Universidad del Norte
Department of Electrical and Electronics Engineering

Adaptive Methodology based on Computational Intelligence for Time Series Modeling

By

Jamer René Jimenez Mares

DOCTORAL THESIS

Advisor
Dr. Christian G. Quintero M.

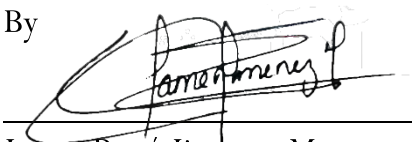
Barranquilla, Atlántico, Colombia
June 2021

Universidad del Norte
Department of Electrical and Electronics Engineering

Adaptive Methodology based on Computational Intelligence for Time Series Modeling

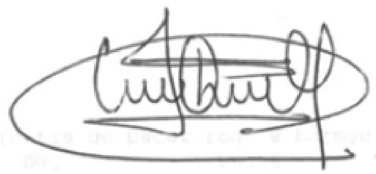
A dissertation presented to the Universidad del Norte in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Electronics Engineering

By



Jamer René Jimenez Mares

Advisor:



Dr. Christian G. Quintero M.

Barranquilla, Atlántico, Colombia
June 2021

Universidad del Norte
Department of Electrical and Electronic Engineering

ABSTRACT

Adaptive Methodology Based on Computational Intelligence for Time Series Modeling

By Jamer René Jimenez Mares

Advisor: Dr. Christian G. Quintero M.

Time series processes are important in several sectors like marketing, transport, energy, telecommunications, etc. Time series forecasting tasks can help in operative and strategic tasks. Several conventional and non-conventional techniques as ARIMA models, artificial neural networks (*ANN*), support vector machines (*SVM*), Regression Tree Ensembles *(RTE)* or combinations of them have been used for time series modeling. The implementation of this type of techniques provides support in time series modeling, however, normally the trained models may lose performance due to the dynamic behavior of the phenomena. A methodology capable to assess the performance and maintenance of the models is necessary to guarantee the automatic adaptability in each case. Hence, in this research an adaptive methodology based on computational intelligence for time series modeling is proposed. In this case, an *Auditor* is developed, which allows identifying when a model must be retrained or updated before losing forecast performance. Furthermore, when the retrain process is not achieving a better performance, a new metric is proposed to choose which time series modeling technique is included in the knowledge base. The intelligent system allows building the time series model automatically, considering exogenous variables such as weather, calendar and statistical transformations to group and simplify the number of models required.

The proposed approach has been experimentally tested for power consumption, road vehicular traffic and energy price time series and its performance has been tested in simulation runs.

Acknowledgments

I want to thank God before anything else for helping me reach this great achievement in my professional life. This would not have been possible without God´s help and only by his guide and hand that we will get any project, we undertake to succeed.

To my wife for all her understanding and support in difficult times and through every phase of this project, because of all her love and support I never felt myself to be alone. To my mother for her unconditional dedication, support and love in every project that I have decided to undertake. To my children (Isaias Jimenez R., Jeronimo Jimenez R. and Jacobo Jimenez N.) whose innocence and love became the greatest motivation for me to improve myself at every level. To the Gutierrez Cervantes family for showing me their love and support for the past three years.

To my tutor, counselor, professor and friend Christian Quintero who through his professional support and motivation showed me the correct path to sort out every adversity that came along the way during these five years. His advice and recommendations guided me in the best possible way to improve myself professionally.

To my father and family for understanding my absences through sad and happy times, but this would not have been possible without them.

To my grandparents (Filomena Cera, Rafael Jimenez and Vicia Melendez) and Jorgito Gutierrez who even though they are not here with us anymore, I know in my heart that they are happy for my achievement. While they were alive, they always showed me their love and happiness with every achievement I got. I know God has awarded me these angels so they can tag along with me in every step I undertake.

At the same time, I want to thank every member of the electrical and electronic engineering lab and the GIRSI research group for their fellowship, support and commitment throughout this time.

Thanks a lot to my professors within my doctorate for their suggestions and effort that helped me improve my experience as a researcher. Very special thanks to

Joaquin Melendez and Joan Colomer from Universidad de Girona for opening to me the doors to an international level that allowed me to expand and apply my knowledge.

To my lab partners for their support and warmth throughout this doctorate that are indeed the base for friendship.

To my friends and loved ones whom supported me to make this goal possible.

