# Accuracy Improvement of Temporal Series Prediction by Analysis of Memory Dependency Through Linear Regression Functions

# 線形回帰関数を用いたメモリ独立性の解析による時系列信号の予測精度の改善

by

Ricardo Moraes Muniz da Silva

A DISSERTATION
submitted to the Graduate School of Engineering,
Department of Computer Science & Engineering,
Nagoya Institute of Technology,
in a partial fulfillment of the requirements for the degree of
DOCTOR OF ENGINEERING
written under the supervision of
Professor Hidekata Hontani

NAGOYA INSTITUTE OF TECHNOLOGY
Nagoya, Japan
January 2021

This work was created using the LaTeX 2$_{\varepsilon}$ system, MiKTeXv2.9, together with the software WinEdt v10.3, TeXMaker 5.0.2, JabRef v5.1, MathType v6.5c, LaTable v0.7.2 and R Language 1.2.5019. The illustrations were created using R Language 1.2.5019 and Adobe Illustrator CS5.

# Abstract

Time series forecasting models are some of the most important type of quantitative models, in which past observations are collected and analyzes to describe their relationship. This modeling approach is particularly useful when little knowledge is available on the generation process or when there is no accurate model that relates the prediction variable to other explanatory variables. When modeling time series, two well-known methodologies are used for prediction in linear time series, the autoregressive integrated moving average (ARIMA) and the autoregressive fractionally integrated moving average (ARFIMA). The prediction of future events depends on the analysis of past values. In case of ARIMA it is limited to a short memory dependency, due to model characteristics on the modelling procedures with stationarity time series. This means that the statistical proprieties such as mean, variance and autocorrelation are all constant over time. However, depending on the dataset, the amount of past values necessary for an accurate prediction may vary, as the correlation between data and their parameter deviate along the time, leading to a long memory dependency. In these cases, the ARFIMA model can be used, since it provides a solution for the tendency of over-differentiation on stationary series that exhibit a long-run dependence. This uncertainty on the amount of necessary past values necessary make an accurate prediction defines the classical case of long and short memory dependency. When modeling a time series, the statistical proprieties can reveal characteristics on the dataset that demonstrates memory dependency that decays exponentially (ARIMA) or hyperbolically (ARFIMA). Thus, is not always clear if a process is stationary or what are the influence of the past samples on the future values and, consequently, which of the two models is the best choice for a given time series. As different datasets contain different characteristics, a common approach to improve the accuracy is the use of preprocessing techniques or the modification on a specific component of the aforementioned models in order to tackle the memory dependency problem. Nevertheless, such solutions are specific to certain datasets, in which different scenarios can generate different results in the performance and accuracy of the model. This research proposes a new methodology which combines the two models and their parameters in order to keep a low accumulative error. The proposed model is applied on a variety of time series datasets without resorting to model modifications or the use of preprocessing techniques. Twelve different datasets of different contexts were selected from the UCI Machine Learning repository, Brazilian stock market and the Time Series Data Library. The proposed method was compared with the original ARIMA and ARFIMA models, as well as wavelet techniques that also tackle the memory dependency problem. Thus, performed better accuracy in the majority of the datasets, except in multivariate biomedical time-series. In addition, the proposed model also presented better accuracy in regions with abrupt changes when compared with other models, being a promise alternative in applications requiring fast decision making based on data forecast. This indicates that the number of features in a dataset can have a direct influence in the correlation between values.

**Keywords**: *ARIMA, ARFIMA, combined methodology, time series forecasting, accumulative error, memory dependency*

# Acknowledgements

This doctoral thesis is part of the requirements for the degree of Doctor of Engineering in Computer Science and Engineering from Nagoya Institute of Technology. This concludes my path as a student, but not concludes my path in research and hopefully in the academia, of course, I continue to be a student for life in the pursuit of knowledge. During my paht I have many adventures with many different people that taught me different perspectives and for sure will be always remembered in my life journey.

I would like to begin by expressing my sincere gratitude to Professor Hidekata Hontani for accepting me at his laboratory as a PhD candidate, providing me all the guidance and support on which helped me to create and conclude this research. Thank you very much for your patience despite my learning curve in research and the Japanese language.

I am thankful to the MEXT - Ministry of Education, the Government of Japan, for being selected and supported through a scholarship for the doctoral course, without it would be impossible to come, study and graduate in Japan, this opportunity gave the change to change my life and to fulfil my dream to become a researcher and hopefully someday a professor. I should also mentioned my most regards for the Nagoya Institute of Technology for the time I could enjoy in this institution.

Next, I would like to thank Professor Taizo Umezaki, whom accepted me in his laboratory as a PhD student, I could learn several aspects of the academic and business life in Japanese society. Thank you for your support and patience, as well your laboratory staff for always guide me in the bureaucracy of university documentation and procedures considering my Japanese language skills.

I would like to thank Professor Ryo Taguchi and Professor Yoshihiko Nankaku for serving at the doctoral committee examination for this dissertation. Their feedback is enriched the improvement of this doctoral dissertation.

Next I would like also to thank Professor Akira Iwata, whom accepted me in his laboratory as a research student when I arrived in Japan, I could learn valuable lessons looking the Japanese perspective.

Also a special gratitude to Professor Mauricio Kugler, whom advised me in both academic and daily life in Japan. Since we first met in 2008, I always admire your strength, knowledge and courage to come to Japan and start a new life building your family. At that time and until now you are a reference for me, of the possibilities that a foreigner from a Latin country can do in the world.

A special thanks to my masters adviser Alessandro Zimmer, who taught me valuable life lessons from the moment we went to Japan in 2008 and through the master course, for sure my life since then has changed forever.

I appreciate all friends and colleagues in the Nagoya Institute of Technology, specially Jana Makovníková for tutoring me when I arrived in Japan, helping me with Japanese procedures and documentations and also my gratitude to Alisson Ninomiya for advises on Japanese language, life and society.

I am without words to say thank you to all my family, special my parents, for all the support they gave me through a good education, affection and always motivating me to follow the path I wanted to choose in my life. Without your support I have not come this far. Thanks to you I can day after day walk the path I choose for my life and follow my heart, I love you very much and I hope that what I have chosen to do with my life is a good way to thank you for all the dedication you have had for me. Also, I am grateful to my Aunt Nati and Uncle Milton, for the guidance and share of life experience.

To my friends, Sexteto Sinister, SWZ and currently BJA childhood friends for both in good times and in bad, they were always by my side, giving me support and helping. That's it and my brothers, I know that regardless of everything, whether time or distance, we will maintain our friendship and our brotherhood, always!

I thank God, this mysterious force that surrounds me and has always been with me, giving me strength and maintaining my faith, regardless of belief or religion. Thank you for always being with me illuminating and protecting me, on this path that I decided to take in my life.

I thank the life I have, that everything I am and will be, always maintaining my honesty and compassion, aiming to help people in the way that I can help through my research and projects. Always being a happy person.

I thank you very much my wife and love of my life, Yuri. You are an angel that fulfilled my heart with your warm energy. I can thank you enough for always be on my side. Love you.

Finally, I would like to say thank you very much to all the people that contributed in this journey, this part of the path of my life, but unfortunately I do not have enough space to write all their their names.

*You can discover more about a person in an hour of play*
*than in a year of conversation.*

Plato, 427 BCE

*You can't teach a man anything,*
*you can only help him find the answer inside him self.*

Galileo Galilei, 1564 CE

*There is nothing outside of yourself that can ever enable you to get better,*
*stronger, richer, quicker, or smarter.*
*Everything is within.*
*Everything exists.*
*Seek nothing outside of yourself.*

Miyamoto Musashi 1584 CE, Gorin No Sho

*Don't look away, all life is transitory, a dream;*
*And we all come together in the same place,*
*in the end of time.*
*If I don't see you again here...*
*I will see you in a little while...*
*in a place where no shadows fall.*

Delenn, 2259 CE

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | | |
|---|---|---|
| ACF | - | AutoCorrelation Function |
| ADF | - | Augmented Dickley-Fuller |
| AE | - | Accuracy Error |
| APE | - | Absolute Percentage Error |
| AR | - | AutoRegressive |
| ARIMA | - | Autoregressive Integrated Moving Average |
| ARFIMA | - | Autoregressive Fractional Moving Average |
| BW | - | Bayesian Wavelet |
| CEPEA-USP | - | Center of Advanced Studies on Applied Economics - University of São Paulo |
| CM | - | Combined Model |
| DFT | - | Discrete Fourier Transform |
| KPSS | - | Kwiatkowski-Phillips-Schmidt-Shin |
| LAD | - | Least Absolute Deviation |
| LSE | - | Least Square Estimate |
| LSTM | - | Long-Short Term Memory |
| MA | - | Moving Average |
| MAPE | - | Mean Absolute Percentage Error |
| MSE | - | Mean Squared Error |
| NAC | - | Number of detected Abrupt Changes |
| PACF | - | Partial AutoCorrelation Functions |
| PM | - | Persistence Model |
| PP | - | Phillips-Perron |
| RNN | - | Recurrent Neural Network |
| UCI | - | University of California, Irvine |
| WDC-SILSO | - | World Data Center - Sunspot Index and Long-term Solar Observations |
| WMA | - | Wavelet Multiresolution Analysis |

# CHAPTER 1

# Introduction

Temporal series prediction is one of the most important types of quantitative models in which past observations of variables are collected and analyzed to develop a model describing their underlying relationship [1]. This modeling approach is particularly useful when little knowledge is available on the generation process or when there is no satisfactory explanatory model that relates the prediction variable to other explanatory variables [2, 3].

Prediction procedures include different techniques and models. Moving averages (MA) techniques, random walks and trend models, exponential smoothing, state space modeling, multivariate methods, autoregressive (AR) models, co-integrated and casual models, method based on neural networks, fuzzy systems or data mining and rule-based techniques are typical models used in temporal series prediction or also commonly known as time series forecasting [4, 5, 6, 7, 8].

When modeling time series, one of the most important and widely used linear models is the Auto-Regressive Integrated Moving Average (ARIMA). The popularity of this model is due to its statistical properties as well as the well-known Box-Jenkins methodology [9] used in the model building process. When building an ARIMA based forecasting model, the basic steps are model identification, parameter estimation and model validation. The main advantage of ARIMA relies on its accuracy over a wide domain of datasets.

In any forecasting method, the prediction of future events depend on the analysis of past values on the time series. With ARIMA, this is limited to a short memory dependency. However, depending on the dataset, the amount of past values necessary for an accurate prediction may vary, as the correlation between data and their parameters deviate along the time. This uncertainty over the number of necessary past values is the classical case of long and short memory dependence [10, 11, 12].

In recent years, the study of long memory dependency or also commonly known as long term memory, has been receiving increasing attention of statisticians and mathematicians, as it can be found in many fields, such as hydrology [13], chemistry [14], physics [15], economic [16] and finances [17]. When modeling time series in the presence of long memory dependence, a common alternative is the Autoregressive Fractionally Integrated Moving Average (ARFIMA). This model, while based on the ARIMA, allows the differencing parameter to be set to non-integer values.

Another approach is to analyze the modelling procedures through a breakdown, i.e., a fragmentation of the mathematical and statistical theorems that compose the model [18, 19, 20].

These solutions, however, are only relevant to specific scenarios under certain conditions[21].

Currently, there is a fair amount of data available for most prediction problems, and a variety of available techniques are applied to different areas with their particular targets and objectives. The main question is then how much data is actually necessary for an accurate prediction. One common approach is the comparison of ARIMA and ARFIMA results, which will perform differently for each dataset, after which the best model is chosen [22, 23].

## 1.1 Motivation

The difficulties on estimating the amount of data necessary for an accurate prediction, as well as the need of different sets of parameters for each model, leads to the motivation to create a new mathematical model under a statistical learning framework. This model will combine some characteristics of the two aforementioned linear models in the hope of achieving higher prediction accuracy when compared to other alternative approaches.

## 1.2 Objectives

The main objective of this research is to develop a new method which combines the dynamics of long and short memory dependency in order to obtain accurate predictions in different scenarios. In each iteration of the modeling process, by observing the accumulative error, estimated residuals and number of lags, the proposed model's objective is to keep the error at a minimum, deviating from the decay characteristics observed in the ARIMA and ARFIMA models. The definition of a relationship between the exponential and hyperbolical decays of these models allows to reduce the accumulative error and, consequently, increase the prediction accuracy of the proposed method.

## 1.3 Expected Results

Different datasets can present significantly different statistical properties. Common approaches to deal with such variations include pre-processing techniques and model-modification. However, such techniques are specific to each dataset, and can generate unexpected results if blindly applied to different problems. The proposed method combines short and long memory dependencies characteristics, respectively from ARIMA and ARFIMA, in order to use enough information for and accurate prediction with minimal impact on the accumulative error.

It is important to notice that, as the proposed method is based on intrinsically linear models, it is expected to underperform in some datasets with high number of features. In such cases, the correlation between values decrease, hiding potential relationships between features that could help the modeling process to obtain a low accumulative error. In contract, some alternative methods specifically tackle the problem of high dimensionality, with partial accomplishment in some datasets, as it will be demonstrated in the experiments.

## 1.4 Overview

The remaining of this dissertation is organized as follows. Chapter 2 reviews the basic methods and groundwork used along this research, followed by Chapter 3, in which the proposed method is presented in details. Chapter 4 describes the datasets and conditions used to perform the experiments, while Chapter 5 shows experimental results and comparisons with the original basic methods and alternative approaches. Finally, Chapter 6 concludes this research and describes possibilities for future work.

# CHAPTER 2

# Literature Review

## 2.1 Temporal Series History

In general, forecasting methods analyze past values to predict future values. Until the early 1920s, forecasts were made with the simple extrapolation of a global value, adjusted according to time. An important contribution to the development of tools and methodologies to analyze time series is the work made by Yule [24], in which the author refers that a time series should be seen simply as the realization of a stochastic process. The author created the autoregressive model (AR), in which the predicted value depended on the past values. In the following years, experts in the field of statistics carried out their work and studies on the basis that the behavior of a series depended on linear models and noise.

However, linear models are insufficient for the analysis of time series, due to the fact that most of real series present strong tendencies of non-linearity [25]. To combat these difficulties, from 1950 onwards, new studies were started in the area of time series forecasting, and the exponential smoothing technique was created. It was also at this time, with the appearance of the computer, that a rapidly evolution in forecasting methods began [26].

A work that is considered a reference in the history of forecasting with time series is the book by Box Jenkins[9]. This book integrated all the existing knowledge to date in relation to time series, regarding parameter estimation, identification and forecasting. These authors approached the construction of stochastic models in the time domain for discrete time series, which were as simple as possible, with a minimum number of parameters and thus satisfying a parsimony criterion. The authors considered that the choice of this criterion was due to the fact that these models are able to reveal something about the nature of the processes that generate the time series and the possibility that these models are used to obtain optimal predictions of future series values[27].

## 2.2 Temporal Series Objectives

Initially, time series analysis had the objective to draw inferences about the basic properties or characteristics of the generating mechanism of the stochastic process of the series observations [28]. When analyzing a time series, it is expected that there is a cause related to time that influenced the data and that they may continue to influence them in the future. It is through the abstraction of regularities contained in the observable phenomena of a time series that the possibility exists to build a mathematical model as a simplified representation of reality [29]. The objectives of analyzing a time series are the following[30, 31]:

1. Description: determine the properties of a series, such as the trend pattern, the existence of structural changes and seasonality, etc. It is preferable to have the fundamental basics of the dataset before an attempt is made to build and model a time series, in order to describe some of its main properties. This will be a very important help in carrying out the modeling process;

2. Explanation / Modeling: find an appropriate statistical model that allows explaining the behavior of the series in the observed period;

3. Forecast: estimation of values that will occur in the Time Series, based on past events;

4. Process supervision: an accurate forecasts allows the statistician to take measures in order to supervise a given process.

Their are four main different application areas for temporal series and their models to be applied dynamically [9]: prediction of future events considering the previous and the current events; transfer function estimation of a system considering the input and output of the model; the use of variables as inputs to do evaluation of the impact in different parts of the temporal series; lastly, to design the control schemes whereby the potential deviations from the system's output values obtained from a desired objective can be adjusted by the values on the feed of the series.

For instance, as an example, in biology, a part that can be isolated to carry out a study on it, in which the internal and external influences at work are known, is called the system. Systems that evolve over time are especially classified as dynamic systems [32, 33]. A dynamic system, in general, can be under the action of $p$ input signals and reveal its behavior through $q$ output signals, in which the independent variable $t$ represents time. Thus, these input and output signals of a dynamic system can be seen as time series, that is, values of a certain quantity as a function of time [19, 34].

## 2.3 Types of Time Series

When a sequence of values are grouped together, considering a period of time, this is namely a time series which contains information related to an event. This data can be analyze with the purpose to find a trend or pattern that can assist on the prediction of future events. When performing this analysis, it is possible to optimize and better predict future actions [9, 35, 36]. Figure 2.1 shows basic methods of time series analysis.

### 2.3.1 Stochastic time series

A stochastic process can be described as a family of random variables $\{x_1, x_2, \ldots, x_t, t \in T\}$ indexed in time, where $T$ is the index. A stochastic time series is stationary if its mean, variance and autovariance (in different phases) remain the same. These characteristics are verified in any period in which the measurement is made, that is, they do not vary with time [25, 30, 31].

The statistical properties of a finite set of data in the series $\{x_1, x_2, \ldots, x_n\} \in X_t$ will always be similar to those of a set $\{x_{1+h}, x_{2+h} \ldots, x_{n+h}\} \in X_t$ for any integer value of $h$, which can happen in the following cases:

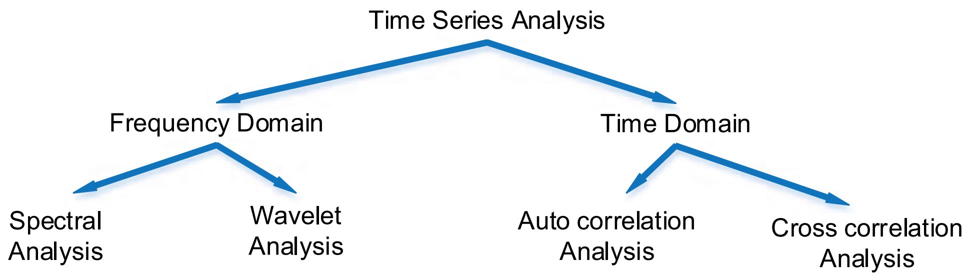


Figure 2.1: Classification of types of time series.

- When a group of variables are combine and hardly change, considering the transfer of variables over time (strict sense):

$$F(x_i, x_j, \ldots, x_k) = F(x_{i+h}, x_{j+h} \ldots, x_{k+h}) \tag{2.1}$$

- When the random values between the two points in the series are constant and rely only on the period of time (weak sense):

$$\begin{aligned} E(X_t) &= \mu \\ Var(X_t) &= \sigma^2 \\ \gamma(t, t-k) = \gamma(t, t+k) &= \gamma(k) \qquad \qquad where\ k = 0, \pm 1, \pm 2, \ldots \end{aligned} \tag{2.2}$$

In the literature there is also an alternative way of describing a stationary stochastic process in the frequency domain, through its function of spectral density, obtained by the Discrete Fourier Transform (DTF). Nevertheless, time series analysis applied to forecast usually does not rely on frequency analysis [2, 9].

Some examples of stationary stochastic processes:

- White noise process, ARIMA (0, 0, 0)

$$X_t = \varepsilon_t \tag{2.3}$$

- Homogeneous order process 1, ARIMA (1, 0, 0)

$$\nabla X_t = \varepsilon_t \tag{2.4}$$

- Autoregressive order process ($p$), ARIMA ($p$, 0, 0)

$$X_t = \rho_1 x_1 + \ldots + \rho_p x_{t-p} + \varepsilon_t \tag{2.5}$$

- Moving average order process ($q$), ARIMA (0, 0, $q$)

$$X_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \tag{2.6}$$

- ARMA process ($p$, 0, $q$)

$$X_t = \rho_1 x_{t-1} + \ldots + \rho_p x_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{2.7}$$

### 2.3.2 Non-stationary stochastic process

A stochastic time series is non-stationary when its mean and/or variance varies with time. In these series it is only possible to determine a part of the data, so they generate a worse forecasts than the stationary series [37]. The statistical properties of a finite set of data in the series $\{x_1, x_2, \ldots, x_n\} \in X_t$ will be different from those of a set $\{x_{1+h}, x_{2+h} \ldots, x_{n+h}\} \in X_t$ for at least one integer value of $h$. These series can be considered:

- Non-stationary time series on average, when the mean of the series is not stable over time, and may have an increasing or decreasing trend:

$$E(X_t) \neq \mu \tag{2.8}$$

- Non-stationary time series in covariance, when the variance or covariance varies with time:

$$\begin{aligned} &Var(X_t) \neq \sigma^2 \\ &\gamma(t, t-k) \neq \gamma(t, t+k) \end{aligned} \tag{2.9}$$

Some examples of non-stationary stochastic processes:

- Random walk, ARIMA (0, 1, 0):

$$(1-B)x_t = \varepsilon_t \tag{2.10}$$

- Simple Exponential Smoothing Process:

$$(1-B)x_t = (1-\theta_1 B)\varepsilon_t \tag{2.11}$$

- ARIMA integrated processes (p, d, q):

$$(1-\rho_1 B - \ldots - \rho_p B^p)(1-B)^d x_t = (1-\theta_1 B - \ldots - \theta_q B^q)\varepsilon_t \tag{2.12}$$

- Long memory processes:

$$(1-B)^d x_t = \varepsilon_t \qquad for - 0.5 < d < +0.5 \tag{2.13}$$

- ARFIMA process (p, d, q):

$$\rho_p(B)\nabla^d z_t = \theta_q(B)\varepsilon_t \qquad for - 0.5 < d < +0.5 \tag{2.14}$$

## 2.4 Trend and Seasonality of Time Series

A very important behavior is the trend, which can be defined as a long-term behavior of the time series. In this case, the series varies due to the constant gain (growth) or loss (decline), in successive periods of time [38]. It is important to notice that, sometimes, there are some cyclical variations and irregular fluctuations that should not be overlooked [39].

If we consider that the time series has no seasonal component, we can consider the model:

$$Z_t = T_t + \varepsilon_t \tag{2.15}$$

where $\varepsilon_t$ is a random variable with zero mean, variance $\sigma_\varepsilon^2$ and $T_t$ are the trend estimator. After estimating the trend $\hat{T}_t$ we can consider an adjusted series:

$$Y_t = Z_t - \hat{T}_t \tag{2.16}$$

Another process that is used to remove the trend is to make successive differences from the original series until a stationary series is found:

$$\nabla Z_t = Z_t - Z_{t-1} \tag{2.17}$$

One of the problems in verifying the trend of a time series is related to the need to know whether the data are independent, since most tests are based on this hypothesis. The most used tests to detect trends are the Wald-Wolfowitz tests, the Cox-Stuart test, the Mann-Kendall test and the Spearman test [40]. The perception of the trend depends, in part, on the duration of the observation of the time series. In addition, the decomposition of seasonal variation and trend components may not be independent [41].

