A Review,” in *Structural, Syntactic, and Statistical Pattern Recognition.* Springer, 2002, pp. 15–30.

[28] P. Aguilera, A. Fernández, R. Fernández, R. Rumí, and A. Salmerón, “Bayesian networks in environmental modelling,” *Environmental Modelling & Software*, vol. 26, no. 12, pp. 1376–1388, 2011.

[29] J. Ayala Solares, H.-L. Wei, and S. A. Billings, “A novel logistic-NARX model as a classifier for dynamic binary classification,” *Neural Computing and Applications*, 2017.

[30] J. Ayala Solares and H.-L. Wei, “Nonlinear model structure detection and parameter estimation using a novel bagging method based on distance correlation metric,” *Nonlinear Dynamics*, pp. 1–15, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11071-015-2149-3

[31] J. R. Ayala Solares, H.-L. Wei, R. J. Boynton, S. N. Walker, and S. A. Billings, “Modeling and prediction of global magnetic disturbance in near-Earth space: A case study for Kp index using NARX models,” *Space Weather*, vol. 14, no. 10, pp. 899–916, 2016, 2016SW001463. [Online]. Available: http://dx.doi.org/10.1002/2016SW001463

[32] J. Leek, “Data analysis,” [Lecture on "What is data?"], January 2013.

[33] H. Wickham, “Tidy data,” *The Journal of Statistical Software*, vol. 59, 2014. [Online]. Available: http://www.jstatsoft.org/v59/i10/

[34] Y. S. Abu-Mostafa, “Machine Learning Video Library,” 2013. [Online]. Available: http://work.caltech.edu/library/

[35] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[36] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[37] T. Söderström and P. Stoica, *System Identification*. Prentice Hall, 1989.

[38] S. L. Kukreja, J. Lofberg, and M. J. Brenner, “A least absolute shrinkage and selection operator (LASSO) for nonlinear system identification,” in *System Identification*, vol. 14, no. 1, 2006, pp. 814–819.

[39] K. J. Pope and P. J. W. Rayner, “Non-linear system identification using Bayesian inference,” in *Acoustics, Speech, and Signal Processing, 1994.*

ICASSP-94., 1994 IEEE International Conference on, vol. IV, 1994, pp. 457 – 460.

[40] D. Chinarro, *System Engineering Applied to Fuenmayor Karst Aquifer (San Julián de Banzo, Huesca) and Collins Glacier (King George Island, Antarctica)*. Springer, 2014.

[41] S. L. Kukreja, H. Galiana, and R. Kearney, "Structure detection of NARMAX models using bootstrap methods," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 1, 1999, pp. 1071–1076.

[42] H.-L. Wei and S. A. Billings, "Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information," *International Journal of Modelling, Identification and Control*, vol. 3, no. 4, pp. 341–356, 2008.

[43] R. Haber and H. Unbehauen, "Structure Identification of Nonlinear Dynamic Systems—A Survey on Input/Output Approaches," *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.

[44] S. L. Kukreja, "A suboptimal bootstrap method for structure detection of NARMAX models," Linköpings universitet, Linköping, Sweden, Tech. Rep. LiTH-ISY-R-2452, 2002.

[45] X. Hong and C. J. Harris, "Nonlinear Model Structure Detection using Optimum Experimental Design and Orthogonal Least Squares," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 435–439, 2001.

[46] H.-L. Wei, S. A. Billings, and J. Liu, "Term and variable selection for nonlinear system identification," *International Journal of Control*, vol. 77, no. 1, pp. 86–110, 2004.

[47] S. A. Billings and H.-L. Wei, "An adaptive orthogonal search algorithm for model subset selection and non-linear system identification," *International Journal of Control*, vol. 81, no. 5, pp. 714–724, 2008.

[48] M. Han and X. Liu, "Forward Feature Selection Based on Approximate Markov Blanket," in *Advances in Neural Networks-ISNN 2012*. Springer, 2012, pp. 64–72.