To my God, wife, sons, mother, family and friends

General Contents

Detailed Contents

List of Figures

List of Tables

PART I

INTRODUCTION AND RELATED WORK

Chapter 1

Introduction

This chapter introduces the work presented in this thesis. Specifically, the motivation in the research area, the pursued aims and the main contributions are briefly described. Finally, the chapter concludes with an overview of the structure and contents of the thesis.

1.1. Motivation

Time series modeling is important in several economical sectors like marketing, energy, telecommunications, etc. Time series forecasting can help in operative and strategic approaches. In the electrical sector, energy demand forecasting is a topic of interest to the different members of the electricity market (regulators and utilities) because they provide valuable information to carry out electrical studies. These studies are used to set the balance of load in electrical circuits. In the marketing sector, the demand and supply curve forecasting gives information about possible campaigns to catch more customers depending on the season. In the transport sector, the analysis of the time series related to the flow of people provides information on the required operational dimensioning.

Some time series present changing characteristics due to population growth, variation in weather conditions and modifications in their structure. This makes it necessary to develop models that consider all these factors in order to increase reliability. This type of forecasting models must provide information to understand the phenomenon to be modeled and the factors that affect its behavior [1].

The demand for electrical energy, the flow of people and the flow of data are time series with non-stationary characteristics and their affectation by exogenous variables (weather, the type of day, types of user, etc.) require the integration of multiple statistical techniques together with computational intelligence techniques [2]–[5].

In order to model the dynamics of time series related to power consumption, vehicular traffic and energy price, several authors have proposed different conventional and unconventional statistical modeling strategies. Among the most widely used conventional statistical modeling techniques are the ARIMA (AutoRegresive Integrated Mobile Average) models [6]–[8]. However, this type of technique has some limitations such as low adaptability, limited forecast scope and difficulty of setting the model parameters due to the non-stationary characteristics of the time series. Furthermore, the time series related to power consumption, vehicular traffic and energy price mostly showed non-stationary characteristics which motivates the use of techniques such as SVM (Support Vector Machines) [9], [10], FL (Fuzzy Logic) [11], [12], MC (Markov Chain) [13] and ANN (Artificial Neural Networks) [2], [3], [14]. Some authors have proposed different alternatives that combine computational intelligence techniques and exogenous variables such as weather [15]–[17]. Other authors such as [18], [19] propose making use of grouping techniques to create sub-profiles that allow them to have different forecasts to add them together and thus obtain more accurate results.

An evaluation methodology is necessary to provide support for the adaptation and/or maintenance of various time series modeling techniques automatically. A performance metric is required to identify when a model should be retrained or replaced by another, anticipating the detriment of its performance, that is, without having to wait for forecast performance to drop below the desired level. In addition, in case of not achieving an improvement in the performance of the current model, a metric is required that allows evaluating which of the available modeling techniques in the knowledge base has the best performance.

The proposed research includes a methodology with a forecasting performance metric based on computational intelligence that allows updating or replacing the structure and parameters of a selected model before the performance decreases to a minimum level setted. When a better performance can not be achieved with the retrained model, it is possible to select another available model in the knowledge base through a novel metric proposed in this thesis. In order to test this proposal, several time series data of economic sectors with a changing behavior (trend and seasonality) are used to measure the adaptability of the proposed methodology.

1.2. Objectives

This work is focused on developing algorithms based on computational intelligence techniques capable of providing support to the adaptation and/or maintenance to the time series models automatically.

- **Problem**: To establish a metric or decision module that indicates when a model requires retraining or even a modification without having to wait for the performance to fall below admissible levels.
- **General Objective**: To develop and implement an adaptive methodology based on computational intelligence for time series modeling that allows maintaining the performance for the constructed models.
- **Goals:**
  - ✓ To characterize and select the significatives variables to the time series behavior.
  - ✓ To identify the more suitable model to carry out a time series forecasting.
  - ✓ To establish when the model must be retrained without affecting the performance of the forecasting.
  - ✓ To define a comparison metric that allows to carry out the comparison and selection of a suitable model for the time series under study.

1.2.1. Thesis Question

The principal question addressed in this dissertation is:

Is it possible to define a methodology based on computational intelligence that allows to identify if a model should be retrained or even modified in order to maintain a suitable performance?