Seasonal variation is a type of variation in which similar patterns of behavior are repeated in equal periods of time, usually during the year, but can also be weekly, monthly or quarterly periods. Examples of these seasonal variations are the mating and reproduction cycles of the animals and the temperature variations in a given region recorded annually or daily. Seasonality can be divided into two types, deterministic, when the seasonal pattern is regular and stable over time or stochastic, when the seasonal pattern varies over time [42, 43].

The tests most used to estimate seasonality are the Kruskal-Wallis test (for several independent samples) [44], Friedman test (indicated for several dependent or paired samples) [45] and the $F$ test, always taking care to remove, if any, the series trend [45]. If the time series has more than one non-random component, a proposal is to test the existence of one of them (trend or seasonality) after the elimination of the other component [30, 31].

## 2.5 Mathematical Methods of Forecasting Time Series

A forecasting method has a set of procedures that, based on a historical data, allows to predict what the expected behavior of the data will be, in the medium or long term. A forecast reflects a manifestation in relation to the unknown successes in a determined future [19]. Forecasting is not an end in itself, but a mean to provide information and contribute for consequent decision-making in order to achieve certain objectives [30, 31].

The time series forecasting methods, which are classified as quantitative, base their predictions on the extrapolation of characteristics of past observations and on the interrelationship between them, providing reliable predictions if the future shows behavior similar to the past. Most time series forecasting methods are based on the assumption that past observations contain all information about the time series pattern of behavior and that pattern recurs over time [46].

Forecasting methodologies can be divided in three types, judicial predictions on which are based on subjective judgment; univariate methods, which rely on the present and past values in a time function; multivariate methods, on which the predictors depends on a single or group of variables [47].

Given that there are countless methods for forecasting time series and is not always with the most complex methods that the best results are obtained, it will be necessary to first evaluate the benefits of the various methods, before starting a forecast of a certain time series [46].

In choosing the forecasting method that can give the best results, it is important to take into consideration some of error measurement approaches:

- Graphical comparison;
- Scatter diagram;
- Correlation coefficient;
- Accumulated quadratic error;
- Root of the mean square error (RMSE);
- Absolute percentage error (MAPE);

### 2.5.1 Simple forecasting methods

Many studies involving time series use simple forecasting methods to forecast future events of the time series by observing past events of the series under study. The objective of these methods is to identify a basic pattern present in the documented dataset of the temporal series and consequently to predict future values. Given that these univariate statistical methods are simple to implement, their overall computing cost is relatively small and they provide a reasonably accurate forecast, these methods are widely used [30, 31].

#### Persistence Model

The Persistence Model (PM) is perhaps the simplest possible forecasting model available. It simply uses the sample $x_{t-1}$ to predict $x_{t+1}$ [30].

#### Simple Moving Average

The simple moving average (not to be confused with the Moving Average model presented in Section 2.7.2) is defined as a models that uses an average of recorded observations, as a forecast for a certain period in the future. In these models, moving averages can be simple, centered or weighted. The term moving average is used because as an observation becomes available, the average of the observations is recalculated using this value and ignoring the oldest one [30, 31]. For models that use the simple moving average, an equation of the type can be defined:

$$X_t = \frac{x_{t-1}, x_{t-2}, \ldots, x_{t-n}}{n} \tag{2.18}$$

where $n$ (observation window) is the number of observations and $X_t$ is the average. In these models the number of observations is proportional to the smoothing effect on the forecast. Larger observation windows should be used for series that present a lot of randomness or small changes in their patterns, so that the calculation of the moving average is immune to noise and short movements. Conversely, smaller observation window should be used for series that show little random fluctuation in the data or significant changes, in order to react more quickly to these changes.

### 2.5.2 Exponential smoothing

The exponential smoothing models are divided into:

- Simple Exponential Smoothing Models;
- Linear Exponential Smoothing Models;
- Holt-Winters Models with and without seasonality.

### Simple Exponential Smoothing Model

Like the moving average models, the straightening models use the recorded observations to determine the behavior of the series. However, while in the simple moving average all the observations used have an equal weight in the calculation of the forecast, in simple smoothing each observation of the series has a different weight. In this model, the value of the last observations is evidenced by the application of a smoothing constant $\alpha$, in the forecast at the previous moment. The formula for calculating exponential smoothing is given by the equation:

$$F_{t+1} = \alpha x_t + (1 - \alpha)F_t \tag{2.19}$$

where:

$F_{t+1}$ represents the forecast at time $t + 1$;

$\alpha(0 < \alpha < 1)$ (smoothing constant) is the weight attributed to the observation;

$x_t$ is the newest observed event;

$F_t$ is the last predicted event.

For small $\alpha$ values, the weight of the initial observations is greater in predicting the behavior of the series, while for larger $\alpha$ values (close to 1), the forecast will be mostly affected by the recent observations.

### Linear Exponential Smoothing Model

This method was developed to recognize the presence of a trend in the time series, thus improving the forecast value, which will be given by the following equation:

$$F_{t+n} = S_t + mT_t \tag{2.20}$$

where:

$S_t$ designates the forecast at time $t$ and is given by:

$$S_t = \alpha x_t + (1 - \alpha)(S_{t-1} + T_{t-1}); \tag{2.21}$$

$T_t$ represents the trend component and is given by:

$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta)(T_{t-1}); \tag{2.22}$$

$\alpha$ and $\beta$ are smoothing coefficients.

### Exponential Smoothing Method Holt-Winter with Seasonality

This forecasting method is similar to the linear exponential smoothing method, however it is capable of analysing time series that show seasonal trends. The forecast value is given by:

$$\begin{aligned}
F_{t+m} &= (S_t + mT_t)l_{t-1+m} \\
S_t &= \alpha \frac{x_t}{l_{t-1}} + (1 - \alpha)(S_{t-1} + T_{t-1}) \\
T_t &= \beta(S_t - S_{t-1}) + (1 - \beta)(T_{t-1}) \\
l_t &= \gamma \frac{x_t}{S_t} + (1 - \gamma)l_{t-1}
\end{aligned} \tag{2.23}$$

where:

$l_t$ is the smoothing of the seasonality factor $\frac{x_t}{S_t}$;

$l$ is the seasonality interval;

$\gamma$ weight attributed to the seasonality factor.

## 2.6 Time Series Analysis

Before the models described in Section 2.7 can be applied, several tests must be performed on the time series in order to determine the order of the models and other parameters.

### 2.6.1 Normality test

There are several tests used to find out whether data from a temporal series contains a normal distribution. For instance, Shapiro-Wilk, Doornik-Hansen, Jarque-Bera and Lilliefors tests are available [48].

### 2.6.2 Unit root test or stationarity test

A time series will be stationary if its properties do not change over time. The analysis of the time series graph may give an idea of the behavior of the data, but it is insufficient to verify whether it will be stationary. There are several tests to do this study, most of which are based on finding a unit root. According to the literature, more than one unit root test should be used in order to certify the stationarity of the series. Some of the most used tests are the Augmented Dickley-Fuller tests, Phillips-Perron test and KPSS test [49].

**Augmented Dickley-Fuller**

This test is also known as the ADF (Augmented Dickley - Fuller) test and is based on the following description:

$$^{\nabla}y_t = \beta_1 + \beta_2 t + \delta y_{y-1} + \sum_{i=1}^{m} \alpha_i \nabla y_{t-1} + \varepsilon_t \tag{2.24}$$

where:

$\beta_1$ is the series intercept, also known as series *drit*;

$\beta_1$ is the trend coefficient;

$\delta$ is the unit root presence coefficient;

$m$ is the series lags number;

In this test, the following hypotheses are considered:

- $H_0 : \delta = 0$ A unit root is contained in the temporal series.
- $H_1 : \delta \neq 0$ The series does not contain stationarity.

A regression of $\nabla y_t$ will be made in $y_{t-1}$, $\nabla y_{t-1}$,..., $\nabla y_{t+p-1}$ and then the $T$ statistic will be calculated, which will be given by:

$$T = \frac{\hat{\delta}}{se(\hat{\delta})} \tag{2.25}$$

where:

$\hat{\delta}$ is the estimator for $\delta$;

$se(\hat{\delta})$ is the estimator considering a standard deviation on the error of $\delta$.

The critical values of the $T$ statistic are calculated by Dickley and Fuller through Monte Carlo simulation. [49, 50]

**Phillips-Perron**

This test is known in the literature by the acronym PP and is a generalization of the Dickley-Fuller test for cases in which the errors $\{\varepsilon_t\}, t \in Z$ are correlated and, possibly, heteroscedastic, therefore considering the same hypotheses, based on the study of the following description:

$$\nabla y_t = \beta_1 + \beta_2 t + \delta y_{y-1} + \sum_{i=1}^{m} \alpha_i \nabla y_{t-1} + \varepsilon_t \tag{2.26}$$

Particularly in this test, the $Z$ statistic is calculated by:

$$Z = n\hat{\delta}_n - \frac{n^2 \hat{\sigma}^2}{2 s_n^2} (\hat{\lambda}_n^2 - \hat{\gamma}_{0,n}) \tag{2.27}$$

Being:

$$\hat{\gamma}_{0,n} = \frac{1}{n} \sum_{i=1+j}^{n} r_i r_{i-1} \tag{2.28}$$

$$\hat{\lambda}_n^2 = \hat{\gamma}_{0,n} + 2 \sum_{j=1}^{q} \left(1 - \frac{j}{q+1}\right) \hat{\gamma}_{j,n} \tag{2.29}$$

$$s_n^2 = \frac{1}{n-k} \sum_{i=1}^{n} r_i^2 \tag{2.30}$$

where:

$r_i$ is represented by the residue in $y_i$ using least squares estimators;

$k$ is the number of covariates in the regression;

$q$ is the number of lags used to calculate $\hat{\lambda}_n^2$.

It should be noted that $Z$ is an adjustment of the Dickley-Fuller statistic and that if the process is not correlated it will have null covariance and therefore $\hat{\lambda}_n^2 = \hat{\gamma}_{0,n}$. On the other hand, if the process is not heteroscedastic it will be $se(\delta) = \frac{1}{n}$ and $Z$ will be given by:

$$Z = n\hat{\delta} = \frac{\hat{\delta}}{se(\hat{\delta})} \tag{2.31}$$

So $Z$ becomes Dickley-Fuller's statistic and therefore will have the same distribution as the ADF test statistic.

### KPSS

This test aims to determine stationarity in a time series and was created by Denis Kwiatkowski, Peter C. B. Phillips, Peter Schmidt and Yongcheol Shin. Comparing with the previous tests (ADF and PP), the hypotheses to determine stationarity are exchanged, that is:

- $H_0$ = The series is non-stationary;
- $H_1$ = The series has a unit root.

Considering $Y_t : t \in N$, the observations of a time series, it will be decomposed into components of deterministic tendency $\xi_t$, random walk $r_t$ and stationary error $\varepsilon_t$ being:

$$Y_t = \xi_t + r_t + \varepsilon_t \tag{2.32}$$

where: $\mu_t$ represents the fluctuations of the random walk.

Considering $e_t, t \in N$ the residuals of a y regression is explained by the trend, random walk and intercept components. The estimator for the variance of the regression errors is given by:

$$\hat{\sigma}_\varepsilon^2 = \frac{SQE}{E} \tag{2.33}$$

The partial sum of the residuals is defined by: $S_t = \sum\limits_{i=1}^{t} e_t, t = 1, 2, ..., T$ the statistics of this test are defined by:

$$LM = \sum_{t=1}^{N} \frac{S_t^2}{N^2 \hat{\sigma}_\varepsilon^2} \tag{2.34}$$

## 2.7 Reference Methods

In this section an analysis will be made to the Box-Jenkins methodology [9], for later comparison with the modeling done with ARFIMA.

### 2.7.1 Autoregressive (AR)

In the Autoregressive (AR) model, past values of a time series are used to predict a future value. Values at different instants of time contributes differently to this calculation. If only the immediately proceeding value is used, it is said to be an AR(1) process. More generally, an AR($p$) corresponds to a process that uses the last $p$ values, where $p$ is know as the lag. The AR($p$) model is described as:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{2.35}$$

where $\varphi_1, \ldots, \varphi_p$ are the model parameters, $\varepsilon_t$ is the error and $c$ is a constant.

### 2.7.2 Moving Average (MA)

In this model, instead of the temporal series values being directly used, the past error terms are the parameterized values used for calculating a prediction. Similarly to the AR model, MA($q$) represents the moving average or order $q$, i.e., the last $q$ error terms are used. The MA($q$) can be mathematically expressed as:

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{2.36}$$

where $\mu$ is the mean of the time series, $\theta_1, \ldots, \theta_q$ are the model parameters and $\varepsilon_{t-1} + \cdots + \varepsilon_{t-q}$ are the past values.

### 2.7.3 ARIMA

When analyzing a temporal series assuming their statistical linearity, an ARIMA (Auto-Regressive Integrated Moving Average) model can be used for predictions on temporal series that are not stationary but can be made so by differentiation. The autoregressive terms correspond to lagged series values and the moving averages terms correspond to lagged random errors, while the differentiation is the integration between the two models [51, 52].

The model is based on a linear combination of past values (AR components) and errors (MA components) [53, 9]. The ARIMA predicted value $x_t$ is given by:

$$\begin{aligned} x_t &= A(L)^d \left(1 + B\left(L\right)\varepsilon_t\right) \\ A\left(L\right) &= 1 - \rho_1 L - \rho_2 L^2 - \ldots - \rho_p L^p \\ B\left(L\right) &= 1 + \theta_1 L + \theta_2 L^2 + \ldots + \theta_q L^q \end{aligned} \tag{2.37}$$

where $d$ is the order of differentiation, $\varepsilon_t$ is the error, $L$ is the lag operator, $\rho_p$ are the parameters of the AR terms on the polynomial of order $p$ and $\theta_q$ indicate the parameters of the MA terms on the polynomial of order $q$. Such model is referenced as an ($p$, $d$, $q$) ARIMA.

This model is intended to be used with stationary time series, i.e. time series with statistical properties that are constant over time [9]. The stationarity of a time series can be evaluated by different statistical tools, such as the Mean Absolute Percentage Error (MAPE), given by:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{x}_i - x_i}{x_i}\right| \times 100\% \tag{2.38}$$

where $N$ is the number of predicted values and $\hat{x}_i$ and $x_i$ are, respectively, the $i^{th}$ predicted and actual values.

There are two phases to the identification of an appropriate ARIMA model: (1) changing the data into a stationary time series and (2) determining the tentative model by observing the behavior of the autocorrelation function (ACF) and partial autocorrelation functions (PACF) [9].

The autocorrelation coefficient $r_k$ measures the correlation between a set of observations and a lagged set of observations in a time series:

$$r_k = \frac{\sum\limits_{t=1}^{n} \left(x_t - \bar{x}\right)\left(x_{t+k} - \bar{x}\right)}{\sum\limits_{t=1}^{n} \left(x_t - \bar{x}\right)^2} \tag{2.39}$$

where $x_t$ is the sample of the stationary time series at time $t$ and $\bar{x}$ is the mean of the stationary time series. Box and Jenkins suggest the number of pairs used to calculate the autocorrelation to be no more than $n = 4$. The sample autocorrelation coefficient $r_k$ is an initial estimate of $\rho_p$ from Eq. (2.37) [9].

The objective of the partial autocorrelation analysis is to measure how $x_t$ and $x_{t+k}$ are related. The PACF is used as a guide, along with the estimated ACF, for choosing one or more ARIMA models that might fit the available data. The partial autocorrelation $\varphi_{k,k}$ is given by:

$$\varphi_{k,k} = \frac{r_k - \sum\limits_{j=1}^{k-1} \varphi_{k-1,j} r_{k-j}}{1 - \sum\limits_{j=1}^{k-1} \varphi_{k-1,j} r_j} \tag{2.40}$$

where:

$$\begin{aligned} \varphi_{k,j} &= r_{k-1,j} - \varphi_{k,k}\varphi_{k-1,k-j} \\ \varphi_{1,1} &= r_1 \end{aligned}$$

In this model, a nonstationary time series is differentiated $d$ times until it becomes stationary, where $d$ is an integer. Such a series is said to be integrated of order $d$, denoted $I(d)$, with the non-differentiated $I(0)$ being the option for stationary series. Is important to notice that many series exhibit too much dependence to be $I(0)$, but are not $I(1)$ [54].

When considering a seasonal ARIMA, the predicted values are evaluated using additional statistical tools, including the Mean Squared Error (MSE) [27], given by:

$$\text{MSE} = \frac{1}{N+1} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 \tag{2.41}$$

The minimal value of equation (2.41) corresponds to the best seasonal model.

It must be noted that, in many cases, it is not always clear what order of differentiation is necessary for a series to become stationary, as well as the range of dependency between values. For these cases, the ARFIMA model can be used, as it can also be applied to nonstationary time series [49, 28].

### 2.7.4 ARFIMA

The ARFIMA is one of the most popular models for problems with long-memory dependency [49]. It provides a solution for the tendency to over-differentiate stationary series that exhibit long-run dependence, allowing a continuum of the fractional differencing parameter $-0.5 < d < +0.5$ [55, 56].

The ARFIMA model is described as follow:

$$\begin{aligned} A(L)(1-L)^d x_t &= B(L)\varepsilon_t \\ (1-L)^d &= \sum_{k=1}^{\infty} \frac{(k-d)L^k}{(-d)(k+1)} \end{aligned} \tag{2.42}$$

The stochastic process $x_t$ is both stationary and invertible if all the roots of $A(L)$ and $B(L)$ present $|d| < 0.5$. An important characteristic of an ARFIMA process is its autocorrelation function's hyperbolic decay rate, which contrasts with the exponential decay rate observed in the ARIMA model [57].

This characteristic is visible when the ACF estimation results in a stable distribution with a negative differentiate parameter [58]. This shows that the dependence in the long memory can occur in diverse circumstances, impacting values in arbitrary distant events that still strongly influence each other [59, 60]. Consequently, predicted ARFIMA values with normal distribution, for the differentiate parameter $d > 0$, have a slow autocovariance function decay that is not absolute summable. This behavior corresponds to the classical definition of long range dependency [2].

Inverting Eq. (2.42), it is possible to evaluate the decay of the ACF in more details:

$$x_t = (1 - L)^{-d}(A\,(L))^{-1}B\,(L)\,\varepsilon_t \tag{2.43}$$

Following the estimated parameter procedure, the temporary results are acquire by putting $d = 0$ in Eq. (2.43), and describe the behavior of the fractionally differenced process $(1 - L)^d x_t$. The long memory dependency is observed by using the estimated value of $d$ from Eq. (2.42), and demonstrate a fractionally integrated $x_t$.

**Spectral Density**

The spectral density of ARFIMA models with $d > 0$ are finite except at null frequency, while ARIMA models have their spectral density finite at all frequencies. Generally, the two models have different spectral densities for frequencies close to zero when $d > 0$ [26, 61].

The ARFIMA model can capture both low-frequency and high-frequency components in the spectral density [62]. In contrast, the ARIMA model cannot differentiate between long-run and short-run effects. Thus, it fails to capture long-range dependencies and its spectral density, although remaining finite, is pulled upward [63, 64].

This assumption is useful when evaluating the memory parameter $d$, providing information on correlations in a time series measured at different time scales. When $d = 0$, the changes in the values of the time series are uncorrelated with each other. When $d > 0$, increases in the values of a time series are more likely to be followed by increases, and, conversely, decreases are more likely to be followed by decreases. Such a time series is called persistent and it has long-memory property. Finally, when $d < 0$, increases in the values of the time series are likely to be followed by decreases and, conversely, decreases are more likely to be followed by increases. Such a time series is called antipersistent or short memory dependent [65]. Antipersistence has been observed in financial time series for electricity price processes [66], in climatology [67] and particularly in nanoscale biophysics [68, 69]. An antipersistence time series reverses itself more often than a stochastic series would [65]. For the ARFIMA model, the influence of a given value on the series decreases with the inverse of the square root of its distance on time. However, although non-stationary long memory time-series can be properly modeled, non-stationary short memory dependencies can result in an incorrect parameter estimation [2, 70, 71].

In the literature, different methods of assessing short and long range dependence and estimating the memory parameter have been developed, as briefly explained in the following section.

## 2.8 Alternative Approaches

Several authors proposed improvements through a breakdown on the AR components of the ARIMA model, in order to increase the accuracy of the parametrization process [72]. Although this approach can follow the variations in the parametrization of the model, it fails when the lag of a time series changes along the time. The variations in the required lag of a time-series can be addressed by the use of statistical tools in order to decompose and evaluate the model. This can reduce instabilities in the forecast and improve the accuracy of the ARIMA model.

However, such procedure is highly dependent on the quality of the statistical tools used in the decomposition process [73, 74].

The use of preprocessing techniques on the datasets had also been reported to improve forecasting accuracy. For instance, making the mean and variance constant and not depending on time was used to improve ARIMA results. However, in the case of ARFIMA, such preprocessing removes important long-term dependencies necessary for an accurate prediction [75, 76].

When applying such techniques, it becomes more difficult to do a parameter identification prior to the model estimation, resulting in a higher accumulative error when applied to datasets with diverse characteristics [77, 78, 79]. An alternative approach to avoid these problems is the use of wavelet analysis, which can be applied to standard models such as ARIMA and ARFIMA without resorting to fragmentation or preprocessing.

### 2.8.1 Wavelet Multiresolution Analysis (WMA)

Wavelet Multiresolution Analysis (WMA) is a technique used to smooth the input values until identifying the overall long-term trend. The fluctuations of the wavelet components can be accurately modeled with ARIMA models [80]. Such combination is quite flexible, presenting good prediction results for several different types of time-series. The major limitation of the method is the assumption of short memory dependency.

### 2.8.2 Bayesian Wavelet (BW)

In the case of long-memory dependency time-series, the Bayesian Wavelet (BW) method can be used to detect multiple changes in the estimation order of an ARFIMA model. However, the process of fitting the model to a given dataset can be a major issue depending on the short-term characteristics of the data [81].

## 2.9 Linear Regression

The relation between two variables in a straight line can be mathematically defined as a linear regression. In this description, when one variable is the dependent one and all others are independent, this is defined as a simple linear regression. If there is more than one independent variable, it is defined as a multiple linear regression. However, a multiple linear regression differs from a multi-variate linear regression, in which there are multiple dependent variables for forecast, instead of a single one [82, 83].

These methods are built using linear predictor functions, which are the parameters used to estimate values from the dataset [82, 84]. In regression analysis, the objective is to measure the normal distribution of the results, which are given by the predictors. This means that the independent variables' values are related conditionally to the average of the result of the dependent variables [85, 82]. Generally, a linear regression can be in the following two forms.

In the case of forecast accuracy improvement, a linear regression can be applied to a prediction model to fit the observed data from the dependent and independent variables. Subsequently to the creation of the model, new values of the independent variable can be added without the dependent variable, on which the already constructed model can forecast a new result.

For analysis of the fluctuation of the dependent variable (which can be a characteristic of the independent variable), the linear regression can be used to measure the connection between them. Particularly, it can be used to analyze if an independent variable does not have linear connection with the dependent variable, or to find if there is some redundant information of the dependent variable in the subgroup of the independent variable [85, 86].

Usually, linear regression models are suitable to be used with the least square method, but also can be uses with other methods, such as the least absolute deviation or the least square minimizing function [86, 87].