[49] L. A. Aguirre and C. Jácôme, "Cluster analysis of NARMAX models for signal-dependent systems," in *Control Theory and Applications, IEE Proceedings-*, vol. 145, no. 4. IET, July 1998, pp. 409–414.

[50] B. Feil, J. Abonyi, and F. Szeifert, "Model order selection of nonlinear input-output models—a clustering based approach," *Journal of Process Control*, vol. 14, no. 6, pp. 593–602, 2004.

[51] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[52] X. Hong and S. Chen, "An elastic net orthogonal forward regression algorithm," in *16th IFAC Symposium on System Identification*, July 2012, pp. 1814–1819.

[53] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[54] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Application in R*, ser. Springer Texts in Statistics. Springer, 2013, vol. 103.

[55] S. Sette and L. Boullart, "Genetic programming: principles and applications," *Engineering Applications of Artificial Intelligence*, vol. 14, no. 6, pp. 727–736, 2001.

[56] J. Madár, J. Abonyi, and F. Szeifert, "Genetic programming for the identification of nonlinear input-output models," *Industrial & Engineering Chemistry Research*, vol. 44, no. 9, pp. 3178–3186, 2005.

[57] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, “Structure detection and parameter estimation for NARX models in a unified EM framework,” *Automatica*, vol. 48, no. 5, pp. 857–865, 2012.

[58] ——, “Computational system identification for Bayesian NARMAX modelling,” *Automatica*, vol. 49, no. 9, pp. 2641–2651, September 2013.

[59] S. Chen, X. Wang, and D. Brown, “Orthogonal Least Square with Boosting for Regression,” in *Intelligent Data Engineering and Automated Learning - IDEAL 2004*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3177, pp. 593–599.

[60] S. A. Billings, S. Chen, and R. J. Backhouse, “The identification of linear and non-linear models of a turbocharged automotive diesel engine,” *Mechanical Systems and Signal Processing*, vol. 3, no. 2, pp. 123–142, 1989.

[61] L. A. Aguirre and C. Letellier, “Modeling Nonlinear Dynamics and Chaos: A Review,” *Mathematical Problems in Engineering*, vol. 2009, no. 35, 2009.

[62] H.-L. Wei and S. A. Billings, “Improved parameter estimates for non-linear dynamical models using a bootstrap method,” *International Journal of Control*, vol. 82, no. 11, pp. 1991–2001, 2009.

[63] L. Z. Guo, S. A. Billings, and D. Q. Zhu, “An extended orthogonal forward regression algorithm for system identification using entropy,” *International Journal of Control*, vol. 81, no. 4, pp. 690–699, 2008. [Online]. Available: http://dx.doi.org/10.1080/00207170701701031

[64] S. A. Billings and H.-L. Wei, “Sparse Model Identification Using a Forward Orthogonal Regression Algorithm Aided by Mutual Information,” *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 306–310, 2007.

[65] S. Chen and S. A. Billings, “Representation of non-linear systems: the NARMAX model,” *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, March 1989.

[66] D. Koller and M. Sahami, “Toward optimal feature selection,” in *In 13th International Conference on Machine Learning*, 1995.

[67] S. Wang, H.-L. Wei, D. Coca, and S. A. Billings, “Model term selection for spatio-temporal system identification using mutual information,” *International Journal of Systems Science*, vol. 44, no. 2, pp. 223–231, 2013.

[68] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, “Detecting novel associations in large data sets,” *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011. [Online]. Available: http://www.sciencemag.org/content/334/6062/1518.abstract

[69] T. Speed, “A correlation for the 21st century,” *Science*, vol. 334, no. 6062, pp. 1502–1503, 2011. [Online]. Available: http://www.sciencemag.org/content/334/6062/1502.short

[70] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[71] M. Gorfine, R. Heller, and Y. Heller, “Comment on "Detecting novel associations in large data sets",” November 2012. [Online]. Available: http://emotion.technion.ac.il/~gorfinm/files/science6.pdf

[72] N. Simon and R. Tibshirani, “Comment on "Detecting Novel Associations In Large Data Sets" by Reshef Et Al, Science Dec 16, 2011,” January 2014. [Online]. Available: http://arxiv.org/abs/1401.7645

[73] J. Madar, J. Abonyi, and F. Szeifert, “Genetic programming for system identification,” in *Intelligent Systems Design and Applications (ISDA 2004) Conference, Budapest, Hungary*, 2004.