1.2.2. Approach

In this research, an adaptive methodology based on computational intelligence for time series is proposed. This approach includes an auditor, a performance metric for the selection of the most suitable model and the use of hyperparameters in order to optimize the training process and the available models.

The auditor module has, as its main task, to indicate in a prescriptive way when a model requires to be adjusted/retrained. For this stage, an intelligent classification system is proposed to compare the characteristics and the performance of the forecast given by the model in order to indicate if there are some changes that cause deviations with respect to actual time series.

On the other hand, some time series have non-stationary characteristics. This condition causes that the statistical characteristics may change from time to time and it also makes the model parameters lose validity when there is the possibility to find another techniques with a better performance [20], [21]. Given this, it has been proposed a performance comparative metric that allows the selection of an appropriate technique within the knowledge base. The modeling of conventional statistical techniques (e.g. ARIMA) and non-conventional (e.g. neural networks and vector support machines) are commonly used in these cases.

Another aspect to be considered within the time series modeling is the definition of the parameters to vary during the training process. To minimize the time required in training, the definition of an experimental design is proposed to identify the significant factors in the performance of the obtained model. To achieve a better performance in the training process, the integration of hyperparameters is proposed in order to optimize the set of parameters to be evaluated.

1.3. Contributions

This thesis makes the following contributions:

- *An adaptive methodology based on intelligence systems for the modeling of the time series that automatically considers the statistical characteristics of the data* *in order to establish the significant variables and ensemble a suitable model for the time series under study.*

- *To evaluate the performance of the obtained models, an auditor module based on an intelligence system has been developed; it will be capable to indicate when a model is losing validation and/or requires retraining or even a change.*
- *To automate the assembling process of the time series models, it has been developed a tool that integrate the characterization of the variables associated with the time series and the auditor for a continuous evaluation of the models performance. In addition, a selection criterion is added that will allow to identify which one of the available models (SVR, RTE, ANN y LSTM) is the more suitable for the characteristics of the time series.*

1.4. Reader's Guide to the Thesis

The following is a general description of the contents of this dissertation. This PhD thesis is mainly organized in three parts distributed by chapters.

Part I: Introduction and Related Works

Chapter 1 presents a motivational introduction on the main topics, objectives and contributions regarding this dissertation.

Chapter 2 gives a general overview of the background information regarding time series modeling and techniques commonly used which are required to develop the proposed approach described in chapter 4 and 5.

Chapter 3 provides a general survey of the most relevant work related to the research addressed in this thesis.

Part II: Proposed Approach

Chapter 4 describes the formal aspects of the adaptive methodology to time series modeling presented in this thesis.

Chapter 5 presents the implementation of the approach proposed in chapter 4. It also contributes to complete the description of the proposal.

Part III: Results and Conclusions

Chapter 6 provides experimental results of the implemented approach. An analysis of results is presented to evaluate the performance of this proposal in simulation runs.

Chapter 7 discusses and analyzes the results, summarizes the conclusions and contributions of the thesis and outlines the most promising directions for future work.

Chapter 2

Background Information

This chapter introduces and reviews general concepts of time series modeling, the box-jenkins methodology and the computational intelligent systems required for developing the proposed approach.

2.1. Time Series Definition

A time series is a sequential set of data points, measured typically over successive times. It is mathematically defined as a set of vectors $x(t), t = 0,1,2,\ldots$ where t represents the time elapsed [22]–[24]. The variable $x(t)$ is treated as a random variable. The measurements taken during an event in a time series are arranged in a proper chronological order.

A time series with records of a single variable is defined as univariate but if records of more than one variable are considered, it is defined as multivariate. A time series can be continuous or discrete. In a continuous time series, the observations are measured at every instance of time, whereas a discrete time series contains observations measured at discrete points of time. The temperature, flow of a river and concentration of a chemical process could be recorded as a continuous time series. On the other hand, the population of a given city, the production of a company, exchange rates between two different currencies may represent discrete time series. Usually in a discrete time series, the consecutive observations are recorded at equally spaced time intervals such as hourly, daily, weekly, monthly or yearly. As mentioned in [23], the variable being observed in a discrete time series is assumed to be measured as a continuous variable using the real number scale. Furthermore, a continuous time series can be easily transformed to a discrete one by merging data together over a specified time interval.