Considering data in the form of $\{y_i, x_{i1}, ..., x_{iy}\}_{i=1}^{n}$ of $n$ real numbers, the connection between the dependent variable $y$ and the $p$-vector of regressors $x$ is defined as linear, describing a linear regression model. This connection is built through an unobserved and random error variable $\varepsilon$ [88, 83]. This is mathematically defined as:

$$y_i = \beta_0 + \beta_1 x_{i1} +, ..., + \beta_p x_{ip} + \varepsilon_i = x_i^T \beta + \varepsilon_i \tag{2.44}$$

where $i = 1, ..., n$ and $T$ denotes the transpose. The matrix form of the $n$ equations can be written as:

$$y = X\beta + \varepsilon \tag{2.45}$$

where:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \ X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & ... & x_{1p} \\ 1 & x_{21} & ... & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & ... & x_{np} \end{pmatrix}, \ \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \ \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

The interpretation of this equation is as follows. The term $y$ is the predicted dependent value, while the matrix $X$ contains the independent variables $x_i$ in each $i^{th}$ row. From these, one row might be constant (e.g. $x_{i0} = 1$), in which the $X$ is $p + 1$-dimensional, and the corresponding $\beta_0$ is called the intercept term. $\beta_i$ are the regression coefficients, and $\varepsilon_i$ are the errors [85, 87]. The error $\varepsilon$ is used as the minimization loss when calculating the optimal value of $\beta$, minimizing the error of the independent variable $y$ in relation to the input values $X$.

# CHAPTER 3

# Proposed Method

This dissertation proposes a new method which combines the ARIMA and ARFIMA models, taking into account both long and short memory characteristics of the input data in order to minimize the accumulative error over time. This allows the proposed method to be applied to several different scenarios, as it will be demonstrated in the experiments on Chapter 5.

In order to combine the characteristics of both methods, a new linear regression function was created. This function joins the error and residual values of the two methods in order to model its own predictor function, regression and error. The function focus on reducing the accumulated error and maximizing the autocorrelation of the time-series values relevant to the current prediction. Thus, the model is dynamically adapted to the data's memory dependence.

This is done by selecting the model with lowest error for each new prediction. As each model uses different amounts of past data to be able to forecast the newest data, the model selection process indirectly adjusts the amount of past information used by the proposed method. This results in an improved prediction accuracy.

The main diagram of the proposed method is shown in Fig. 3.1. It starts by evaluating the statistical tools over a given dataset, followed by the independent parametrization of both ARIMA and ARFIMA standard models. The percentage error of both models is measured and the parameters of the model with the smallest error are selected. These parameters are then used by the combined model (CM) to generate a predicted value and the process is repeated. The combination of the short and long term memory characteristics of both models helps to limit the increase of the accumulative error.

Considering a time series $\mathbf{X}$ given by:

$$\mathbf{X}^T = x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_t, \ldots \tag{3.1}$$

where $x_1, x_2, \ldots$ are past values, $x_n$ is the current value, $x_{n+1}, \ldots$ are future values and $x_t$ is a sample at a given time $t$.

In order to distinguish between the predicted values of ARIMA, ARFIMA and the proposed method, $\mathbf{X}$ will be renamed, respectively, $\mathbf{X}^G$, $\mathbf{X}^F$ and $\mathbf{X}^C$, where the values $x_1, \ldots, x_n$ are the same for all three models. The design matrix $\mathbf{M}$ is then given by:

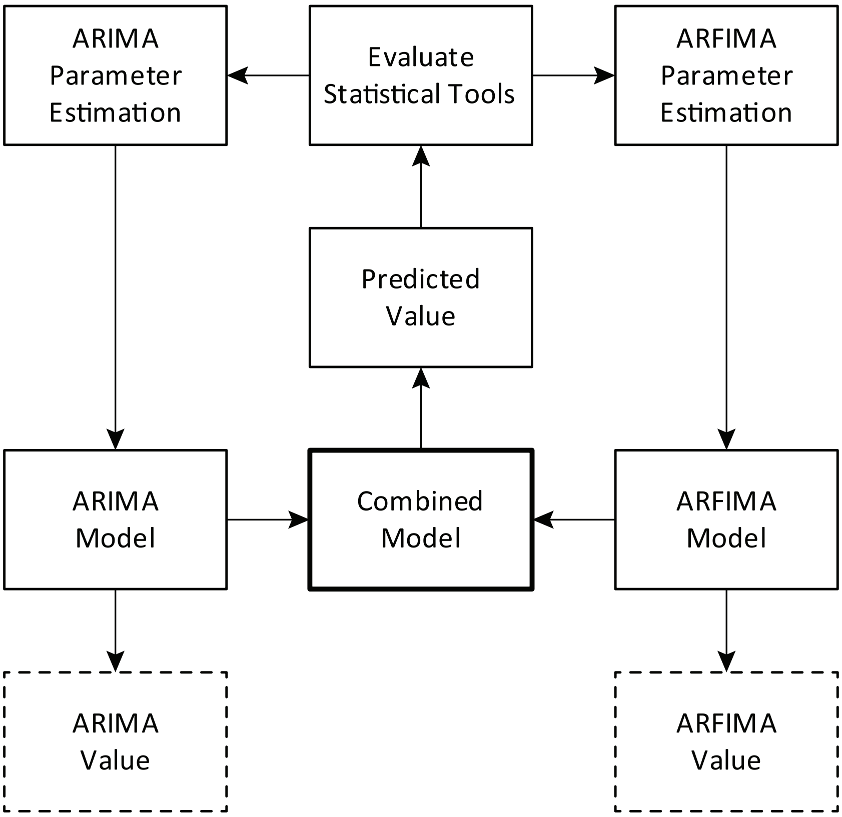

ARIMA Parameter Estimation
Evaluate Statistical Tools
ARFIMA Parameter Estimation
Predicted Value
ARIMA Model
Combined Model
ARFIMA Model
ARIMA Value
ARFIMA Value

Figure 3.1: Proposed method block diagram, combining the ARIMA and ARFIMA models.

$$\mathbf{M} = \begin{bmatrix} x_1^G & x_1^F & x_1^C \\ x_2^G & x_2^F & x_2^C \\ \vdots & \vdots & \vdots \\ x_n^G & x_n^F & x_n^C \\ x_{n+1}^G & x_{n+1}^F & x_{n+1}^C \\ \vdots & \vdots & \vdots \\ x_t^G & x_t^F & x_t^C \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{3.2}$$

The first predicted value calculated by the CM model is based on the values of $x_t^G$ and $x_t^F$, as well as the error $\varepsilon_t$ from equations (2.37) and (2.42), renamed $\varepsilon_t^G$ and $\varepsilon_t^F$, respectively.

The prediction $x_t^*$ of the CM model is given by:

$$x_t^* = B_t^T A + e_t \tag{3.3}$$

where $A$ is the regression coefficient vector, $e_t$ is the mean-zero error on the stationary process and $B_t^T$ is given by:

$$B_t^T = \sum_{j=t-w}^{t-1} \beta_j \varepsilon_j^C \tag{3.4}$$

where $\beta_t$ is the MA at a given time $t$, $w$ is the MA interval and the error $\varepsilon_t^C$ is defined as:

$$\varepsilon_t^C = \min\left(\varepsilon_t^G, \varepsilon_t^F\right) \tag{3.5}$$

There is a variety of statistical tools that can be used to analyse a dataset, depending on the memory dependency scale in use, in order to achieve an accurate prediction. This research uses the Least Square Estimate (LSE), as it presents an optimal regression analysis for the probabilistic distribution in linear models [83]. The LSE is given by:

$$\hat{A}_{lse} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{X} \tag{3.6}$$

When the error $\varepsilon_t$ is heavy-tailed, it is more desirable to use robust estimation procedures such as the Least Absolute Deviation (LAD), given by:

$$\hat{A}_{lad} = \arg\min_{A} \left| \sum_{i=1}^{t} x_i - \mathbf{X}^T A \right| \tag{3.7}$$

In both the LSE and LAD cases, the estimated residuals can be written as:

$$\begin{aligned} e_t^{lse} &= x_t - \mathbf{X}^T \hat{A}_{lse} \\ e_t^{lad} &= x_t - \mathbf{X}^T \hat{A}_{lad} \end{aligned} \tag{3.8}$$

The value of $e_t$ on Eq. (3.3) is based on the selection between $e_t^{lse}$ and $e_t^{lad}$, depending on the characteristics of the distribution of $\varepsilon_t^C$.

In order not to exceed the lag operators of the ARIMA and ARFIMA models (respectively $L^G$ and $L^F$), the lag of the CM model is defined as $L^C = \min\left(L^G, L^F\right)$. It is then possible to validate the CM model through a log loss function, defined as:

$$LF\left(x_t^*, y_t\right) = \left(1 - y_t\right) \ln\left(1 - x_t^*\right) - y_t \ln x_t^* \tag{3.9}$$

where:

$$y_t = \begin{cases} 1 & \text{if} \quad t \leqslant L^C \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

This function defines a relationship between equations (3.4) and (3.8), combining the exponential and hyperbolical decay characteristics of ARIMA and ARFIMA models in order to keep a low accumulative error. The minimal accumulative error point $k$ is then calculated by:

$$k = \max_t \left[ \tau_t \geqslant \sum_{i=n+1}^{n+L^C} x_i^* \tau_i \right] \tag{3.11}$$

where $\tau_t$ is the likelihood of convergence in the CM model, given by:

$$\tau_t = \exp\left(LF\left(x_t^*, y_t\right)\right) \tag{3.12}$$

Finally, the predicted value $x_{n+1}^C$ is given by:

$$x_{n+1}^C = \frac{x_k^* \tau_k}{\sum\limits_{i=n+1}^{n+L^C} x_i^* \tau_i} \tag{3.13}$$

The parametrization of the obtained ARIMA, ARFIMA, respectively $P^G = \left\{p^G, d^G, q^G\right\}$ and $P^F = \left\{p^F, d^F, q^F\right\}$ are then updated for the next iteration. On the first iteration, the parameters of the CM model, $P^C = \left\{p^C, d^C, q^C\right\}$ receive the parameters of the selected model from (3.5), and afterwards changed according to (3.3), based on the polynomial order of (2.37).

The aforementioned steps can be summarized in the following algorithm:

1. Calculate $x^G_{n+1}$ and $x^F_{n+1}$ using (2.37) and (2.42);
2. Calculate $\varepsilon^C_{n+1}$ using (3.5);
3. Calculate $e_t$ using (3.6) to (3.8);
4. Calculate $x^*_{n+i}$ $(i = 1 \ldots L^C)$ using (3.3);
5. Validate model using (3.9) to (3.12);
6. Calculate $x^C_{n+1}$ using (3.13);
7. Update $P^G$, $P^F$ & $P^C$;
8. Increment $n$ and repeat.

It must be noticed that some statistical tools might not work properly when dealing with both short and long memory dependency, which would reduce the accuracy of the proposed method.

# CHAPTER 4

## Experimental Framework

In order to demonstrate the efficiency of the proposed algorithm, several datasets of different contexts were used in the experiments, shown in Table 4.1. All the datasets were obtained from the UCI Machine Learning Repository [89], except the *Sugar Price Database* (dataset 1), obtained from CEPEA-USP [90], *Sunspot Number* (dataset 6), obtained from WDC-SILSO [91], and the *Nile River* (dataset 9), obtained from the Forecast Package for *R* [92]. Considering the multivariate characteristics of the datasets and their application in univariate models described in this paper, a single feature was selected or calculated from each dataset, as described in 4.2.

Each dataset was split into training and test sets, with each set containing 50% of the available samples. The training data was used to generate some tentative models, which were compared between each other in order to find the one with the smallest error. The best fit parameters for each model was determined by the analysis of the statistical tools described in Chapter 2. The trained models were then applied to the unseen test datasets in order to obtain the actual forecast results. All experimental results correspond to a one-step ahead forecast.

Table 4.1: Time series datasets.

| **Name** | **Dataset** | **Samples** |
|---|---|---|
| Sugar Price Database | 1 | 3346 |
| Greenhouse Gas Observing Network | 2 | 327 |
| Electricity Load Diagrams | 3 | 140256 |
| Individual Household Electric Power Consumption | 4 | 2075259 |
| Combined Cycle Power Plant | 5 | 9568 |
| Sunspot Number | 6 | 3253 |
| Istanbul Stock Exchange | 7 | 536 |
| Dow Jones Index | 8 | 750 |
| Nile River | 9 | 1299 |
| Epileptic Seizure Recognition | 10 | 4097 |
| Basic Hand Movements | 11 | 3000 |
| Arrhythmia | 12 | 452 |

Table 4.2 describes the selected feature of each of the datasets used in the experiments in Section 5.

Table 4.2: Time series dataset selected feature.

| **Dataset** | **Selected Feature** |
|---|---|
| 1 | Price of 50 kg bag of crystal sugar (daily close value) |
| 2 | Average of all regions |
| 3 | Consumption of client MT-166 |
| 4 | Global active power |
| 5 | Net hourly electrical energy output (EP) |
| 6 | Monthly mean of total sunspot number [Jan. 1749 – Dec. 2019] |
| 7 | Istanbul Stock return value |
| 8 | Close value of the stock price at end of week |
| 9 | Annual minimum level of Nile River, 622 – 1921 |
| 10 | Response variable of a single selected individual (first subject) |
| 11 | Holding of spherical tools by a single female individual (first subject) |
| 12 | Sum of areas of all segments (QRSA) |

Table 4.3 describes the B3 Ibovespa Stock Market, which contains the value of 50kg sugar bag from November 2003 to May 2009 [90].

Table 4.3: Sugar Price Database.

| Dataset Characteristics | Time Series |
|---|---|
| Number of Instances | 3346 |
| Number of Attributes | 3 |
| Associated Task | Classification / Regression |
| Area | Business |

Table 4.4 describes the concentration of greenhouse gas (GHG) emissions in California. This dataset is divided over the period May 10th to July 31th, 2010 [89, 93].

Table 4.4: Greenhouse Gas Observing Network.

| Dataset Characteristics | Multivariate, Time Series |
|---|---|
| Number of Instances | 2921 |
| Number of Attributes | 5232 |
| Associated Task | Regression |
| Area | Physical |

Table 4.5 describes the electrical consumption from 370 points per clients from 2011 to 2014 period in Portugal [89].

Table 4.5: Electricity Load Diagrams.

| Dataset Characteristics | Time Series |
|---|---|
| Number of Instances | 370 |
| Number of Attributes | 140256 |
| Associated Task | Regression / Clustering |
| Area | Computer |

Table 4.6 describes a set of measures of electrical power consumption in one house considering a every minute measure over a period of around 4 years [89].

Table 4.6: Individual Household Power Consumption.

| Dataset Characteristics | Multivariate, Time Series |
|---|---|
| Number of Instances | 2075259 |
| Number of Attributes | 9 |
| Associated Task | Regression / Clustering |
| Area | Physical |

Table 4.7 describes a datase on which contains several events gathered from a combined cycle power plant during a 6 years period [89, 94, 95].

Table 4.7: Combined Cycle Power Plant.

| Dataset Characteristics | Multivariate |
|---|---|
| Number of Instances | 9568 |
| Number of Attributes | 4 |
| Associated Task | Regression |
| Area | Computer |

Table 4.8 describes the average month of sunspot number considering there total measurement [91].

Table 4.8: Sunspot Number.

| Dataset Characteristics | Time Series, Multivariate |
|---|---|
| Number of Instances | 3253 |
| Number of Attributes | 5 |
| Associated Task | Regression |
| Area | Physical |

Table 4.9 describes in details the Istanbul Stock Exchange with other international indexes as described$(SP, DAX, FTSE, NIKKEI, BOVESPA, MSCE-EU, MSCI-EM)$, from June 2009 to February 2011 [89, 6].

Table 4.9: Istanbul Stock Exchange.

| Dataset Characteristics | Multivariate, Time Series |
|---|---|
| Number of Instances | 536 |
| Number of Attributes | 8 |
| Associated Task | Classification / Regression |
| Area | Business |

Table 4.10 describes the Dow Jones Industrial Index. As a reminder: assume a daily data collection. When Monday is over all of the data for that day is collected. However it is not possible to invest on Monday, because the data collection completes only at the end of the day. It is possible to use the data from Monday to invest on Tuesday [89, 8]

Table 4.10: Dow Jones Index.

| Dataset Characteristics | Time Series |
|---|---|
| Number of Instances | 750 |
| Number of Attributes | 16 |
| Associated Task | Classification / Clustering |
| Area | Business |

Table 4.11: Nile River Annual Flow.

| Dataset Characteristics | Time Series |
|---|---|
| Number of Instances | 1299 |
| Number of Attributes | 2 |
| Associated Task | Regression |
| Area | Physical |

Table 4.11 describes the flow measurements annually on the Nile river at Aswan city (formerly Assuan), from 1871 to 1970 [92, 96].

Table 4.12 describes a group of datasets which contains EEG signals [89, 97].

Table 4.12: Epileptic Seizure Recognition.

| Dataset Characteristics | Multivariate, Time Series |
|---|---|
| Number of Instances | 4097 |
| Number of Attributes | 179 |
| Associated Task | Classification, Clustering |
| Area | Life |

Table 4.13 contains data of sEMG signals applied for basic hand movements [89, 98].

Table 4.13: sEMG for Basic Hand Movements.

| Dataset Characteristics | Time Series |
|---|---|
| Number of Instances | 3000 |
| Number of Attributes | 144 |
| Associated Task | Classification |
| Area | Life |

Table 4.14 contains a selection of different classes of cardiac arrhythmia [89, 99].

Table 4.14: Arrhythmia.

| Dataset Characteristics | Multivariate, Time Series |
|---|---|
| Number of Instances | 452 |
| Number of Attributes | 279 |
| Associated Task | Classification |
| Area | Life |

As shown in Tables 4.3 – 4.14, the datasets previously described shown several different instances and attributes, also from a variety of fields of application. On Chapter 5, the experimental results and comparison were made between the models described on the Chapters 2 and 3.

# CHAPTER 5

# Experiments

The experiments demonstrate how the proposed method better handles the memory dependency in most datasets, resulting in more accurate forecast when compared with the traditional ARIMA and ARFIMA models. For each of the compared methods, the best fit parameters, resulting in the smallest accumulative average error per window of time, as well as the respective training error, were obtained for each dataset and are presented in Table 5.1.

As previously mentioned, some datasets present lower average absolute error with ARIMA, while others perform better with ARFIMA, due to their long or short memory dependency characteristics. As shown in Table 5.2, when compared to both methods, the proposed model presents a lower testing error for most of the datasets, with the exception of datasets 10, 11 & 12. Datasets 1 to 9 present an average error reduction of 3.27%, while datasets 10 to 12 present an average error increase of 9.43%, when compared with ARIMA and ARFIMA.

Table 5.1: Best fit parameters ($p$, $d$, $q$) and training error (MAPE) for each model.

| **Dataset** | **ARIMA** | | **ARFIMA** | | **CM** | |
|---|---|---|---|---|---|---|
| 1 | (1,1,3) | 31.67% | (2,0.23,4) | 32.23% | (1,0.43,2) | 28.48% |
| 2 | (1,0,2) | 18.47% | (2,0.17,1) | 17.26% | (1,0.2,3) | 16.82% |
| 3 | (2,1,1) | 9.43% | (3,0.5,2) | 10.87% | (2,1,2) | 8.21% |
| 4 | (4,0,1) | 15.64% | (1,-0.34,5) | 17.12% | (3,0.2,4) | 12.23% |
| 5 | (3,1,5) | 21.05% | (1,0.47,3) | 19.22% | (2,0.15,3) | 18.55% |
| 6 | (1,1,6) | 9.68% | (2,0.33,1) | 10.42% | (1,1,4) | 8.75% |
| 7 | (2,0,1) | 29.12% | (3,0.29,5) | 28.41% | (2,-0.3,3) | 27.63% |
| 8 | (3,0,5) | 30.05% | (4,-0.42,3) | 29.85% | (2,1,4) | 28.79% |
| 9 | (1,1,4) | 20.45% | (1,0.45,2) | 17.23% | (1,0.36,3) | 16.86% |
| 10 | (2,1,3) | 54.65% | (2,-0.5,3) | 46.79% | (4,-0.3,3) | 62.47% |
| 11 | (4,1,2) | 68.41% | (3,0.34,2) | 57.61% | (2,0.5,4) | 74.52% |
| 12 | (3,1,4) | 67.23% | (2,-0.3,3) | 58.12% | (3,0.4,4) | 73.67% |

Table 5.2: Testing error (MAPE) for ARIMA, ARFIMA, WMA, BW, CM & PM models.

| Dataset | ARIMA | ARFIMA | WMA | BW | CM | PM |
|---|---|---|---|---|---|---|
| 1 | 33.57% | 35.21% | 32.68% | 31.52% | **29.56%** | 38.85% |
| 2 | 21.13% | 19.97% | 17.53% | 18.21% | **17.05%** | 27.84% |
| 3 | 12.56% | 13.89% | 15.35% | 14.64% | **9.35%** | 19.23% |
| 4 | 18.84% | 19.17% | 18.47% | 16.85% | **14.71%** | 25.61% |
| 5 | 22.77% | 21.93% | 21.65% | 22.13% | **19.07%** | 26.43% |
| 6 | 10.81% | 12.05% | 9.34% | 9.12% | **8.98%** | 13.58% |
| 7 | 31.23% | 30.11% | 31.05% | 29.74% | **28.16%** | 39.72% |
| 8 | 32.19% | 31.08% | 30.88% | 31.49% | **28.93%** | 38.89% |
| 9 | 21.67% | 17.54% | 19.15% | 18.37% | **17.11%** | 24.21% |
| 10 | 58.71% | 49.63% | 51.87% | **43.54%** | 64.28% | 70.41% |
| 11 | 72.15% | 60.48% | 64.68% | **54.98%** | 76.32% | 81.34% |
| 12 | 71.79% | 63.16% | 65.11% | **56.84%** | 75.64% | 79.87% |

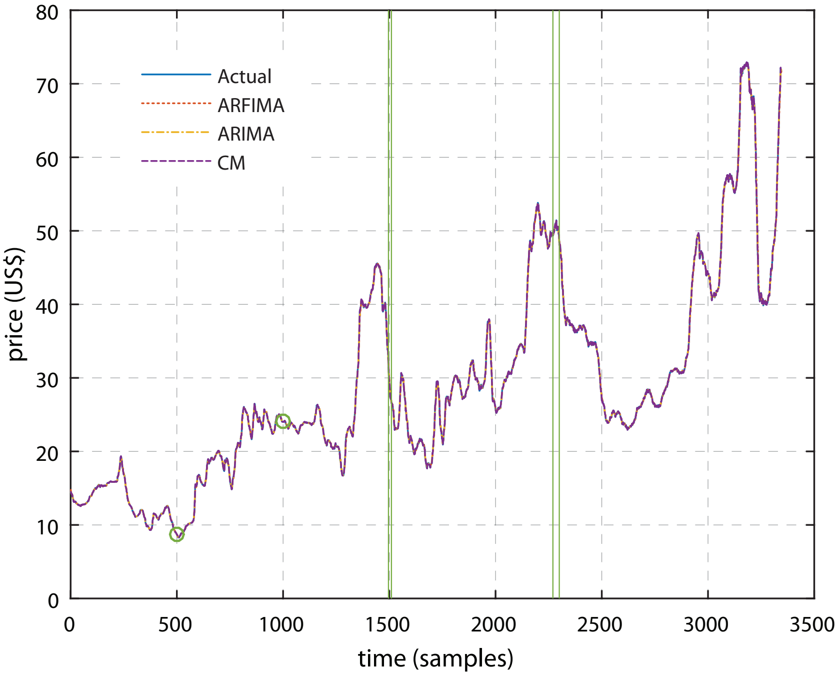


Figure 5.1: Comparison of the predicted values of ARIMA, ARFIMA and CM methods for the Sugar Price Database. The vertical lines indicate two regions emphasized in Figures 5.2 and 5.3. The marked points at times 500 and 1000 are used on the analysis of the proposed method in Figures 5.4 and 5.5.