[74] C. K. Chui, *An Introduction to Wavelets*. Elsevier, 2016.

[75] G. Nason, *Wavelet Methods in Statistics with R.* Springer Science & Business Media, 2010.

[76] A. K. Alexandridis and A. D. Zapranis, "Wavelet neural networks: A practical guide," *Neural Networks*, vol. 42, pp. 1–27, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608013000129

[77] S. A. Billings and H.-L. Wei, "A New Class of Wavelet Networks for Nonlinear System Identification," *Neural Networks, IEEE Transactions on*, vol. 16, no. 4, pp. 862–874, 2005.

[78] ——, "The wavelet-NARMAX representation: a hybrid model structure combining polynomial models with multiresolution wavelet decompositions," *International Journal of Systems Science*, vol. 36, no. 3, pp. 137–152, 2005.

[79] H.-L. Wei, M. A. Balikhin, and S. A. Billings, "Nonlinear time-varying system identification using the NARMAX model and multiresolution wavelet expansions," The University of Sheffield, United Kingdom, Tech. Rep. 829, 2003.

[80] S. A. Billings and Q. H. Tao, "Model validity tests for non-linear signal processing applications," *International Journal of Control*, vol. 54, no. 1, pp. 157–194, 1991.

[81] D. Coca and S. A. Billings, "Non-linear system identification using wavelet multiresolution models," *International Journal of Control*, vol. 74, no. 18, pp. 1718–1736, 2001.

[82] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice.* OTexts, 2014.

[83] E. G. Nepomuceno and S. A. M. Martins, "A lower bound error for free-run simulation of the polynomial NARMAX," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 50–58, 2016. [Online]. Available: http://dx.doi.org/10.1080/21642583.2016.1163296

[84] L. Piroddi and W. Spinelli, “An identification algorithm for polynomial NARX models based on simulation error minimization,” *International Journal of Control*, vol. 76, no. 17, pp. 1767–1781, 2003. [Online]. Available: http://dx.doi.org/10.1080/00207170310001635419

[85] Y. Guo, L. Guo, S. Billings, and H.-L. Wei, “An iterative orthogonal forward regression algorithm,” *International Journal of Systems Science*, vol. 46, no. 5, pp. 776–789, 2015. [Online]. Available: http://dx.doi.org/10.1080/00207721.2014.981237

[86] Y. Guo, L. Z. Guo, S. A. Billings, and H.-L. Wei, “Ultra-orthogonal forward regression algorithms for the identification of non-linear dynamic systems,” *Neurocomputing*, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215011741

[87] H. Mayfield, C. Smith, M. Gallagher, L. Coad, and M. Hockings, “Using Machine Learning to Make the Most out of Free Data: A Deforestation Case Study,” in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[88] M. Poch, J. Comas, J. Porro, M. Garrido-Baserba, L. Corominas, and M. Pijuan, “Where Are We In Wastewater Treatment Plants Data Management? A Review and a Proposal,” in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[89] R. Richardsa, J.-O. Meyneckec, O. Sahinb, R. Tillere, and Y. Liuf, “Ocean acidification and fisheries-a Bayesian network approach to assessing a wicked problem,” in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[90] D. P. Ames, N. W. Quinn, and A. E. Rizzoli, “Modelling spatial relationships between ecosystem services and agricultural production in an agent-based

model," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[91] D. Groenendyk, K. Thorp, T. Ferré, W. Crow, and D. Hunsaker, "A k-means clustering approach to assess wheat yield prediction uncertainty with a HYDRUS-1D coupled crop model," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[92] C. Benoît, C. Benoit, and B. Frederic, "Monitoring agricultural landscapes dynamics using the complementarities of optical, microwave and thermal remote sensing data," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[93] I. N. Athanasiadis, V. G. Kaburlasos, P. A. Mitkas, and V. Petridis, "Applying machine learning techniques on air quality data for real-time decision support," in *In: First International NAISO Symposium on Information Technologies in Environmental Engineering (ITEE'2003*. ICSC-NAISO Publishers, 2003, pp. 24–27.

[94] J. Zeng, N. Saigusa, T. Matsunaga, and T. Shirai, "Machine Learning in Surface Ocean CO2 Mapping," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[95] D. P. Ames and N. W. Quinn, "A model component for simulating CO2 emissions growth," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[96] T. Xu, B. Croke, M. F. Hutchinson *et al.*, "Identification of spatial and temporal patterns of Australian daily rainfall under a changing climate," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2014.

[97] A. Ausseil, K. Bodmin, A. Daigneault, E. Teixeira, E. Keller, M. Kirschbaum, L. Timar, A. Dunningham, C. Zammit, S. Stephens *et al.*, "Climate change

impacts and implications: an integrated assessment in a lowland environment of New Zealand," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[98] M. G. Porter and O. Sahin, "A Dynamic Water Supply Portfolio Optimisation Approach," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[99] C. Troost and T. Berger, "Simulating structural change in agriculture: Modelling farming households and farm succession," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[100] E. Bertone, O. Sahin, R. Richards, and A. Roiko, "Modelling with stakeholders: a systems approach for improved environmental decision making under great uncertainty," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[101] S. A. Abdulkareem, E.-W. Augustijn-Beckers, Y. T. Mustafa, and T. Filatova, "Artificial intelligence techniques to enhance actors' decision strategies in socio-ecological agent-based models," in *Proceedings of the 7th International Congress on Environmental Modelling and Software*, 2016.

[102] C. Vitolo, Y. Elkhatib, D. Reusser, C. J. Macleod, and W. Buytaert, "Web technologies for environmental Big Data," *Environmental Modelling & Software*, vol. 63, no. Supplement C, pp. 185 – 198, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815214002965

[103] J. Kerrou, G. Deman, L. Tacher, H. Benabderrahmane, and P. Perrochet, "Numerical and polynomial modelling to assess environmental and hydraulic impacts of the future geological radwaste repository in meuse site (france)," *Environmental Modelling & Software*, vol. 97, no. Supplement C, pp. 157 – 170, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815216301876

[104] D. C. Camilo, L. Lombardo, P. M. Mai, J. Dou, and R. Huser, "Handling high predictor dimensionality in slope-unit-based landslide susceptibility models through lasso-penalized generalized linear model," *Environmental Modelling & Software*, vol. 97, pp. 145 – 156, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815216311203

[105] F. V. Jensen, *Bayesian Networks and Decision Graphs.* Springer, 2009.

[106] V. Romero, R. Rumí, and A. Salmerón, "Learning hybrid Bayesian networks using mixtures of truncated exponentials," *International Journal of Approximate Reasoning*, vol. 42, no. 1–2, pp. 54–68, 2006, pGM'04. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X05000642

[107] B. R. Cobb, P. P. Shenoy, and R. Rumí, "Approximating probability density functions in hybrid Bayesian networks with mixtures of truncated exponentials," *Statistics and Computing*, vol. 16, no. 3, pp. 293–308, 2006.

[108] "R Documentation." [Online]. Available: https://www.r-project.org/other-docs.html

[109] "Datacamp: Introduction to R." [Online]. Available: https://www.datacamp.com/courses/free-introduction-to-r

[110] "Coursera: Data Science Specialization." [Online]. Available: https://www.coursera.org/specializations/jhu-data-science

[111] H. Wickham and G. Grolemund, *R for Data Science.* O'Reilly Media, 2016.

[112] R. Ihaka and R. Gentleman, "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[113] L. Torgo, *Data Mining with R: Learning with Case Studies.* Chapman & Hall/CRC, 2010.

[114] R. D. Peng, *R Programming for Data Science.* Leanpub, 2016.