## 5.1 Sugar Price Database

A detailed comparison of the three methods is presented for the Sugar Price Database (dataset 1). Figure 5.1 shows an overall view of the predicted values along the time. Two regions were selected to emphasize the differences in the predicted values, marked with vertical lines in Fig. 5.1 around 1500 and 2880, shown in Fig. 5.2 and 5.3, respectively. Also, points marked at samples 500 and 1000 were used on the analysis of the proposed method in Fig. 5.4 and 5.5.

Figure 5.2 shows an example in which the ARFIMA model fails to accurately predict the

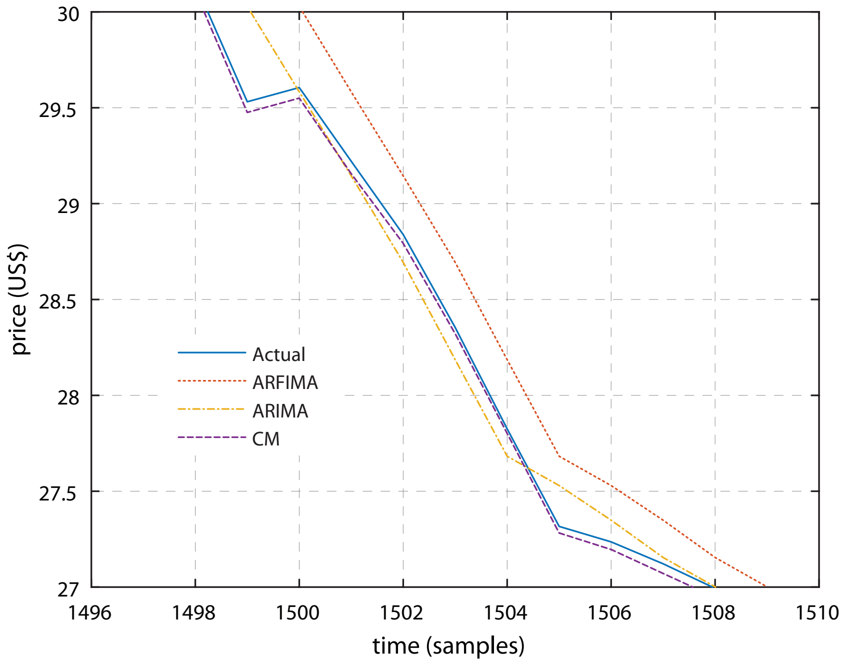


Figure 5.2: Detailed view of the region around sample 1500 from Fig. 5.1. The ARFIMA models fails due to the over-differentiated trend in the short-run.

future values. The reason for this is the low fluctuation of the values, which causes ARFIMA to over-differentiate the trend, thus failing to foreseen the correct value in the short run.

Figure 5.3 shows an area of prediction where the ARIMA model fails to accurately predict the future values. The reason is the long term fluctuations in the actual values, which have an inconsistent trend in a fractional level, preventing ARIMA to maintain a low level of accumulative error.

In both cases, the CM method presents a lower accumulative error during the course of the prediction of future values. Figure 5.4 demonstrates the CM model details (sample 500 in Fig. 5.1) working with several statistical tools in order to choose the lowest derivative value for the AR part, keeping it close to zero and consequently a lower accumulative error.

Figure 5.5 shows the different decay functions (sample 1000 in Fig. 5.1), from which the slowest one closest to zero is chosen. This keeps a low MA derivative value that can properly fit the structure of the model during the prediction of the next values.

## 5.2 Combined Cycle Power Plant

An addition detailed comparison of the three methods is presented for dataset 5. This dataset contains 9568 data points of collected hourly from a Combined Cycle Power Plant between 2006 & 2011, during which plant was working with full load [89]. Figure 5.6 shows an overall view of the predicted values along the time. Two regions were selected to emphasize the differences in the predicted values, marked with vertical lines in Fig. 5.6 at 70–80 and 130–140, shown in Fig. 5.7 and 5.8, respectively. Also, points marked at samples 23 and 116 were used on the analysis of the proposed method in Fig. 5.9 and 5.10.

Figure 5.7 shows an example in which the ARFIMA model fails to accurately predict the future values. The reason for this is the low fluctuation of the values, which causes ARFIMA to over-differentiate the trend, thus failing to foreseen the correct value in the short run.

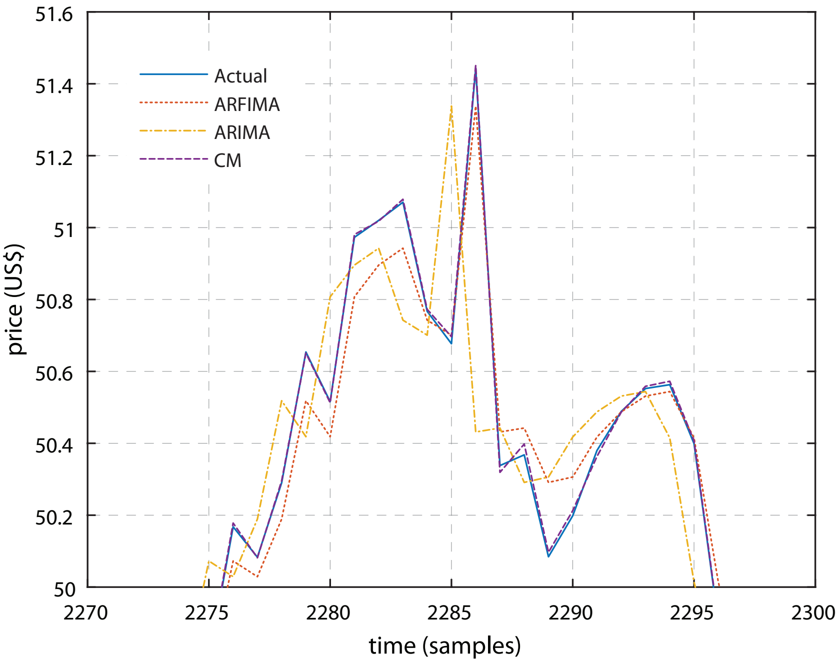


Figure 5.3: Detailed view of the region around sample 2280 from Fig. 5.1. The ARIMA model fails to predict future values due to the inconsistent trend in a fractional level.

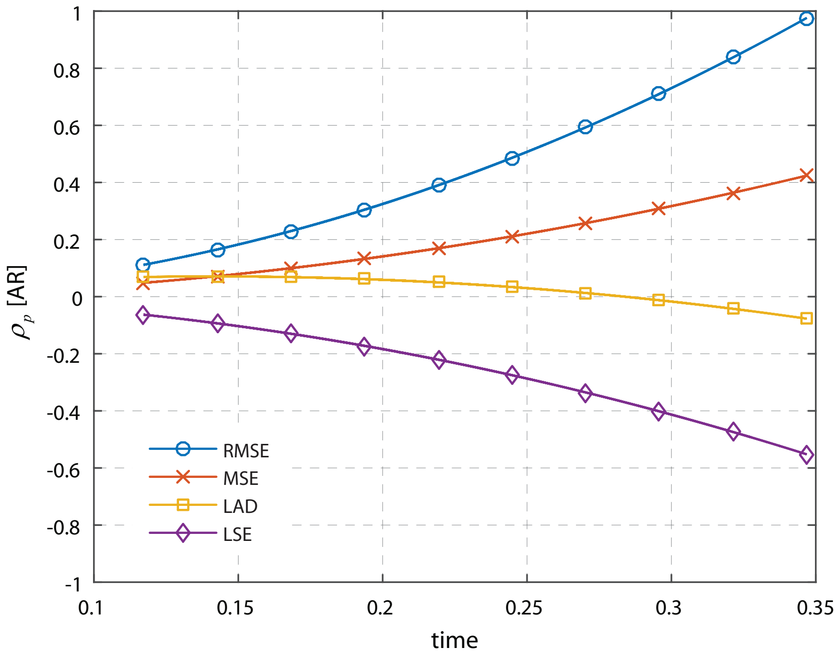


Figure 5.4: Derivative value corresponding to several statistical tools used in the CM method (sample 500 in Fig. 5.1). Choosing the value closest to zero helps to keep a low accumulative error.

Figure 5.8 shows an area of prediction where the ARIMA model fails to accurately predict the future values. The reason is the long term fluctuations in the actual values, which have an inconsistent trend in a fractional level, preventing ARIMA to maintain a low level of accumulative error.

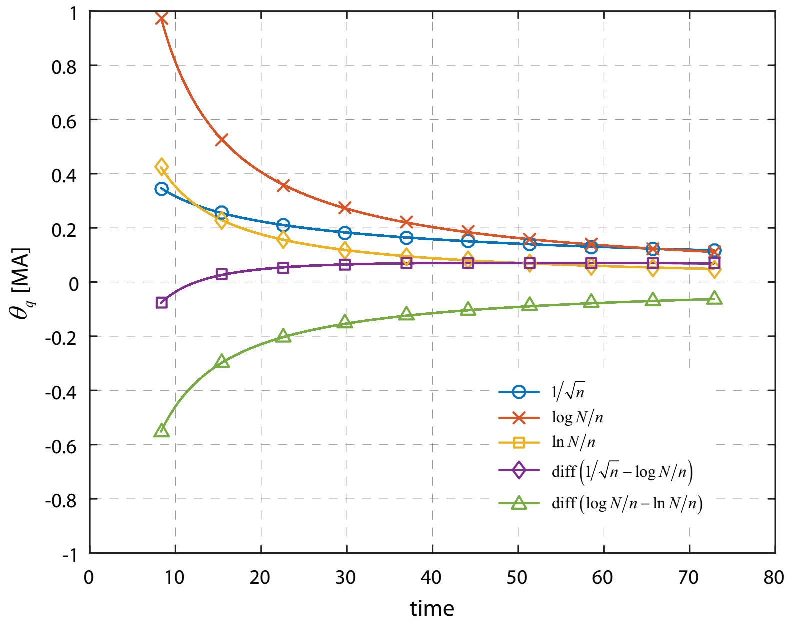


Figure 5.5: Selection of decay functions (sample 1000 in Fig. 5.1). Choosing the slowest one that is closest to zero helps to keep a low MA derivative value.

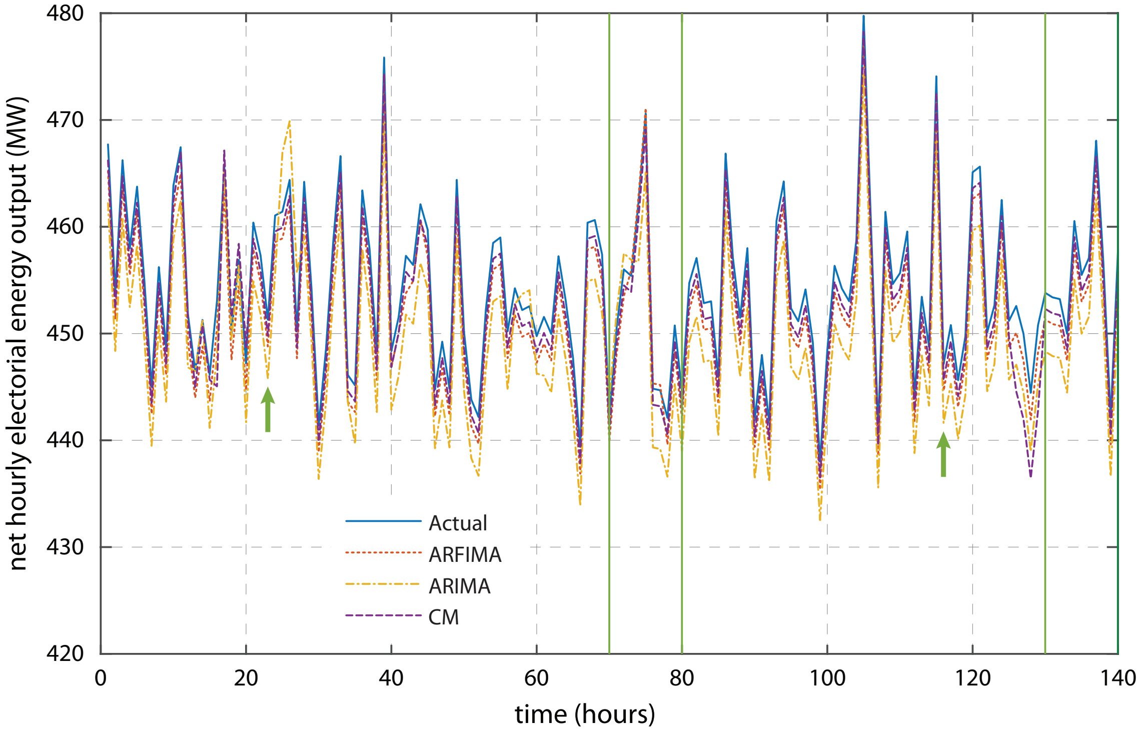


Figure 5.6: Comparison of the predicted values of ARIMA, ARFIMA and CM methods for the Combined Cycle Power Plant (dataset 5). The vertical lines indicate two regions emphasized in Figures 5.7 and 5.8. The marked points (arrows) at times 23 and 116 are used on the analysis of the proposed method in Figures 5.9 and 5.10.

In both cases, the CM method presents a lower accumulative error during the course of the

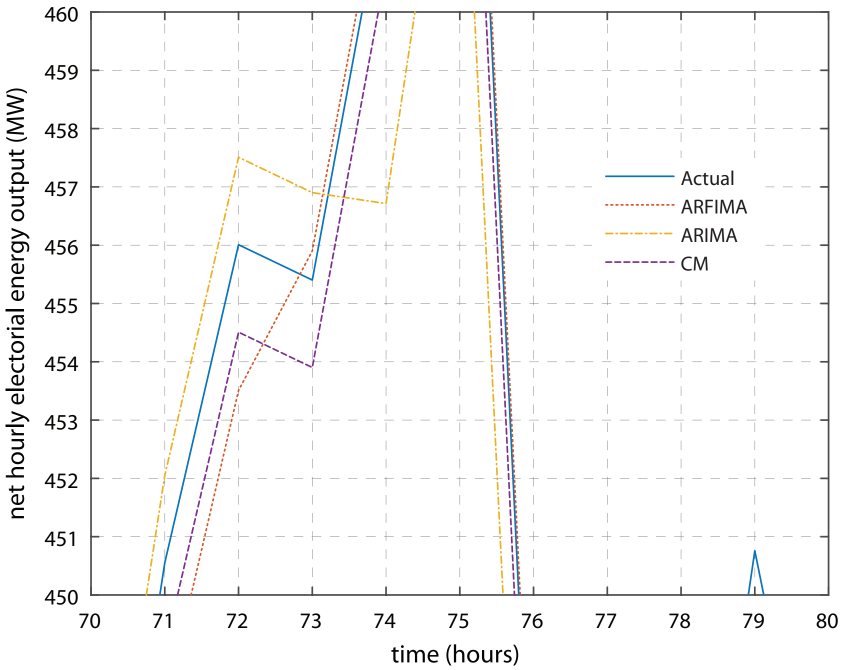


Figure 5.7: Detailed view of the region at times 70–80 from Fig. 5.6. The ARFIMA models fails due to the over-differentiated trend in the short-run.

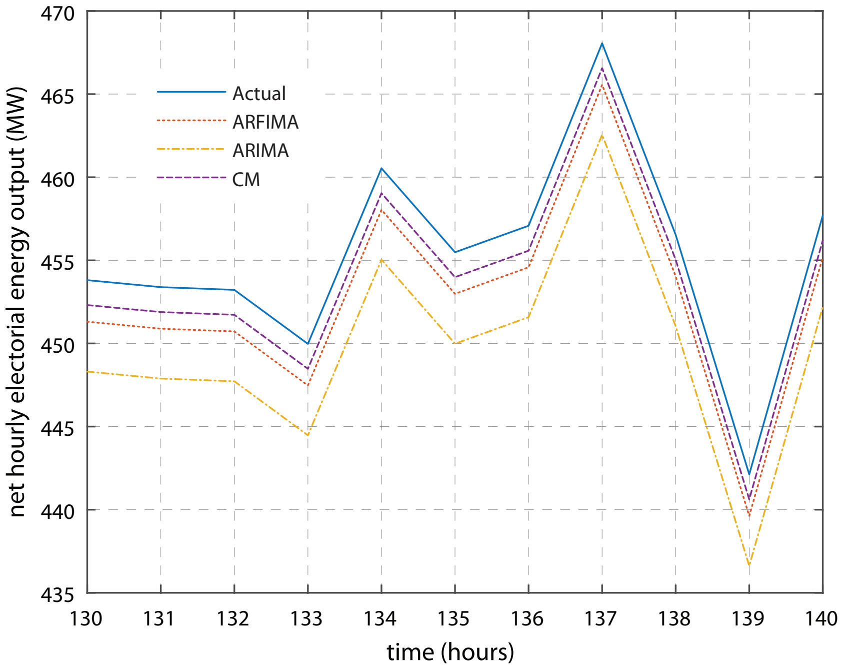


Figure 5.8: Detailed view of the region at times 130–140 from Fig. 5.6. The ARIMA model fails to predict future values due to the inconsistent trend in a fractional level.

prediction of future values. Figure 5.9 demonstrates the CM model details (sample 23 in Fig. 5.6) working with several statistical tools in order to choose the lowest derivative value for the AR part, keeping it close to zero and consequently a lower accumulative error.

Figure 5.10 shows the different decay functions (sample 116 in Fig. 5.6), from which the

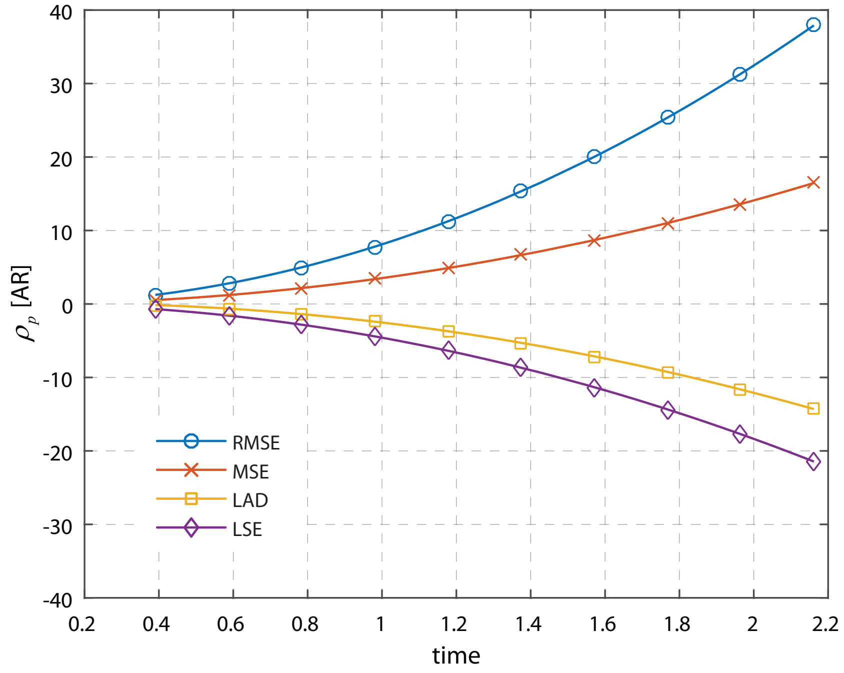


Figure 5.9: Derivative value corresponding to several statistical tools used in the CM method (sample 23 in Fig. 5.6). Choosing the value closest to zero helps to keep a low accumulative error.

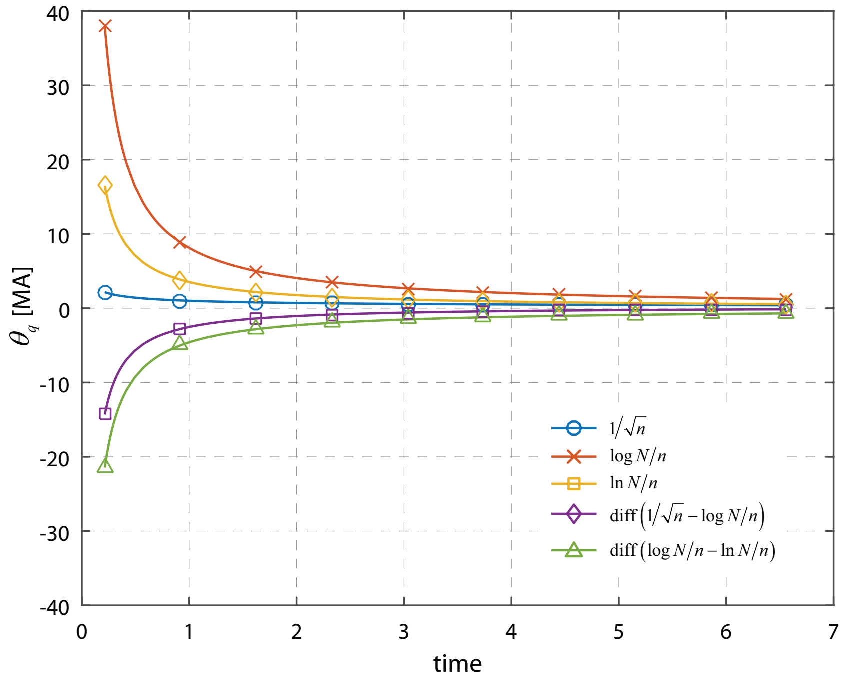


Figure 5.10: Selection of decay functions (sample 116 in Fig. 5.6). Choosing the slowest one that is closest to zero helps to keep a low MA derivative value.

slowest one closest to zero is chosen. This keeps a low MA derivative value that can properly fit the structure of the model during the prediction of the next values.