[115] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis.* Springer Science & Business Media, 2009.

[116] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," Ph.D. dissertation, The University of Waikato, 1999.

[117] F. Harrell, *Regression Modeling Strategies: with Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis.* Springer, 2015.

[118] B. Ninness and T. Brinsmead, "A Bayesian Approach to System Identification using Markov Chain Methods," University of Newcastle, New South Wales, Australia, Tech. Rep. EE02009, 2003.

[119] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[120] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of Statistical Planning and Inference*, vol. 143, no. 8, pp. 1249–1272, 2013.

[121] S. Chen, S. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[122] B. Efron, "Computers and the theory of statistics: Thinking the unthinkable," *SIAM Review*, vol. 21, no. 4, pp. 460–480, October 1979.

[123] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, ser. Monographs on Statistics and Applied Probability. Chapman & Hall, 1993, vol. 57.

[124] A. C. Davison, *Bootstrap Methods and their Application.* Cambridge University Press, 1997.

[125] L. Breiman, "Bagging predictors," University of California, Berkeley, California, USA, Tech. Rep. 421, September 1994.

[126] (2003) Sunspot Data. [Online]. Available: http://sidc.oma.be/silso/datafiles

[127] P. Kampstra, "Beanplot: A boxplot alternative for visual comparison of distributions," *Journal of Statistical Software*, vol. 28, no. 1, November 2008.

[128] H. Lin, J. Varsik, and H. Zirin, "High-resolution observations of the polar magnetic fields of the Sun," *Solar Physics*, vol. 155, no. 2, pp. 243–256, 1994.

[129] G. Rilling, P. Flandrin, P. Goncalves *et al.*, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, vol. 3. IEEER, 2003, pp. 8–11.

[130] G. R. Bigg, R. C. Levine, and C. L. Green, "Modelling abrupt glacial North Atlantic freshening: rates of change and their implications for Heinrich events," *Global and Planetary Change*, vol. 79, no. 3, pp. 176–192, 2011.

[131] S. A. Cunningham, T. Kanzow, D. Rayner, M. O. Baringer, W. E. Johns, J. Marotzke, H. R. Longworth, E. M. Grant, J. J.-M. Hirschi, L. M. Beal *et al.*, "Temporal variability of the Atlantic meridional overturning circulation at 26.5 N," *science*, vol. 317, no. 5840, pp. 935–938, 2007.

[132] D. Smeed, G. McCarthy, S. Cunningham, E. Frajka-Williams, D. Rayner, W. E. Johns, C. Meinen, M. Baringer, B. Moat, A. Duchez *et al.*, "Observed decline of the Atlantic meridional overturning circulation 2004–2012," *Ocean Science*, vol. 10, no. 1, pp. 29–38, 2014.

[133] G. Danabasoglu, S. G. Yeager, W. M. Kim, E. Behrens, M. Bentsen, D. Bi, A. Biastoch, R. Bleck, C. Böning, A. Bozec *et al.*, "North Atlantic simulations in Coordinated Ocean-ice Reference Experiments phase II (CORE-II). Part II: Inter-annual to decadal variability," *Ocean Modelling*, vol. 97, pp. 65–90, 2016.

[134] S. F. Tett, T. J. Sherwin, A. Shravat, and O. Browne, "How much has the North Atlantic Ocean overturning circulation changed in the last 50 years?" *Journal of Climate*, vol. 27, no. 16, pp. 6325–6342, 2014.

[135] L. Hermanson, N. Dunstone, K. Haines, J. Robson, D. Smith, and R. Sutton, "A novel transport assimilation method for the Atlantic meridional overturning circulation at 26 N," *Quarterly Journal of the Royal Meteorological Society*, vol. 140, no. 685, pp. 2563–2572, 2014.

[136] A. Karspeck, D. Stammer, A. Köhl, G. Danabasoglu, M. Balmaseda, D. Smith, Y. Fujii, S. Zhang, B. Giese, H. Tsujino *et al.*, "Comparison of the Atlantic meridional overturning circulation between 1960 and 2007 in six ocean reanalysis products," *Climate Dynamics*, pp. 1–26, 2015.

[137] L. C. Jackson, K. A. Peterson, C. D. Roberts, and R. A. Wood, "Recent slowing of Atlantic overturning circulation as a recovery from earlier strengthening," *Nature Geoscience*, vol. 9, pp. 518–522, 2016.

[138] M. Collins, R. Knutti, J. Arblaster, J.-L. Dufresne, T. Fichefet, P. Friedlingstein, X. Gao, W. Gutowski, T. Johns, G. Krinner *et al.*, "Long-term climate change: projections, commitments and irreversibility," 2013.

[139] A. G. Barnston and R. E. Livezey, "Classification, seasonality and persistence of low-frequency atmospheric circulation patterns," *Monthly Weather Review*, vol. 115, no. 6, pp. 1083–1126, 1987.

[140] "Monitoring Weather and Climate," October 2002. [Online]. Available: http://www.cpc.ncep.noaa.gov/products/precip/CWlink/pna/nao_index.html

[141] A. E. Gill, *Atmosphere-Ocean Dynamics.* Elsevier, 2016.

[142] J. B. Ramsey, "Tests for specification errors in classical linear least-squares regression analysis," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 350–371, 1969.

[143] D. L. Hodson and R. T. Sutton, "The impact of resolution on the adjustment and decadal variability of the Atlantic meridional overturning circulation in a coupled climate model," *Climate Dynamics*, vol. 39, no. 12, pp. 3057–3073, 2012.

[144] J. Prior and M. Kendon, "The UK winter of 2009/2010 compared with severe winters of the last 100 years," *Weather*, vol. 66, no. 1, pp. 4–10, 2011.

[145] J. Pallant, *SPSS Survival Manual.* McGraw-Hill Education (UK), 2013.

[146] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A%3A1010933404324

[147] P. Komarek, "Logistic regression for data mining and high-dimensional classification," Master's thesis, Robotics Institute - School of Computer Science, Carnegie Mellon University, USA, 2004.

[148] A. Senawi, H.-L. Wei, and S. A. Billings, "A New Maximum Relevance-Minimum Multicollinearity (MRmMC) Method for Feature Selection and Ranking," *Pattern Recognition*, 2017, accepted.

[149] H.-L. Wei, S. A. Billings, Y. Zhao, and L. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio–temporal system identification," *Neural Networks, IEEE Transactions on*, vol. 20, no. 1, pp. 181–185, 2009.

[150] H.-L. Wei, D.-Q. Zhu, S. A. Billings, and M. A. Balikhin, "Forecasting the geomagnetic activity of the Dst index using multiscale radial basis function networks," *Advances in Space Research*, vol. 40, no. 12, pp. 1863–1870, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0273117707002086

[151] S. A. Billings, H.-L. Wei, and M. A. Balikhin, "Generalized multiscale radial basis function networks," *Neural Networks*, vol. 20, no. 10, pp. 1081–1094, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608007001876

[152] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimization methods and software*, vol. 1, no. 1, pp. 23–34, 1992.

[153] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[154] M. Lichman, "UCI Machine Learning Repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[155] WHO, "Breast cancer: prevention and control." [Online]. Available: http://www.who.int/cancer/detection/breastcancer/en/

[156] T. Wang, S.-U. Guan, K. L. Man, and T. O. Ting, "EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[157] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[158] S. Wing, J. R. Johnson, J. Jen, C.-I. Meng, D. G. Sibeck, K. Bechtold, J. Freeman, K. Costello, M. Balikhin, and K. Takahashi, "Kp forecast models," *Journal of Geophysical Research: Space Physics*, vol. 110, no. A4, 2005, a04203. [Online]. Available: http://dx.doi.org/10.1029/2004JA010500

[159] P. T. Newell, T. Sotirelis, K. Liou, C.-I. Meng, and F. J. Rich, "A nearly universal solar wind-magnetosphere coupling function inferred from 10 magnetospheric state variables," *Journal of Geophysical Research: Space Physics*, vol. 112, no. A1, pp. n/a–n/a, 2007, a01206. [Online]. Available: http://dx.doi.org/10.1029/2006JA012015