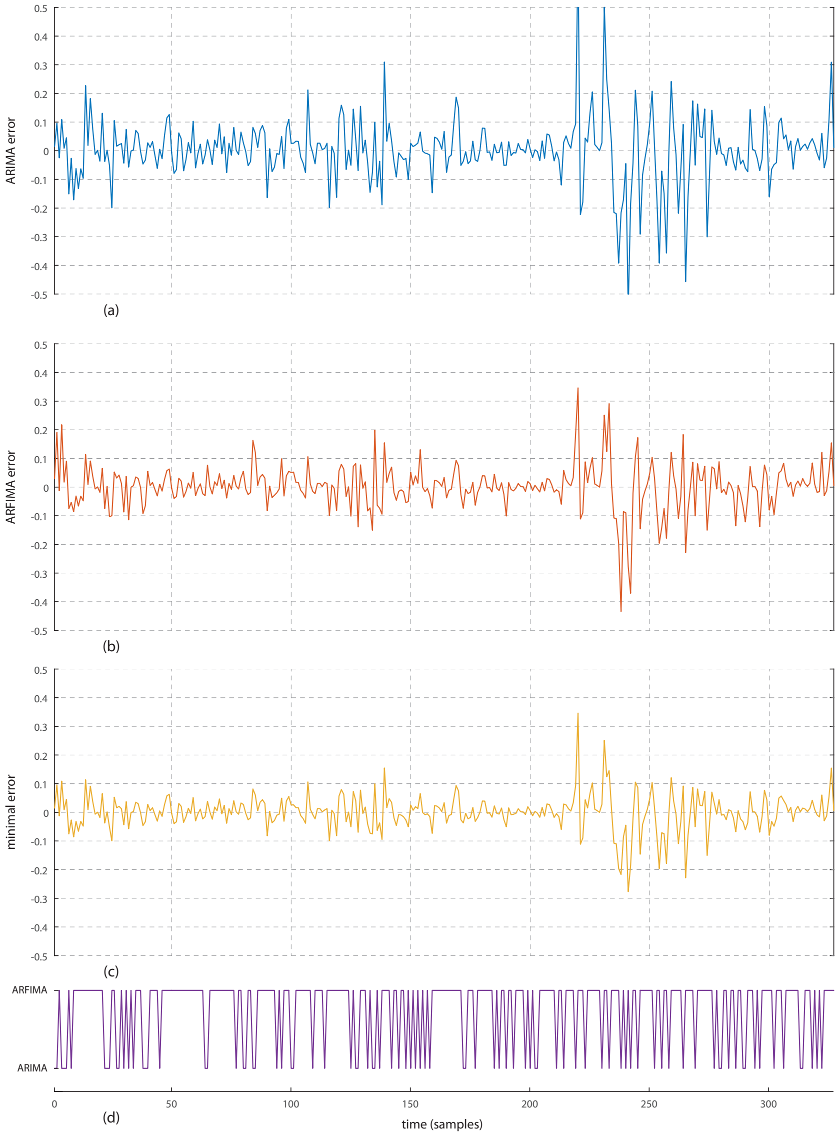


Figure 5.11: Minimal error and model selection for dataset 2: (a) ARIMA error, (b) ARFIMA error, (c) minimal error and (d) selected model.

## 5.3 Model selection analysis

In order to clarify the model selection process used by the proposed method, the individual errors of each model and the selected model along three time-series were calculated. Figure 5.11 shows the results for the dataset 2 (Greenhouse Gas Observing Network), in which Figure 5.11(a) & (b) shows the errors for the ARIMA and ARFIMA models, respectively. From the total of 327 samples, the ARFIMA model presented a lower error 239 times, while the ARIMA model was selected 88 times, as shown in Figures 5.11(c) (minimal error) and (d) (selected model). A major characteristic of this dataset is the large amount of peaks and abrupt changes.

Figure 5.12 shows the difference between ARIMA and ARFIMA absolute errors $|\varepsilon_t^G| - |\varepsilon_t^F|$, together with the respectively selected models, for samples 100 to 150 of dataset 2. Positive values indicate that the ARIMA error is larger and thus the ARFIMA model was selected.

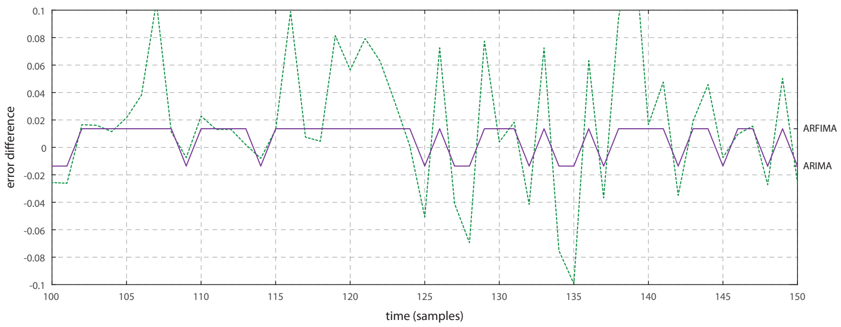


Figure 5.12: Error difference and selected model for dataset 2.

Reciprocally, negative values indicate a larger ARFIMA error, with the ARIMA model selected instead.

The results for the first 500 samples of dataset 6 (Sunspot Number) are shown in Figure 5.13. Again, the errors of the ARIMA and ARFIMA models are presented, in Figures 5.13(a) and (b). This time, however, from the total of 3253 samples, the ARIMA model was selected 1420 times, while the ARFIMA model presented a lower error 1833 times, as shown in Figures 5.13(c) and (d). This dataset presents a strong seasonal trend. Figure 5.14 shows the difference between ARIMA and ARFIMA absolute errors and the selected model for samples 100 to 150 of dataset 6.

Finally, Figure 5.15 display the results for dataset 12 (Arrhythmia). In this case, from 452 samples, the ARIMA model was selected 194 times while the ARFIMA model was selected 258 times. Although the final prediction accuracy of the proposed model was worse than either models, the tendency of selecting the better model for the task (ARFIMA) can be observed. Figure 5.16 shows the difference between ARIMA and ARFIMA absolute errors and the selected model for samples 100 to 150 of dataset 12.

It must be noticed that the errors presented in Figures 5.11 to 5.16 for ARIMA and ARFIMA do not correspond to the error of these models when used individually. This is due to the fact that, when used within the proposed method's framework, their prediction values are not directly used; instead, the final forecast value of the proposed method is used by both models in the following iterations, thus resulting in a different error profile. Figure 5.17 shows the error of ARIMA and ARFIMA when used independently and within the proposed method with dataset 2. The error values of the independent models are naturally higher.

## 5.4 Comparison with alternative approaches

Finally, the proposed method was also compared with two other methods described in Sect. 2.8. Table 5.2 shows the comparison of MAPE values between the ARIMA, ARFIMA, WMA, BW and CM models. Table 5.2 also includes the results for the Persistence Model (PM) for reference. This model simply uses the sample $x_{t-1}$ to predict $x_{t+1}$[30]. The proposed method again presents a lower accumulative error for all but three of the datasets. Datasets 1 to 9 presented and average error reduction of 12.52%, while datasets 10 to 12 presented an average error increase of 37.63%, when compared with the WMA, BW and PM methods.

### 5.4.1 Individual Household Electric Power Consumption

Figure 7 shows an interval of 500 days (November 2nd, 2007 to March 16th, 2009) of dataset 4, in which the predicted values of the WMA, BW and CM models are displayed together

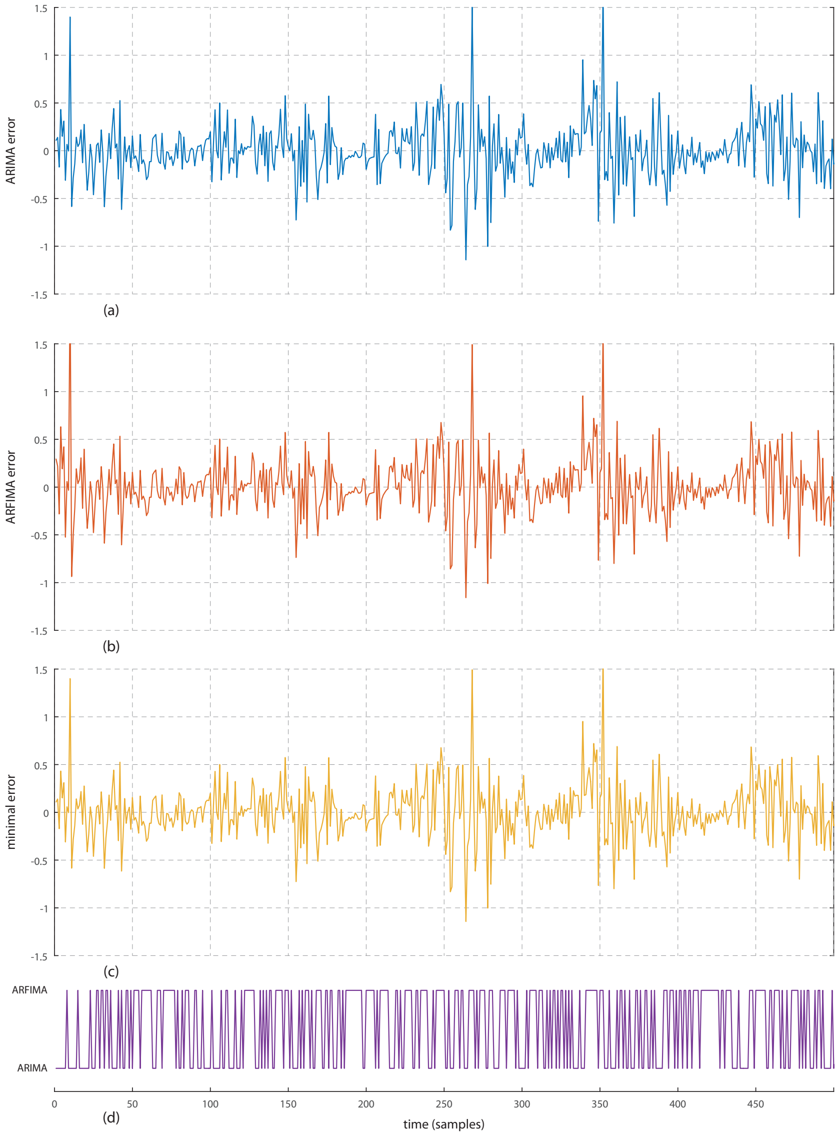


Figure 5.13: Minimal error and model selection for dataset 6: (a) ARIMA error, (b) ARFIMA error, (c) minimal error and (d) selected model.

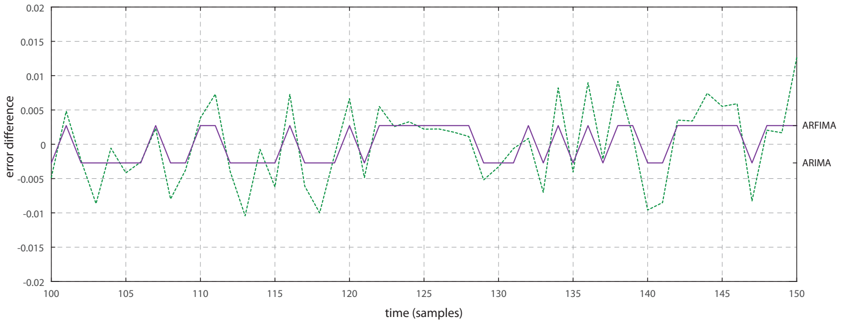


Figure 5.14: Error difference and selected model for dataset 6.

with the actual values. This dataset contains the electric power consumption of a household in

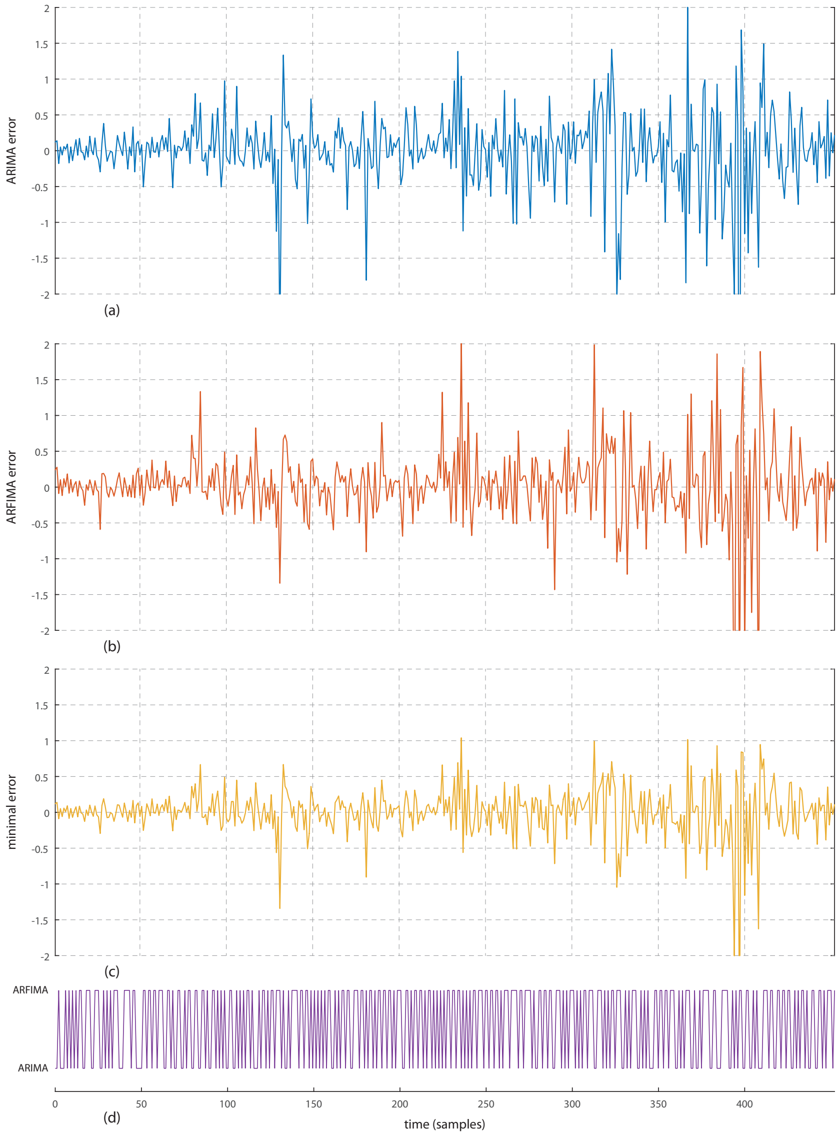


Figure 5.15: Minimal error and model selection for dataset 12: (a) ARIMA error, (b) ARFIMA error, (c) minimal error and (d) selected model.

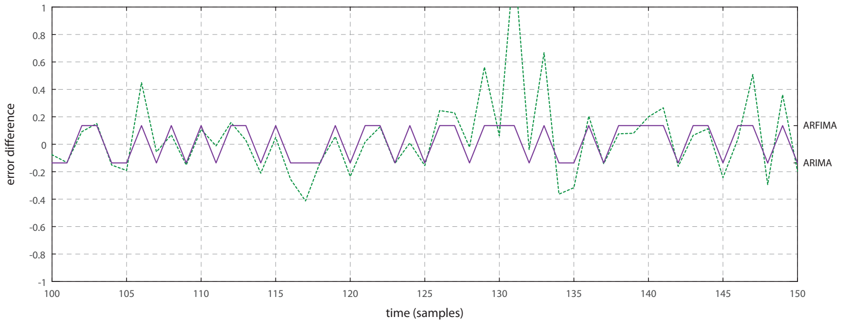
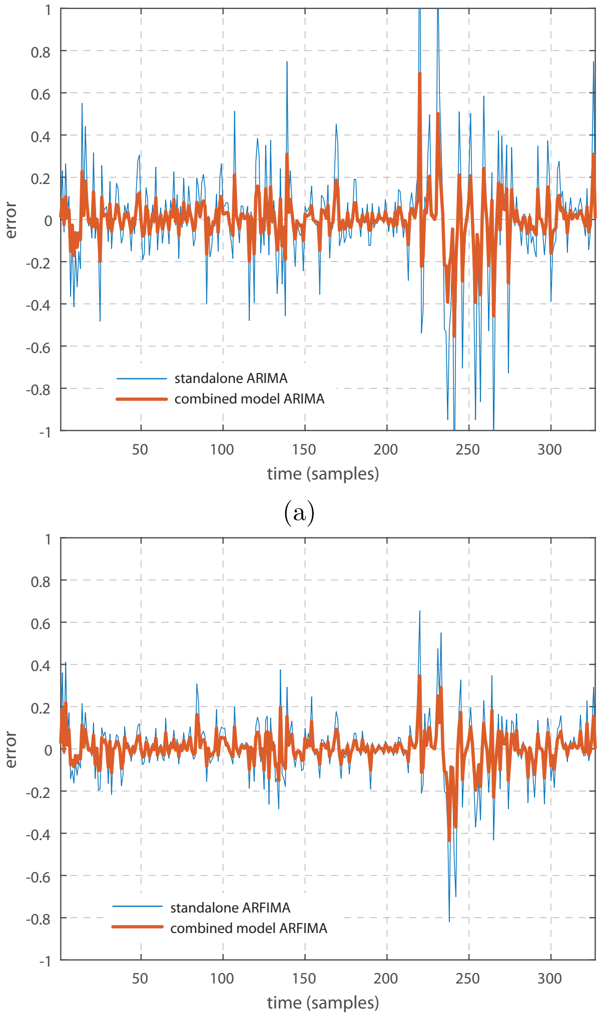


Figure 5.16: Error difference and selected model for dataset 12.

Sceaux, France, sampled every minute between 2006 and 2010 [89]. The region between vertical



(a)

(b)

Figure 5.17: Comparison of (a) ARIMA and (b) ARFIMA errors when used standalone or within the proposed model.

lines correspond to the 2008 year period, while the marked point (December 11th, 2008) is the period shown in Fig. 5.19. The lower accumulative error of the proposed CM model can be observed, due to its combined use of long and short memory dependency. The large error values of the PM method can also be clearly observed. Figure 5.19 describes the power usage in a one-day cycle, showing the consumption habits of the residents. The measurements correspond to one-minute average intervals, during a one-day cycle from midnight of December 11th until midnight of December 12th, 2008.

### 5.4.2 Abrupt Changes

The performance of the compared methods was also evaluated in segments containing abrupt changes in datasets 3, 4 & 5. These regions can be detected by the analysis of the absolute

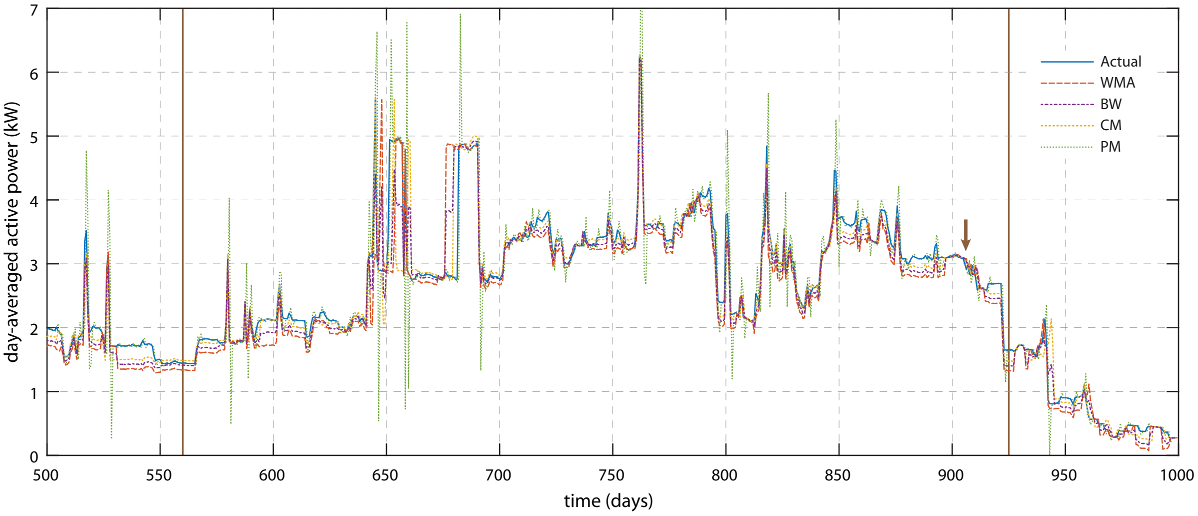


Figure 5.18: Sampling of 500 days of the Individual Household Electric Power Consumption time series (dataset 4), showing a comparison between the predicted values of the WMA, BW, CM & PM models in relation with the actual data. The region between vertical lines (days 560 & 925) correspond to the 2008 year period, while the marked point (arrow, day 906, December 11th, 2008) is the period shown in Fig. 5.19.

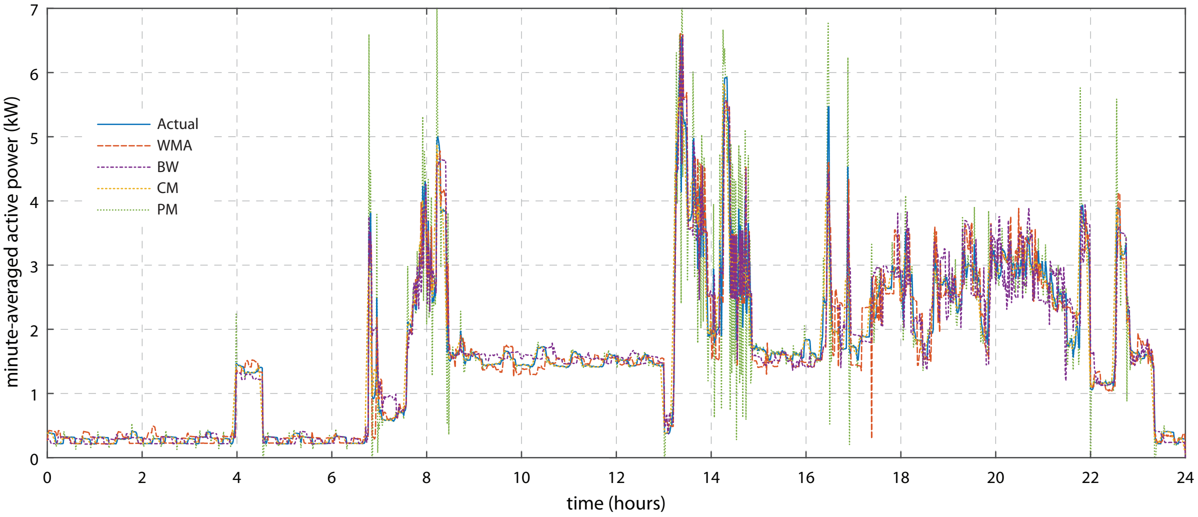


Figure 5.19: Power usage in a one-day cycle (one-minute average intervals) of the Individual Household Electric Power Consumption time series (dataset 4), from midnight of December 11th until midnight of December 12th.

percentage error (APE) and accuracy error (AE), defined as follows:

$$\mathrm{APE}_i = \left| \frac{\hat{x}_i - x_i}{x_i} \right| \times 100\% \tag{5.1}$$

$$\mathrm{AE}_i = \mathrm{APE}_i - \mathrm{MAPE} \tag{5.2}$$

When a transition from positive to negative values of the APE is observed, it corresponds to an abrupt change in the actual value of the time series [100]. As different transitions would be observed for each method, the abrupt changes detected by the PM model were used as a

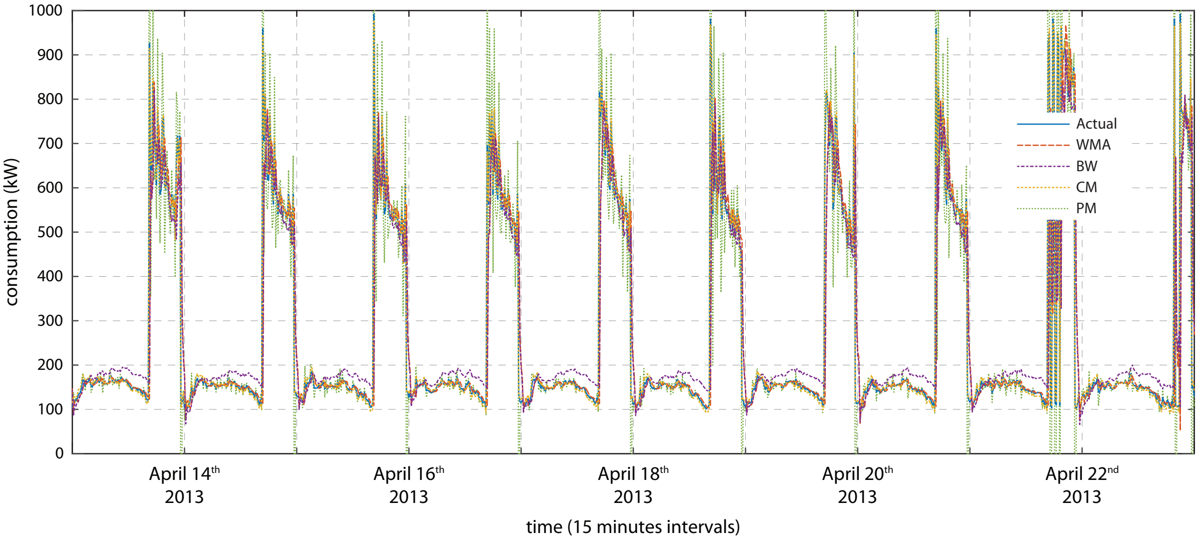


Figure 5.20: Sampling of 10 days of the Electricity Load Diagrams time series (dataset 3), showing a comparison between the predicted values of the WMA, BW, CM & PM models in relation with the actual data.

Table 5.3: Testing error (MAPE) for regions containing abrupt changes for WMA, BW, CM & PM models, with the NAC indicated for each dataset.

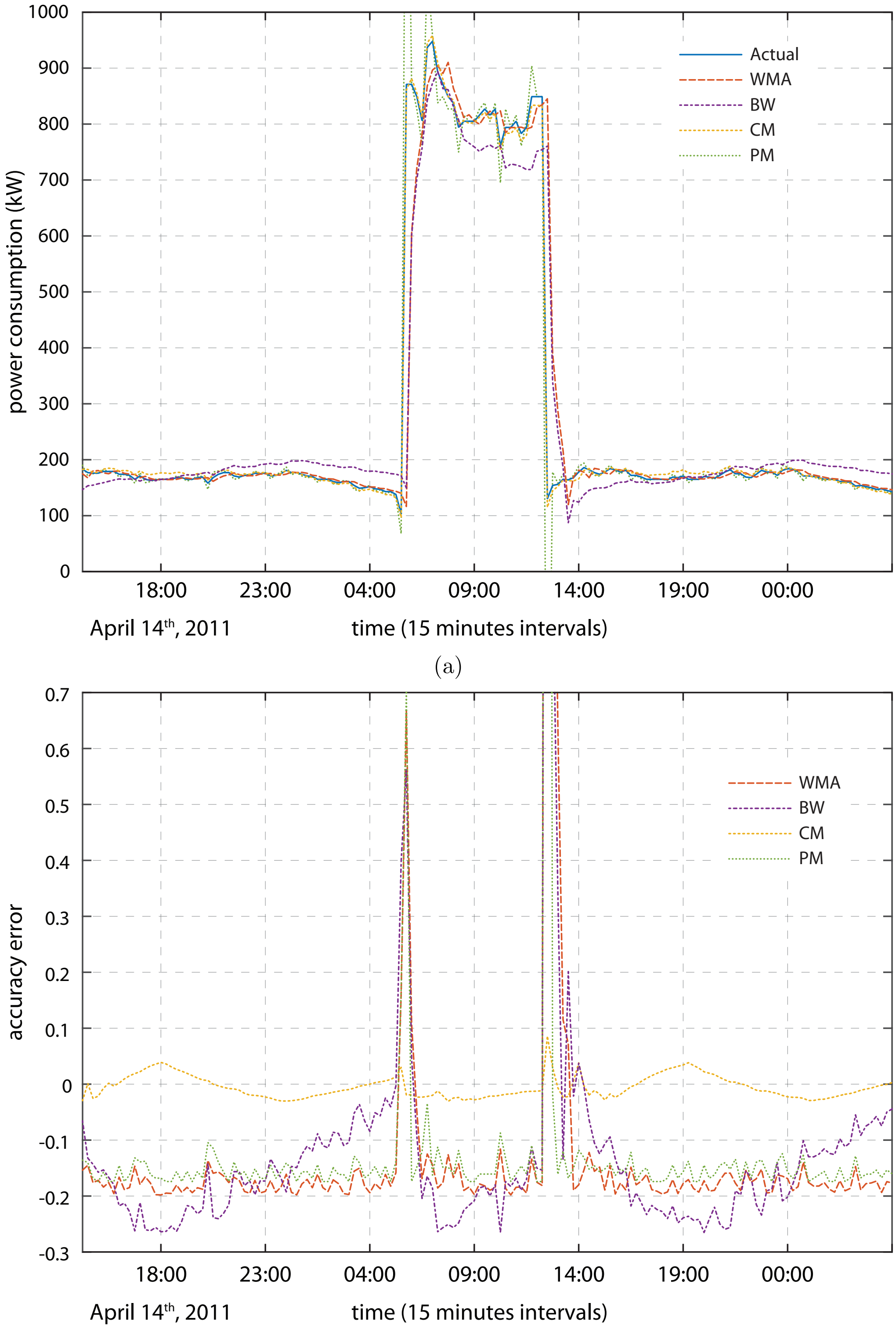
| Dataset | WMA | BW | CM | PM | NAC |
|---|---|---|---|---|---|
| 3 | 77.979% | 73.141% | **2.440%** | 65.041% | 7053 |
| 4 | 63.381% | 61.874% | **36.257%** | 48.298% | 51824 |
| 5 | 25.895% | 22.579% | **16.312%** | 18.671% | 2316 |

reference, as it is the most susceptible model for such events among the compared methods [101].

Figure 5.20 shows an interval of 10 days (April 13$^{\text{th}}$ to April 23$^{\text{rd}}$) of dataset 3. This dataset contains electricity consumption of 370 Portuguese clients in kWh, taken in 15 minutes intervals, from which client number 166 was selected due to the steep variations present in the data [89]. Fig. 5.21(a) shows a region containing an abrupt change, while the AE values for each method at the same time interval are shown in Fig. 5.21(b).

The MAPE value for the 3 following points of a detected abrupt change was calculated for each compared method. The average of all regions, as well as the number of detected abrupt changed (NAC) are shown in Table 5.3.

The proposed method showed significantly lower prediction error in the segments containing abrupt changes, especially for dataset 3. As it can be observed in Fig. 5.20, this dataset has a strong seasonal trend, which can be handled particularly well by the ARIMA model [27], and subsequently the proposed CM method, as it is partially based on the former.



(a)

(b)

Figure 5.21: Abrupt change contained in dataset 3: (a) comparison between the predicted values of the WMA, BW, CM & PM models in relation with the actual data, (b) AE values for each model, at the same time interval.

# CHAPTER 6

## Conclusions

This dissertation presented a solution that addresses the problem of memory dependency in time series forecasting by combining two well-known methods, ARIMA and ARFIMA, to create a new prediction model capable of achieving high accuracy over different types of datasets.

Instead of employing model fragmentation or preprocessing techniques, which performances are usually data-dependent, the proposed method combines the short and long-term memory properties of ARIMA and ARFIMA, respectively, so that to keep the accumulative error low along the time. This is achieved by calculating the likelihood of convergence of the exponential and hyperbolical decays of the two models and combining these values in order to obtain the final prediction.

Several different datasets were used to compare the performance of the proposed method with the original ARIMA and ARFIMA models. The results were also compared with the WMA and BW methods, which tackle the memory dependency problem, as well as the simple PM method. From the ARIMA and ARFIMA results, it is possible to observe the different cases of memory dependency among the datasets. Nevertheless, the proposed method presented lower MAPE values for most of the datasets, with the exception of datasets 10 to 12.

As shown in Table 4.1, these datasets consist of multivariate time-series of biomedical signals with large number of features, which causes the correlation between values to decrease, leading to a divergence in the prediction estimation of the proposed method. In contrast, the wavelet-transform based decomposition of the WMA and BW methods can better handle the large number of features, achieving the lowest error for these datasets.

A common assumption in temporal series prediction is that the training and test data follow the same probability distribution. A dataset shift is said to occur when the distribution differs between the two sets [9, 10]. This divergence can be due to several factors: how input features are used, how the training and test sets are selected, shifts in distribution due to non-stationarity environments, among others.

Datasets of biomedical signals commonly present such variations due to the conditions in which the signals are acquired. They also present highly non-linear and interdependent features, which can be confirmed by the overall poor performance of all used methods in the experiments in Chapter 5. Wavelet-based methods (e.g. WMA and BW) are able to estimate both the overall trend and multiple changes in the estimation order of a time series, which probably contributed for their lower susceptivity to the dataset shift problem.

The effects of dataset shift could be mitigated, for instance, with the use of multivariate model, together with a more detailed analysis of the model selection process by correlation tools, such as PP (equation 2.27) or KPSS (equation 2.34). Moreover, in the particular case of the proposed model, by modifying the calculation of the minimal accumulative error point (equation 3.11) and analyzing the impact on the likelihood of convergence (equation 3.12) may give a better understanding of the effects of a dataset shift and possibly contribute to the improvement of accuracy to a better prediction of the model.

The proposed model also presented better accuracy in regions with abrupt changes when compared with other models, being a promising alternative in applications requiring fast decision making based on data forecast.

## 6.1 Future Work

The proposed method was developed with the objective to understand the dynamics of long and short memory dependency in different scenarios. In order to improve the accuracy of the CM model on more complex datasets, future work includes the use of neural networks in order to create a hybrid model, with the objective of optimizing the model estimation and parametrization procedures[102, 103, 104]. A possible development is a hybrid neural network prediction model using the linear optimization techniques here presented, in conjunction with the backpropagation algorithm[105, 106].

The proposed model could also be the basic to create a Long-Short Term Memory (LSTM) unit to be used in a Recurrent Neural Network (RNN). Such model could have potential applications in biomedical signal processing. This would eliminate the need of external parameters when modeling datasets with long and short memory dependence characteristics.

# APPENDIX A

## R Language Source Code & Toolchain

The methods described in this thesis were developed using the R programming language. R is free, available on Windows, Mac OS and Linux. There are many types of add-on packages to different purposes and analysis. This tutorial the R language was used with RStudio (IDE). The sequence of operations described here is based on the workflow by Hyndman & Khandakar [92].

## A.1 Installing R and RStudio

The following links describes where to download and how to install the R langugage and the RStudio IDE:

- Website with installation of R in `https://cran.r-project.org/`
- Website with installation of RStudio in `https://rstudio.com/products/rstudio/download/`

After the installation, run the RStudio. On the "Packages" tab, click on "Install packages" and install the package fpp2 (make sure "install dependencies" is marked checked).

## A.2 Code Format

The R code are in this format:

```
autoplot(a10)
```

```
h02 %>% ets() %>% forecast() %>% summary()
```

## A.3 Language Packages

On the following links there is a description on how to install the packages in R, necessary do the development of this tutorial.

How to install packages: `https://rdrr.io/r/utils/install.packages.html`

### A.3.1 Utility packages

The following packages were downloaded separated from the main packages, considering their specific functions used in some situations during the analysis of time series.

- `install.packages(c('FitAR', 'fUnitRoots'))`
- TSA `https://www.rdocumentation.org/packages/TSA`
- `install.packages('useful')`

Getting started with R, these codes assume that you have the fpp2 package loaded (and that you are using at least v2.3 of the package). So you should use the command library (fpp2) before you try any examples provided here.

### A.3.2 Forecast package

If you have never previously used R, please first try to read the following link:

- Forecast: `http://pkg.robjhyndman.com/forecast/`

This contains libraries with methods and tools to analyze time series [92].

Newer version can be downloaded from CRAN and installed inside RStudio using the following command:

- `install.packages ('forecast', dependencies = TRUE)`

### A.3.3 Fable package

The following link describes how to install the Fable package:

- Fable: `http://fable.tidyverts.org/`

The v0.1.0 pre-release can be installed using:

- `install.packages ("fable", repos = "https://tidyverts.org/")`

### A.3.4 Forecasting package

This package is takes as reference the book title “Forecasting: Principles and Practice by Rob. J. Hyndman and George Athanasopoulos”. Possible to access completely free on the following link:

- Online textbook on forecasting: `https://otexts.com/fpp3/`
- All the R examples contained in this book uses the fpp3 package first: library(fpp3)

This library will upload datasets and functions from many different packages. For example, different versions of the `tidyverse`, `tsibble`, `tsibbledata`, `fable`, and `feasts` packages (more details `tidyverts.org` website).

## A.4 ARIMA Modeling

In order to apply the ARIMA model to a given dataset, the data needs first to be converted into R's time series data format.

```
sugar_price.timeseries <- ts(sugar_price,start=c(1997,5,2),
end=c(2010,10,15),frequency=252)
```

where *sugar_price* is the data being converted to time series, *start* is the initial time of the data (1997 May 2nd, in this example). As it is a monthly data divided on working days, *frequency=252* is set (quantity of working days in an average month). After that, the dataset will be process and a figure simmilar to Fig. A.1 will be displayed. To explore the components of a time series, the R code is:

```
components.ts = decompose(tsDataset)
plot(components.ts)
```

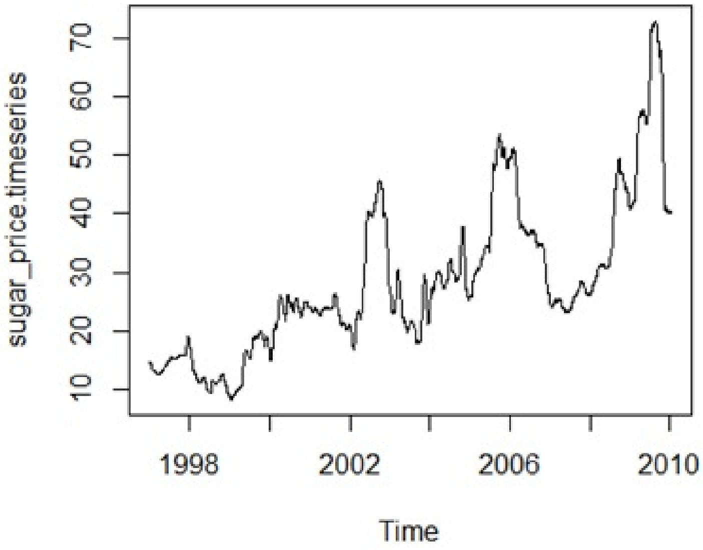


Figure A.1: Sugar price dataset on a time series

This will generate the components of this time series, as shown on Fig. A.2. It is possible to observe different components as the figure described. Analyzing Fig. A.2, we can procedure on the ARIMA model analysis. After, considering the stationarity of the time series it is necessary to remove non-stationary part for ARIMA. To gain stationarity, the Unit Root test is a common approach, such as the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
library("fUnitRoots")
urkpssTest(tsDataset, type = c("tau"), lags = c("short"),use.lag = NULL, doplot =
TRUE)
tsdatastationary = diff(tsDataset, differences=2)
plot(tsdatastationary)
```

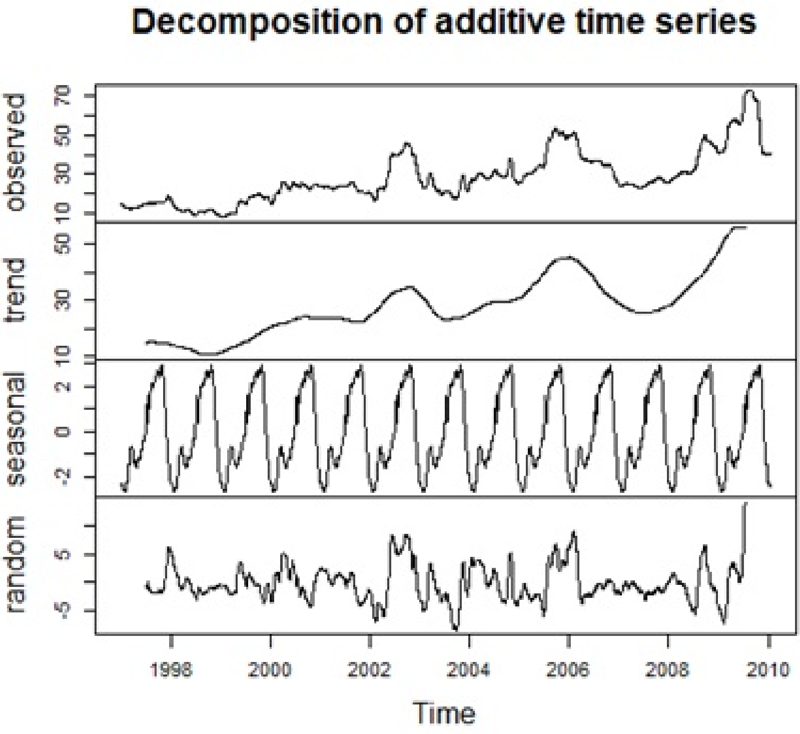


Figure A.2: Decomposition of the Sugar Price Time Series.

This results in the plot shown in Fig. A.3. We then proceed to remove the non-stationarity of the Sugar Price Time series. After removing non-stationarity, the series will become as shown in Fig. A.4. The autocorrelation can be calculated with the following code:

```
acf(tsDataset,lag.max=33)
```

It is then possible to display the ACF of the Sugar Price Time series as shown in Fig. A.5. Be aware that, the autocorrelation function `acf()` is given for all possible lags. It also is possible to set different sizes of lag, depending on the situation.

```
tsseasonaladjusted <- tsDataset- tscomponents$seasonal
ts1stationary <- diff(tsseasonaladjusted, differences=2)
```

As the code shows, it is possible to remove the seasonal factor from the data, on which makes the data stationary, as presented in Fig. A.6. It is also possible to smooth the see more clear the patterns in the time series.

## A.5 Model Fitting

After analyzing the data, the next step is to model fitting, done by observing the order, variables, etc. In this point is also necessary to look the values of $p$ and $q$ that will be used, for this we use the `acf()` and `pacf()` functions.

```
acf(t1stationary, lag.max=25)
pacf(t1stationary, lag.max=25)
```

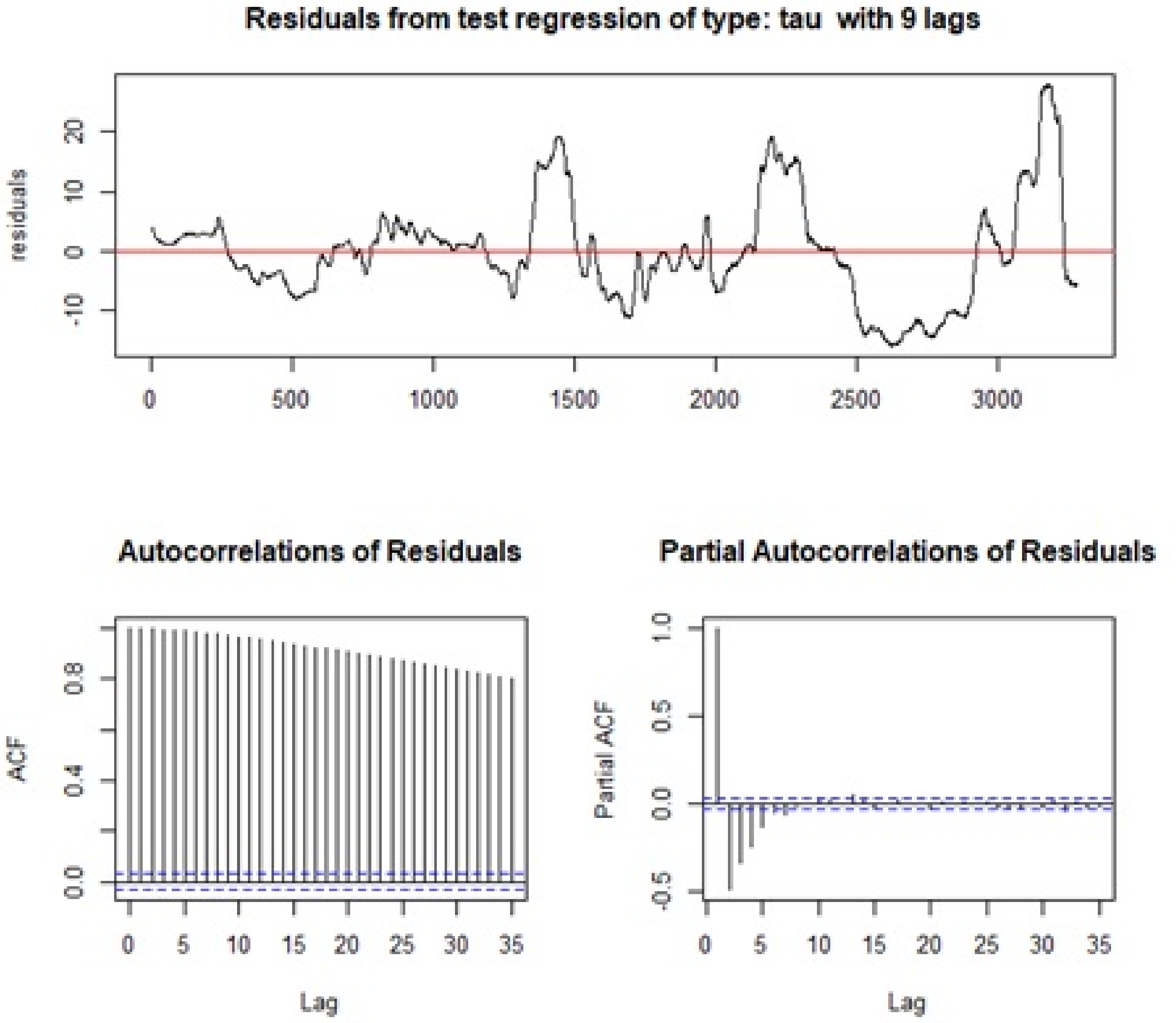


Figure A.3: Residuals from the regression test on the Sugar Price Time Series, also the ACF and PACF of the residuals considering 9 lags.

The resulting plots are presented in Fig. A.7. Analyzing the plots and the data results from the process, we can select values as presented in Fig. A.8.

```
fitARIMA2 <- arima(tsDataset, order=c(1,1,1),seasonal = list(order = c(1,0,0),
period = 12),method="ML")
library(lmtest1)
coeftest(fitARIMA2)
```

Considering the `fitARIMA` code, each parameter of this function can be as described in Fig.A.9, displaying the results of the `fitARIMA` function. In order to select with coefficients and errors, it is possible to do this selection by the function `confint()`. The Fig.A.10 displays the results of the `confitARIMA` function.