[160] H. A. Elliott, J.-M. Jahn, and D. J. McComas, "The Kp index and solar wind speed relationship: Insights for improving space weather forecasts," *Space Weather*, vol. 11, no. 6, pp. 339–349, 2013. [Online]. Available: http://dx.doi.org/10.1002/swe.20053

[161] T. Detman and J. Joselyn, “Real-time Kp predictions from ACE real time solar wind,” *AIP Conference Proceedings*, vol. 471, no. 1, pp. 729–732, 1999.

[162] F. Boberg, P. Wintoft, and H. Lundstedt, “Real time Kp predictions from solar wind data using neural networks,” *Physics and Chemistry of the Earth, Part C: Solar, Terrestrial & Planetary Science*, vol. 25, no. 4, pp. 275 – 280, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1464191700000167

[163] R. Bala and P. Reiff, “Improvements in short-term forecasting of geomagnetic activity,” *Space Weather*, vol. 10, no. 6, 2012, s06001. [Online]. Available: http://dx.doi.org/10.1029/2012SW000779

[164] Y. Liu, B. X. Luo, and S. Q. Liu, “Kp Forecast Models Based on Neural Networks,” *Manned Spaceflight*, vol. 27, no. 2, 2013.

[165] E.-Y. Ji, Y.-J. Moon, J. Park, J.-Y. Lee, and D.-H. Lee, “Comparison of neural network and support vector machine methods for Kp forecasting,” *Journal of Geophysical Research: Space Physics*, vol. 118, no. 8, pp. 5109–5117, 2013. [Online]. Available: http://dx.doi.org/10.1002/jgra.50500

[166] J. Wang, Q. Zhong, S. Liu, J. Miao, F. Liu, Z. Li, and W. Tang, “Statistical analysis and verification of 3-hourly geomagnetic activity probability predictions,” *Space Weather*, vol. 13, no. 12, pp. 831–852, 2015, 2015SW001251. [Online]. Available: http://dx.doi.org/10.1002/2015SW001251

[167] M. A. Balikhin, R. J. Boynton, S. N. Walker, J. E. Borovsky, S. A. Billings, and H. L. Wei, “Using the NARMAX approach to model the evolution of energetic electrons fluxes at geostationary orbit,” *Geophysical Research Letters*, vol. 38, no. 18, 2011, l18105. [Online]. Available: http://dx.doi.org/10.1029/2011GL048980

[168] R. J. Boynton, M. A. Balikhin, S. A. Billings, H. L. Wei, and N. Ganushkina, “Using the NARMAX OLS-ERR algorithm to obtain the most influential

coupling functions that affect the evolution of the magnetosphere," *Journal of Geophysical Research: Space Physics*, vol. 116, no. A5, 2011, a05218. [Online]. Available: http://dx.doi.org/10.1029/2010JA015505

[169] H.-L. Wei, S. A. Billings, and M. A. Balikhin, "Prediction of the Dst index using multiresolution wavelet models," *Journal of Geophysical Research: Space Physics*, vol. 109, no. A7, 2004.

[170] M. A. Balikhin, O. M. Boaghe, S. A. Billings, and H. S. C. K. Alleyne, "Terrestrial magnetosphere as a nonlinear resonator," *Geophysical Research Letters*, vol. 28, no. 6, pp. 1123–1126, 2001. [Online]. Available: http://dx.doi.org/10.1029/2000GL000112

[171] O. M. Boaghe, M. A. Balikhin, S. A. Billings, and H. Alleyne, "Identification of nonlinear processes in the magnetospheric dynamics and forecasting of Dst index," *Journal of Geophysical Research: Space Physics*, vol. 106, no. A12, pp. 30 047–30 066, 2001. [Online]. Available: http://dx.doi.org/10.1029/2000JA900162

[172] C. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2007.

[173] M. Nielsen, *Neural Networks and Deep Learning.* Determination Press, 2015.

[174] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.