```
confint(fitARIMA)
```

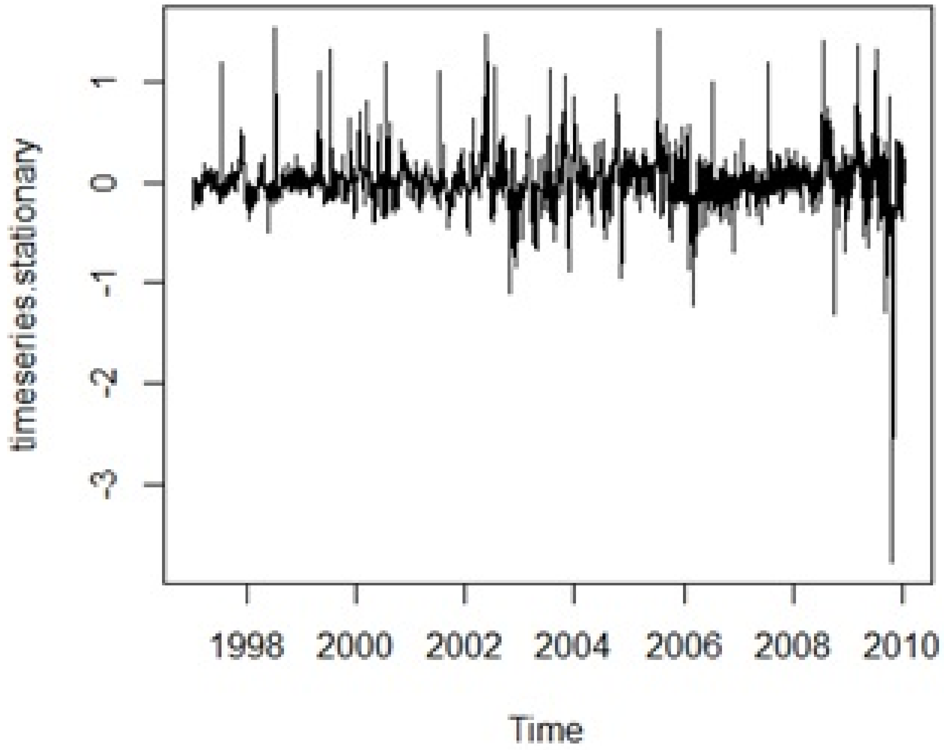


Figure A.4: Stationary Sugar Price Time Series.

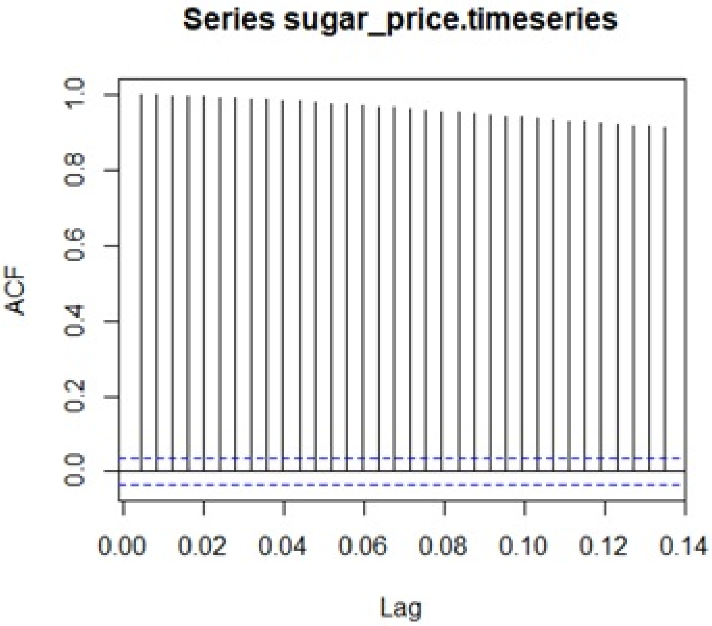


Figure A.5: ACF of the Sugar Price Time Series.

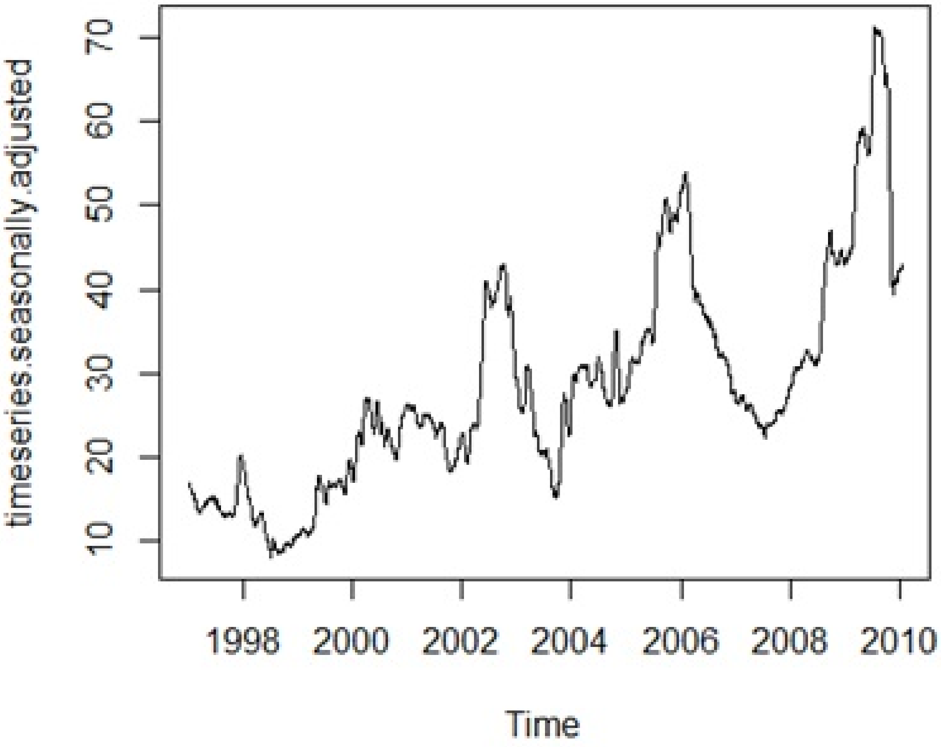


Figure A.6: Stationary Sugar Price Time Series.

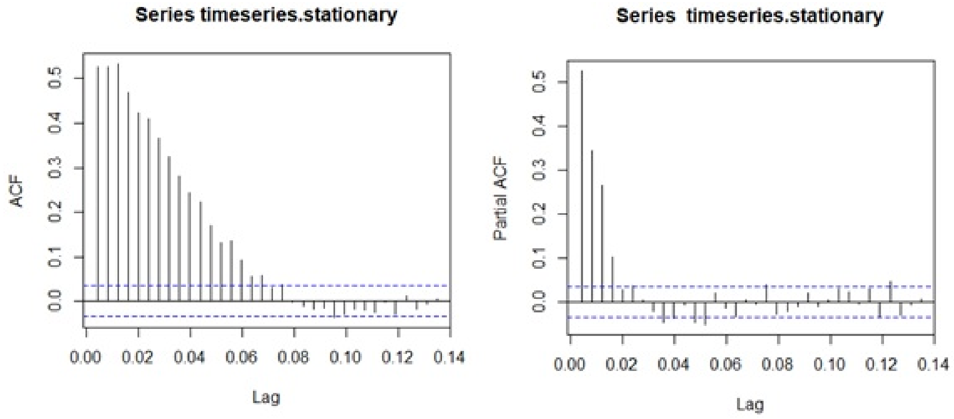
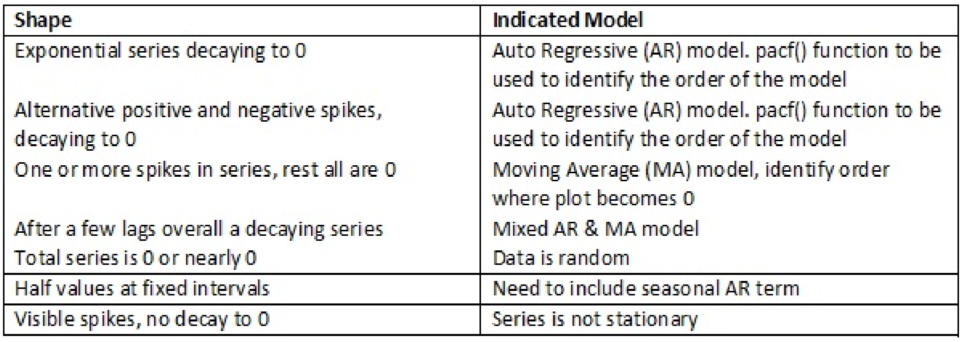


Figure A.7: ACF and PACF on the Stationary Sugar Price Time Series.

## A.6 Box Ljung Test

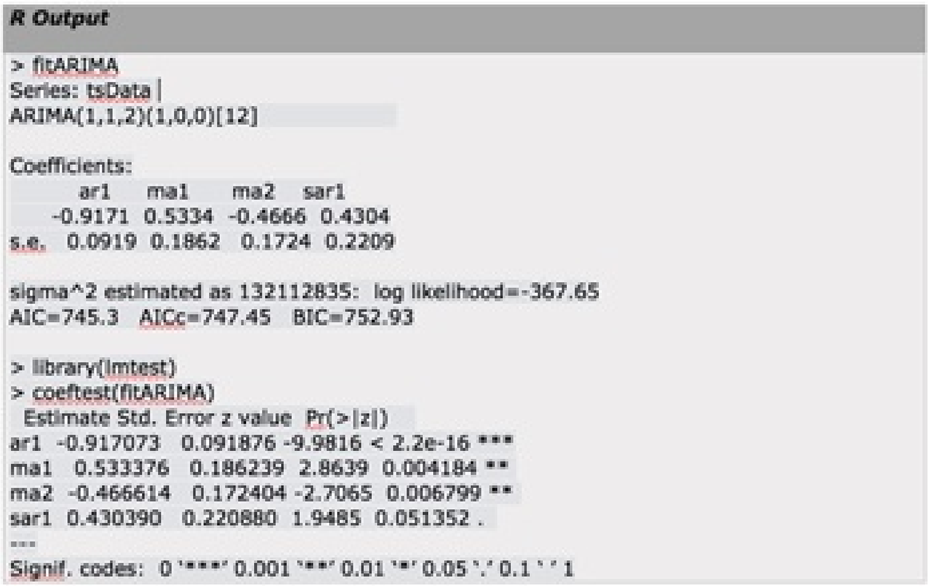
The ARIMA is based on the Box Ljung test:

```
acf(fitARIMA2$residuals1)
library(FitAR)
boxresult-LjungBoxTest (fitARIMA2$residuals1,k=3,StartLag=2)
plot(boxresult[,3],main= "Ljung-Box Q Test", ylab= "P-values", xlab= "Lag")
```



| Shape | Indicated Model |
|---|---|
| Exponential series decaying to 0 | Auto Regressive (AR) model. pacf() function to be used to identify the order of the model |
| Alternative positive and negative spikes, decaying to 0 | Auto Regressive (AR) model. pacf() function to be used to identify the order of the model |
| One or more spikes in series, rest all are 0 | Moving Average (MA) model, identify order where plot becomes 0 |
| After a few lags overall a decaying series | Mixed AR & MA model |
| Total series is 0 or nearly 0 | Data is random |
| Half values at fixed intervals | Need to include seasonal AR term |
| Visible spikes, no decay to 0 | Series is not stationary |

Figure A.8: Shape and Indication on the which model to be selected.



**R Output**

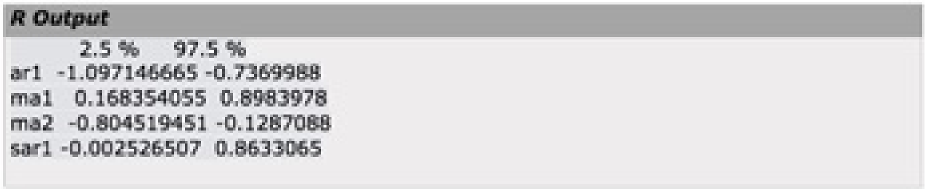
```
> fitARIMA
Series: tsData |
ARIMA(1,1,2)(1,0,0)[12]

Coefficients:
         ar1     ma1     ma2    sar1
     -0.9171  0.5334  -0.4666  0.4304
s.e.  0.0919  0.1862   0.1724  0.2209

sigma^2 estimated as 132112835:  log likelihood=-367.65
AIC=745.3   AICc=747.45   BIC=752.93

> library(lmtest)
> coeftest(fitARIMA)
  Estimate Std. Error z value  Pr(>|z|)
ar1  -0.917073   0.091876 -9.9816 < 2.2e-16 ***
ma1   0.533376   0.186239  2.8639  0.004184 **
ma2  -0.466614   0.172404 -2.7065  0.006799 **
sar1  0.430390   0.220880  1.9485  0.051352 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure A.9: R language program, displaying the results of the *fitARIMA* function.



**R Output**

```
            2.5 %     97.5 %
ar1  -1.097146665 -0.7369988
ma1   0.168354055  0.8983978
ma2  -0.804519451 -0.1287088
sar1 -0.002526507  0.8633065
```

Figure A.10: R language program, displaying the results of the *confitARIMA* function.

```
qqnorm(fitARIMA2$residuals1)
qqline(fitARIMA2$residuals1)
```

Processing this code will generate the plots shown in Fig. A.11. This plot shows the results of the `fitARIMA` function with the corresponding residuals. The output of the ACF shows that there is no significant autocorrelation.

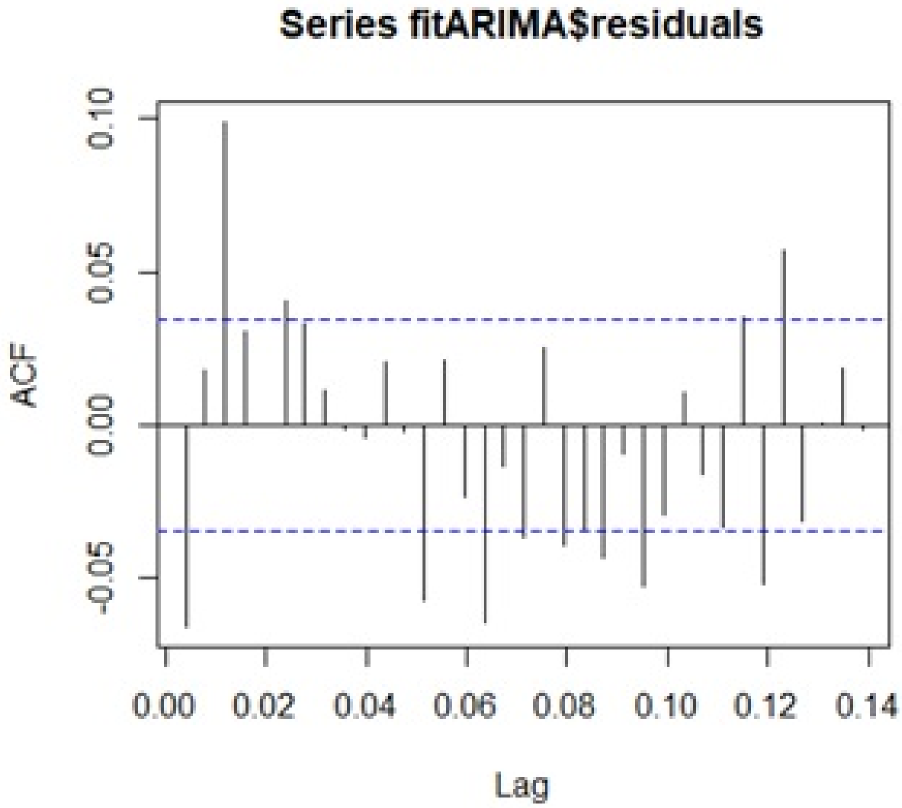


Figure A.11: The fitARIMA function showing the residuals on ACF

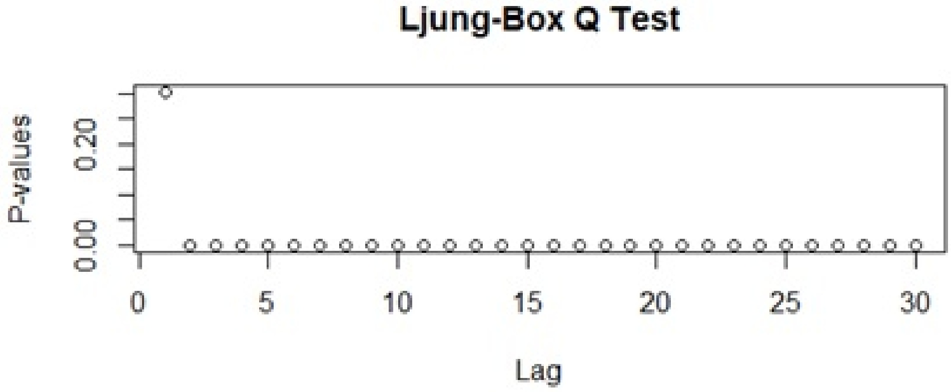


Figure A.12: The displays of the Ljung-Box Q Test, indicating non-significance values.

Is is possible to observe that the p-values for the Ljung Box Q test are much higher than 0.05, which indicates "non-significance". The rest of the values are standard as they are close to the line as shown on Fig. A.12 and also is possible to analyze the Normal Q-Q plot on A.13.

With all this information, we can assume that there is no visible pattern on the residuals, so the next logical step is to forecast. It is possible to use two different functions `ets()` and `auto.arima()`, which will automate the process selection for possible ARIMA models.

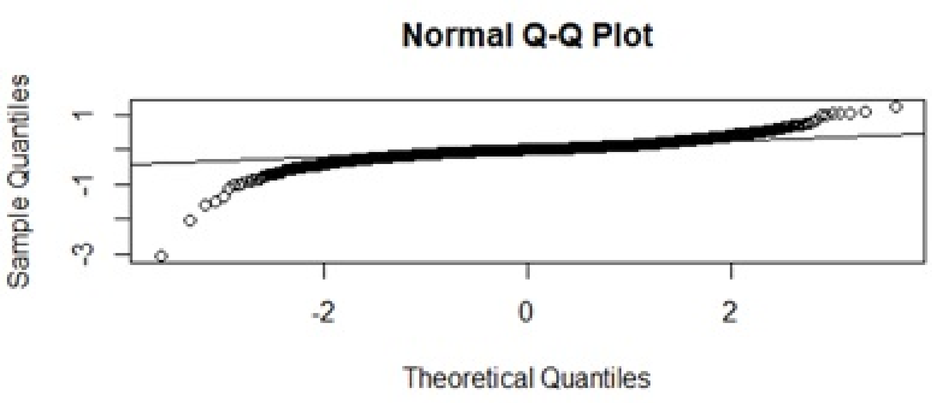
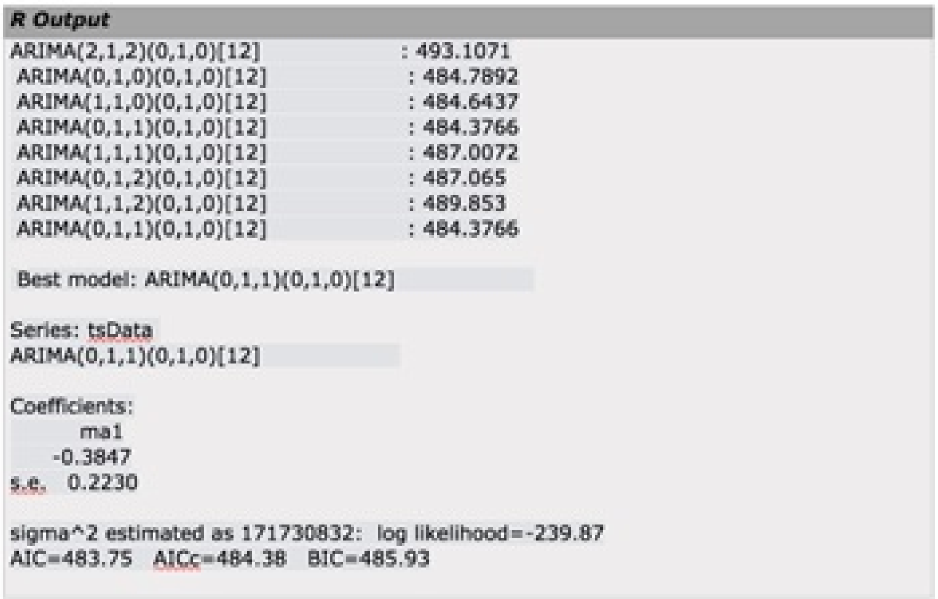


Figure A.13: Normal Q-Q Plot indicating the non-significance values as indicated on the p values aforementioned.

```
auto.arima(tsDataset, trace=TRUE)
```

Considering the AIC, it is possible to repeat the process until no lower AIC can be found. Figure A.14 shows the results of the `auto.arima()` function.



```
R Output
ARIMA(2,1,2)(0,1,0)[12]                    : 493.1071
 ARIMA(0,1,0)(0,1,0)[12]                    : 484.7892
 ARIMA(1,1,0)(0,1,0)[12]                    : 484.6437
 ARIMA(0,1,1)(0,1,0)[12]                    : 484.3766
 ARIMA(1,1,1)(0,1,0)[12]                    : 487.0072
 ARIMA(0,1,2)(0,1,0)[12]                    : 487.065
 ARIMA(1,1,2)(0,1,0)[12]                    : 489.853
 ARIMA(0,1,1)(0,1,0)[12]                    : 484.3766

 Best model: ARIMA(0,1,1)(0,1,0)[12]

Series: tsData
ARIMA(0,1,1)(0,1,0)[12]

Coefficients:
          ma1
      -0.3847
s.e.   0.2230

sigma^2 estimated as 171730832:  log likelihood=-239.87
AIC=483.75   AICc=484.38   BIC=485.93
```

Figure A.14: ARIMA automatic function shows the output of best model fit and their coefficients.

### A.6.1 Forecasting with ARIMA

The function `predict()` can be used for predicting result with different fitted models:

```
predict(fitARIMA2,n.ahead = 8)
```

Alternately the function `forecast.Arima()`, which is inside the forecast R package, can be used:

```
futureValue <- forecast.Arima(fitARIMA2,h=15, level=c(98.9))
plot.forecast(futureValue)
```

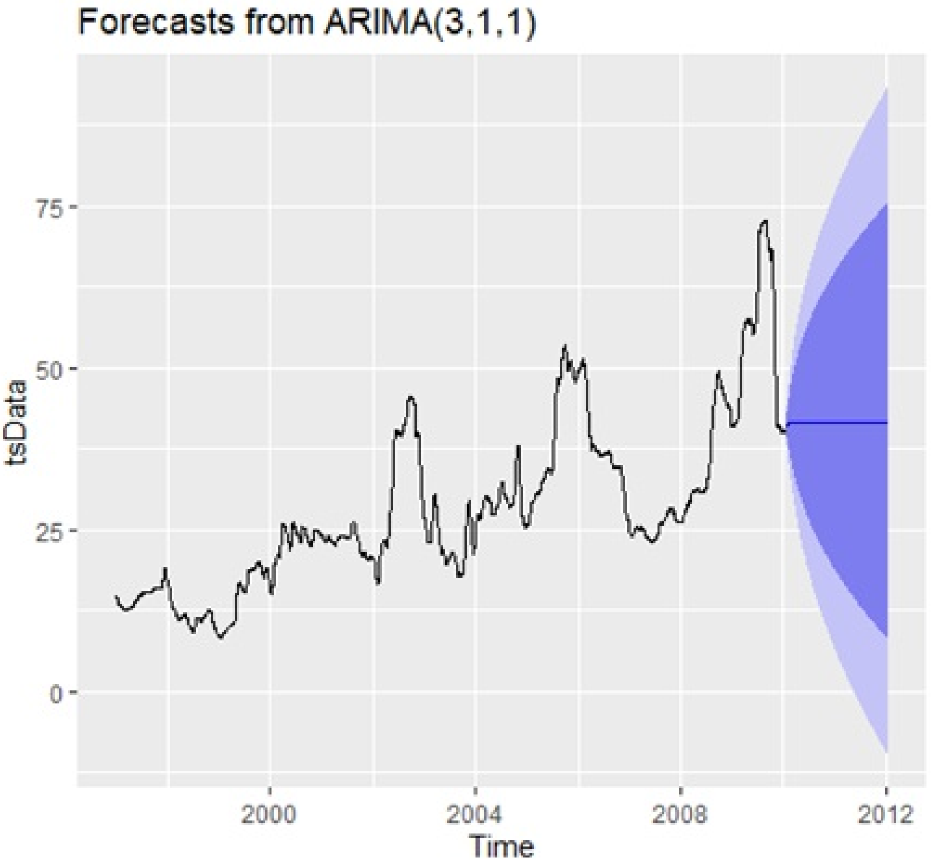


Figure A.15: The forecasts shown in blue line with 80% prediction are the dark shaded, and the 95% prediction are the light shaded area.

Running the plot forecast on `futureValue` will generate Fig. A.15. This plot shows that there is a forecast with 80% confidence within the a dark blue area, and of 95% within the light blue area. This describes the sequence of steps to analyze and forecast a time series using ARIMA.

### A.6.2 Complete R Source Code

The entire code used in this explanation is available below and in the folder R language, which is part of this tutorial package.

```
#Load Libraries
library("fUnitRoots")
library(lmtest)
library("forecast")
library(FitAR)
library(readr)

#import data -> in this example Sugar Price from the Stock Market
sugar_price <- read_csv("C:/R/Sugar_price.csv")

#convert to time series
sugar_price.timeseries <- ts(sugar_price,start=c(1997,5,2),end=c(2010,10,15),
frequency=252)

#Display the dataset
plot(sugar_price.timeseries)

#decompose into time series components
timeseries.components <- decompose(sugar_price.timeseries)
plot(timeseries.components)

#detemine stationarity of the dataset
#KPSS Unit Root Test
urkpssTest(sugar_price.timeseries, type = c("tau"), lags = c("short"),
use.lag = NULL, doplot = TRUE)
timeseries.stationary <- difference(sugar_price.timeseries, differences=2)

plot(timeseries.stationary)
acf(sugar_price.timeseries,lag.max=25)

#remove seasonality
timeseries.seasonally.adjusted <- sugar_price.timeseries-
timeseries.components$seasonal
plot(timeseries.seasonally.adjusted)

timeseries.stationary <- diff(timeseries.seasonally.adjusted, differences=1)
plot(timeseries.stationary)

par(mfrow=c(2,1))
acf(timeseries.stationary, lag.max=25)
pacf(timeseries.stationary, lag.max=25)

#fit the model
fitARIMA<-arima(sugar_price.timeseries, order=c(1,1,1),
seasonal = list(order = c(1,0,0), period = 252),method="ML")
fitARIMA

#significance of coefficients
coeftest(fitARIMA)
par(mfrow=c(1,1))
acf(fitARIMA2$residuals1)

#residual diagnostics
```

```
boxres2<-LjungBoxTest (fitARIMA2$residuals1,k=3,StartLag=2)
par(mfrow=c(2,1))
plot(boxres2[,4],main="Ljung Box Q Test", ylab="P values", xlab="Lag")

qqnorm(fitARIMA2$residuals1)
qqline(fitARIMA2$residuals1)

auto.arima(sugar_price.timeseries, trace=TRUE)

#forecast future values
par(mfrow=c(1,1))
predict(fitARIMA,n.ahead = 5)

futureValue <- forecast(fitARIMA,h=15, level=c(98.9))
autoplot(futureValue)
```

# Credits for Illustrations

Figure 2.1: Adapted from [9]

Other figures not included in this list are original form the author.

# Publications

## Journals

- Moraes Muniz da Silva, R., Kugler, M., Umezaki, T. Combined Methodology for Linear Time Series Forecasting, *IEEJ Transactions on Electrical and Electronic Engineering*, vol.15, no.12, 2020 pp.1780–1790.

## International Conferences

- Moraes Muniz da Silva, R., Kugler, M., Umezaki, T. The Impact of Memory Dependency on Precision Forecast An Analysis on Different Types of Time Series Databases. In: *6th International Conference on Pattern Recognition Applications and Methods (ICPRAM2017)*, Porto, Portugal. Proceedings..., SCITEPRESS, v.01. 2017. pp.575-582.

# Scholarships and Grants

- A scholarship from the MEXT - Ministry of Education, Culture, Sports, Science and Technology, Government of Japan, Monbukagakusho, from April 2013 to March 2017.

# Bibliography

[1] Dilli R. Aryal and Yao wu Wang. Neural network forecasting of the production level of chinese construction industry. *Journal of Comparative International Management*, 6(2):45–64, 2003.

[2] Clive William John Granger and Roselyne Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1(1):15–29, January 1980.

[3] Guoqiang Peter Zhang. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175, January 2003.

[4] Minvydas Ragulskis and Kristina Lukoseviciute. Non-uniform attractor embedding for time series forecasting by fuzzy inference systems. *Neurocomputing*, 72(10):2618–2626, June 2009. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007).

[5] Pesaran M. Hashem and Shin Yongcheol. *An Autoregressive Distributed-Lag Modelling Approach to Cointegration Analysis*, chapter 11, pages 371–413. Econometric Society Monographs. Cambridge University Press, 1999.

[6] Oguz Akbilgic, Hamparsum Bozdogan, and Mehmet Erdal Balaban. A novel hybrid RBF neural networks model as a forecaster. *Statistics and Computing*, 24(3):365–375, May 2014.

[7] Qiang Zhang, Ben-De Wang, Bin He, Yong Peng, and Ming-Lei Ren. Singular spectrum analysis and ARIMA hybrid model for annual runoff forecasting. *Water Resources Management*, 25(11):2683–2703, September 2011.

[8] Michael Scott Brown, Michael J. Pelosi, and Henry Dirska. Dynamic-radius species-conserving genetic algorithm for the financial forecasting of dow jones index stocks. In Petra Perner, editor, *Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 27–41. Springer Berlin Heidelberg, 2013.

[9] George Edward Pelham Box and Gwilym Meirion Jenkins. *Time series analysis: forecasting and control.* Holden-Day, 1976.

[10] Valdério Anselmo Reisen and Fabio Fajardo Molinares. Robust estimation in time series with long and short memory properties. In *Proceedings of the Conference on Stochastic Models and their Applications*, number 39 in Statistics: Methodology, pages 207–224, January 2012.

[11] Wilfredo Palma. *Long-Memory Time Series: Theory and Methods*. Wiley Series in Probability and Statistics. Wiley, April 2007.

[12] Richard T. Baillie. Long memory processes and fractional integration in econometrics. *Journal of Econometrics*, 73(1):5 – 59, July 1996.

[13] Jonathan Richard Morley Hosking. Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research*, 20(12):1898–1908, December 1984.

[14] Ying Siew Lim, Ying Chin Lim, and Mah Jin Wee Pauline. ARIMA and integrated ARFIMA models for forecasting air pollution index in Shah Alam, Selangor. *Malaysian Journal of Analytical Sciences*, 43(7):257–263, 2008.

[15] Rosario N. Mantegna and H. Eugene Stanley. *An Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, 2000.

[16] Mahendran Shitan, Pauline Mah Jin Wee, Ying Chin Lim, and Ying Siew Lim. ARIMA and integrated ARFIMA models for forecasting annual demersal and pelagic marine fish production in Malaysia. *Malaysian Journal of Mathematical Sciences*, 2(2):41–54, July 2008.

[17] Alireza Erfani and Ahmad Jafari Samimi. Long memory forecasting of stock price index using a fractionally differenced ARMA model. *Journal of Applied Sciences Research*, 5(10):1721–1731, 2009.

[18] David A. Pierce. A duality between autoregressive and moving average processes concerning their least squares parameter estimates. *Annals of Mathematical Statistics*, 41(2):422–426, April 1970.

[19] H. Tong and K. S. Lim. Threshold autoregression, limit cycles and cyclical data. *Journal of the Royal Statistical Society*, 42(3):245–292, 1980.

[20] A. Hossein Vahabie, M. Mahdi Rezaei Yousefi, Babak N. Araabi, Caro Lucas, and Saeedeh Barghinia. Combination of singular spectrum analysis and autoregressive model for short term load forecasting. In *Proceedings of the IEEE Lausanne Power Tech*, pages 1090–1093, July 2007.

[21] Mehdi Khashei and Mehdi Bijari. Which methodology is better for combining linear and nonlinear models for time series forecasting? *Journal of Industrial and Systems Engineering*, 4(4):265–285, 2011.

[22] Hamid Amadeh, A. Amini, and F. Effati. ARIMA and ARFIMA prediction of persian gulf gas-oil F.O.B. *Iranian Journal of Investment Knowledge*, 2(7):213–231, 2013.

[23] Ricardo Moraes Muniz da Silva, Mauricio Kugler, and Taizo Umezaki. The impact of memory dependency on precision forecast - an analysis on different types of time series databases. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications, 2017.

[24] George Udny Yule. Why do we sometimes get nonsense-correlations between time-series?– a study in sampling and the nature of time-series. *Journal of the Royal Statistical Society*, 89(1):1–63, January 1926.

[25] Richard T. Baillie and Sang-Kuck Chung. Modeling and forecasting from trend-stationary long memory models with applications to climatology. *International Journal of Forecasting*, 18(2):215 – 226, 2002. Forecasting Long Memory Processes.

[26] G. M. Jenkins and M. B. Priestley. The spectral analysis of time-series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 19(1):1–12, 1957.

[27] Eric Ghysels, Denise R. Osborn, and Paulo M. M. Rodrigues. *Forecasting Seasonal Time Series*, chapter 13, pages 659–711. Elsevier, 2006.

[28] Clive William John Granger. Combining forecasts - twenty years later. *Journal of Forecasting*, 8(3):167–173, July 1989. Invited review.

[29] Andrew C. Harvey and Neil Shephard. *Structural time series models*, volume Vol. 11:Econometrics, pages 261–302. North Holland, Amsterdam, (edited by g.s. maddala, c.r. rao and h.d. vinod) edition, 1993.

[30] Jr. Everette S. Gardner and Ed. McKenzie. Forecasting trends in time series. *Management Science*, 31(10):1237–1246, October 1985.

[31] C. Gourieroux, A. Monfort, and G. Gallo. *Time Series and Dynamic Models*, volume 3 of *Themes in Modern Econometrics*. Cambridge University Press, 1997.

[32] LUIS A. AGUIRRE and LEONARDO A. B. TÔRRES. CONTROL OF NONLINEAR DYNAMICS: WHERE DO MODELS FIT IN? *International Journal of Bifurcation and Chaos*, 10(03):667–681, mar 2000.

[33] Ewout Steyerberg. *Clinical Prediction Models A Practical Approach to Development, Validation, and Updating.* Statistics for Biology and Health. Springer New York, 2009.

[34] Hamed Valizadeh Haghi, M. Tavakoli Bina, and Masoud Aliakbar Golkar. Nonlinear modeling of temporal wind power variations. *IEEE Transactions on Sustainable Energy*, 4(4):838–848, Oct 2013.

[35] Markus Bögl, Wolfgang Aigner, Peter Filzmoser, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Visual analytics for model selection in time series analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2237–2246, December 2013.

[36] Ukaegeu L. U. Omekara C. O., Okereke O. E. Forecasting liquidity ratio of commercial banks in nigeria. *Microeconomics and Macroeconomics*, 4(1):28–36, 2016.

[37] Robert Fry Engle III and Aaron D. Smith. Stochastic permanent breaks. *The Review of Economics and Statistics*, 81(4):553–574, November 1999.

[38] Michelangelo Ceci, Roberto Corizzo, Fabio Fumarola, Donato Malerba, and Aleksandra Rashkovska. Predictive modeling of pv energy production: How to set up the learning task for a better prediction? *IEEE Transactions on Industrial Informatics*, 13(3):956–966, June 2017.

[39] A. I. McLeod. Duality and other properties of multiplicative seasonal autoregressive-moving average models. *Biometrika*, 71(1):207–211, April 1984.

[40] Thorsten Pohlert. trend: Non-Parametric Trend Tests and Change-Point Detection, jul 2018.

[41] Mohamed Boutahar and Rabeh Khalfaoui2. Estimation of the long memory parameter in non stationary models: A simulation study. *GREQAM - Groupement de Recherche en Économie Quantitative d'Aix-Marseille*, May 2011.

[42] OLANREWAJU K. ONASANYA, OLUSEGUN ALFRED OLAKUNLE, and ADENIJI OYEBIMPE EMMANUEL. Forecast performance of multiplicative seasonal arima model: an application to naira/us dollar exchange rate. *American Journal of Applied Mathematics and Statistics*, 2(3):172–178, 2014.

[43] Zhou Zhou, Zhiwei Xu, and Wei Biao Wu. Long-term prediction intervals of time series. *IEEE Transactions on Information Theory*, 56(3):1436–1446, March 2010.

[44] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, dec 1952.

[45] Knut M. Wittkowski. Friedman-type statistics and consistent multiple comparisons for unbalanced designs with missing data. *Journal of the American Statistical Association*, 83(404):1163–1170, dec 1988.

[46] Spyros Makridakis. *Forecasting methods for management.* Wiley, New York, 1989.

[47] Charles D. Bailey and Sanjay Gupta. Judgement in learning-curve forecasting: a laboratory study. *Journal of Forecasting*, 18(1):39–57, jan 1999.

[48] Chanmi Lee, Suhwi Park, and Jaesik Jeong. Comprehensive comparison of normality tests: Empirical study using many different types of data. *Journal of the Korean Data and Information Science Society*, 27(5):1399–1412, sep 2016.

[49] David Alan Dickey and Wayne Arthur Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, June 1979.

[50] Leonardo R. Souza and Jeremy Smith. Effects of temporal aggregation on estimates and forecasts of fractionally integrated processes: a Monte-Carlo study. *International Journal of Forecasting*, 20(3):487–502, July 2004.

[51] Tracey Cassar, Kenneth P. Camilleri, and Simon G. Fabri. Order estimation of multivariate arma models. *IEEE Journal of Selected Topics in Signal Processing*, 4(3):494–503, June 2010.

[52] Greta M. Ljung and George Edward Pelham Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

[53] Aidan Meyler, Geoff Kenny, and Terry Quinn. Forecasting irish inflation using arima models. *Central Bank and Financial Services Authority of Ireland Technical Paper Series*, 1998(3):1–48, December 1998.

[54] Javier E. Contreras-Reyes and Wilfredo Palma. Statistical analysis of autoregressive fractionally integrated moving average models in R. *Computational Statistics*, 28(5):2309–2331, October 2013.

[55] Christopher F. Baum. Tests for stationarity of a time series. *Stata Technical Bulletin*, 10(57):36–39, September 2000.

[56] Robert Fildes and Spyros Makridakis. The impact of empirical accuracy studies on time series analysis and forecasting. *International Statistical Review*, 63(3):289–308, December 1995.

[57] A. I. McLeod. Hyperbolic decay time series. *Journal of Time Series Analysis*, 19(4):473–483, 1998.

[58] Krzysztof Burnecki and Grzegorz Sikora. Estimation of FARIMA parameters in the case of negative memory and stable noise. *IEEE Transactions on Signal Processing*, 61(11):2825–2835, June 2013.

[59] Clifford M. Hurvich and Bonnie K. Ray. Estimation of the memory parameter for nonstationary or noninvertible fractionally integrated processes. *Journal of Time Series Analysis*, 16(1):17–41, 1995.

[60] John Geweke and Susan Porter-Hudak. The estimation and application of long-memory time series models. *Journal of Time Series Analysis*, 4(4):221–238, July 1983.

[61] J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

[62] Fallaw Sowell. Modeling long-run behavior with the fractional ARIMA model. *Journal of Monetary Economics*, 29(2):277–302, April 1992.

[63] M. B. Priestley. *Spectral Analysis and Time Series.* Academic Press, 1st edition, October 1981.

[64] Niels Haldrup and J. Eduardo Vera Valdés. Long memory, fractional integration, and cross-sectional aggregation. *Journal of Econometrics*, 199(1):1 – 11, 2017.

[65] Jan Beran and Yuanhua Feng. Local polynomial fitting with long-memory, short-memory and antipersistent errors. *Annals of the Institute of Statistical Mathematics*, 54(2):291–311, June 2002.

[66] Rafał Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach.* John Wiley & Sons Ltd., 2006.

[67] L. M. V. Carvalho, A. A. Tsonis, C. Jones, H. R. Rocha, and P. S. Polito. Anti-persistence in the global temperature anomaly field. *Nonlinear Processes in Geophysics*, 14:723–733, November 2007.

[68] Krzysztof Burnecki. FARIMA processes with application to biophysical data. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(5):P05015, 2012.

[69] Samuel C. Kou. Stochastic modeling in nanoscale biophysics: Subdiffusion within proteins. *The Annals of Applied Statistics*, 2(2):501–535, 2008.

[70] Jan Beran, Yuanhua Feng, Sucharita Ghosh, and Rafal Kulik. *Long-Memory Processes: Probabilistic Properties and Statistical Methods.* Springer, Berlin, Heidelberg, 2013.

[71] Jeremy Smith, Nick Taylor, and Sanjay Yadav. Comparing the bias and misspecification in ARFIMA models. *Journal of Time Series Analysis*, 18(5):507–527, 1997.

[72] Bruce E. Hansen. Threshold autoregression in economics. *Statistics and Its Interface*, 4(2):123–127, 2011.

[73] Ahmad Kalhor, Hossein Iranmanesh, and Majid Abdollahzade. Online modeling of real-world time series through evolving AR models. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 1–6, June 2012.

[74] Wen chuan Wang, Kwok wing Chau, Dong mei Xu, and Xiao-Yun Chen. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Resources Management*, 29(8):2655–2675, June 2015.

[75] Emeka Nkoro and Aham Kelvin Uko. Autoregressive distributed lag (ARDL) cointegration technique: application and interpretation. *Journal of Statistical and Econometric Methods*, 5(4), 2016.

[76] Michael K. Andersson. Do long-memory models have long memory? *International Journal of Forecasting*, 16(1):121–124, January 2000.

[77] Donald W. K. Andrews. Tests for parameter instability and structural change with unknown change point. *Econometrica*, 61(4):821–856, July 1993.

[78] Wai-Sum Chan. A note on time series model specification in the presence of outliers. *Journal of Applied Statistics*, 19(1):117–124, 1992.

[79] Wai-Sum Chan. Outliers and financial time series modelling: A cautionary note. *Mathematics and Computers in Simulation (MATCOM)*, 39(3):425–430, 1995.

[80] Konstantina Papagiannaki, Nina Taft, Zhi-Li Zhang, and Christophe Diot. Long-term forecasting of internet backbone traffic: observations and initial models. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1178–1188, March 2003.

[81] Kyungduk Ko and Marina Vannucci. Bayesian wavelet-based methods for the detection of multiple changes of the long memory parameter. *IEEE Transactions on Signal Processing*, 54(11):4461–4470, November 2006.

[82] David A. Freedman. *Statistical Models*. Cambridge University Press, 2009.

[83] Neri Merhav and Meir Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6):2124–2147, October 1998.

[84] Ernst Eberlein and Murad S. Taqqu, editors. *Dependence in Probability and Statistics*, volume 11 of *Progress in Probability*. Birkhäuser Boston, 1986.

[85] T. Schweder and N.L. Hjort. *Confidence, Likelihood, Probability: Statistical Inference with Confidence Distributions*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 2016.

[86] Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, March 1987.

[87] Yin-Wong Cheung and Francis X. Diebold. On maximum likelihood estimation of the differencing parameter of fractionally-integrated noise with unknown mean. *Journal of Econometrics*, 62(2):301–316, June 1994.

[88] Yoosoon Chang, Yongok Choi, and Joon Y. Park. A new approach to model regime switching. *Journal of Econometrics*, 196(1):127 – 143, 2017.

[89] M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[90] Center for Advanced Studies on Applied Economics University of São Paulo. CEPEA/ESALQ Crystal Sugar Price Index. `https://www.cepea.esalq.usp.br/en/indicator/sugar.aspx`. Accessed on February 2020.

[91] SILSO World Data Center. The International Sunspot Number. *International Sunspot Number Monthly Bulletin and online catalogue*, 1749–2019. Royal Observatory of Belgium, avenue Circulaire 3, 1180 Brussels, Belgium.

[92] Rob J. Hyndman and Yeasmin Khandakar. Automatic Time Series Forecasting: The Forecast Package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

[93] D. D. Lucas, C. Yver Kwok, P. Cameron-Smith, H. Graven, D. Bergmann, T. P. Guilderson, R. Weiss, and R. Keeling. Designing optimal greenhouse gas observing networks that consider performance and cost. *Geoscientific Instrumentation, Methods and Data Systems*, 4(1):121–137, 2015.

[94] Pinar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power using machine learning methods. *International Journal of Electrical Power and Energy Systems*, 60:126 – 140, 2014.

[95] Heysem Kaya, Tüfekci, and Sadik Fikret Gürgen. Local and global learning methods for predicting power of a combined gas & steam turbine. In *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering*, pages 13–18, March 2012.

[96] Keith W. Hipel and A. Ian McLeod. *Time Series Modelling of Water Resources and Environmental Systems*, volume 45 of *Developments in Water Science*. Elsevier Science Ltd., 1st edition, 1994.

[97] Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6), nov 2001.

[98] Christos Sapsanis, George Georgoulas, Anthony Tzes, and Dimitrios Lymberopoulos. Improving EMG based classification of basic hand movements using EMD. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, jul 2013.

[99] E. Alhoniemi, A. Honkela, K. Lagus, S.J. Seppa, P. Wagner, and H. Valpola. Compact modeling of data using independent variable group analysis. *IEEE Transactions on Neural Networks*, 18(6):1762–1776, nov 2007.

[100] Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes: theory and application.* Prentice Hall, 1993.

[101] James W. Taylor, Lilian M. de Menezes, and Patrick E. McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22(1):1 – 16, January 2006.

[102] Mehdi Khashei and Mehdi Bijari. A new hybrid methodology for nonlinear time series forecasting. *Modelling Simulation in Engineering*, 2011(15), Jan 2011.

[103] Pedro Lopez-Garcia, Enrique Onieva, Eneko Osaba, Antonio D. Masegosa, and Asier Perallos. A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):557–569, Feb 2016.

[104] Parham Piroozan. Pattern recognition in real time using neural networks: An application for pressure measurement. In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, pages 566–572. INSTICC, SciTePress, 2016.

[105] Reza Askari Moghadam and Mehrnaz Keshmirpour. *Hybrid ARIMA and Neural Network Model for Measurement Estimation in Energy-Efficient Wireless Sensor Networks*, pages 35–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[106] Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa. Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems*, 29(2):143–155, May 2012.