2.2. Time Series Components

Time series are generally affected by the four main components: trend, seasonal, cycle, and randomness. The trend component refers to the time series own characteristics to increase, decrease and stationary over time, e.g., increasing series related to the growing of the population, power consumption and decreasing series associated to the mortality rate during pandemics time. The seasonal component has the variations presented within the time series in a year. These effects are related to weather factors, users and consumption habits, e.g., winter clothing sale rates, energy consumption. The cycling component is related to medium-term variations that repeat themselves in a two-plus years' period, e.g., time series related to business and selling stages. Finally, there is the random component that describes all the phenomena that do not have a clear pattern. These variations are caused by external factors like natural disasters, pandemic or political causes.

Now, depending on the interaction among the four components, it is possible to define two different models [23]:

- Additive model

$$y(t) = T(t) + S(t) + C(t) + I(t) \tag{2.2-1}$$

- Multiplicative model

$$y(t) = T(t) \cdot S(t) \cdot C(t) \cdot I(t) \tag{2.2-2}$$

where,

$y(t)$: observed time series.
$T(t)$: trend component.
$S(t)$: stationary component.
$C(t)$: cycling component.
$I(t)$: irregular component.

The time series are modeled from a multiplicative model when it is figured out that its components are not necessarily independent, this means, there is affectation among the components, which is opposite to the additive model in which independence among the components is assumed.

2.3. Introduction to Time Series Analysis

A suitable model is the one that adjusts to the time series, and its parameters are estimated from the known values. The procedure to determine that a model is fitted to a certain time series is known as an analysis of a time series.

In order to build the models for time series forecasting, it is necessary to collect and analyse the historical data to get the parameters for the data generator process that allows reliably describing the phenomenon of interest [25], [26]. The analysis of the time series works when there is no visible evidence of the patterns within the observation or it does not have a mathematical model to describe truthfully the behavior of the phenomenon. The statistical analysis process works as a platform to build models that are used to forecast the time series. The achieved results are used for the decision-making process, operational sizing (technical supporting, marketing, etc.), potential customers (energy price, vehicular traffic, etc.). For this reason, the development of reliable and accurate models, minimize the economical loss or the oversizing of resources.

2.4. Concept of Stationarity

Stationarity is defined as the characteristic of the time series that shows the temporal dependence of the statistical parameters, that is, a time series is stationary when the mean and variance parameters remain invariant over time. This characteristic is suitable to build a forecast model because it reduces the mathematical complexity.

A process $\{x(t), t = 1, 2, 3, \dots\}$ is stationary in a strict sense, if the joint probability distribution function $\{x_{t-s}, x_{t-s+1}, \dots, x_t, \dots x_{t+s-1}, x_{t+s}\}$ is independent from t to s. To sum up, a stationary process strictly requires that the joint probability distribution function of any random variable would be independent from time. On the other hand, a stochastic process is stationary in a weak sense of order k, if the statistical moments of the process until that moment depend only on the time differences and not on the occurrence moment of the data that is used for calculating the moments. For instance, a stochastic process $\{x(t), t = 1, 2, 3, \dots\}$ is a second order stationary, if it is independent in time on the mean, the variance and the covariance and its values depend only on s.

A stochastic and stationary process in a weak sense is not necessarily stationary in a strict sense. The Dickey-Fuller test is used to validate the existence of unitarian roots in the serie [27]. Stationarity concept is widely used between time series modeling because it helps to simplify this process on a theorical and practical level.

One of the desired features within the time series is stationarity; however, it is not always possible to have this possibility. Time series with a wide history window present a greater probability of presenting non-stationary characteristics due to the presence of trend or heteroscedasticity. However, the non-stationary time series can be setted such that their statistical properties adjust weakly or strictly to the stationarity requirements. This requires several transformations associated with the mean or variance, e.g., differentiation to suppress the non-constant trend or the application of power transformations to stabilize the variance.

2.5. Model Parsimony

The selection of suitable models to describe the phenomenon of interest is not an easy task and it is possible to fall into associated limitations with overfitting and complexity. In this sense, it is very common to evaluate different model alternatives to be considered with different quantity of parameters, which imply an increase in the complexity of the model by itself. The conception of the model becomes difficult in interpretation and the possibility of overfitting can limit the generalization when facing new data. Some authors recommend the principle of *Ockham* razor, which expresses that in similar conditions the simplest explanation is the most probable [25], [26]. This refers to a model with a lesser quantity of parameters that can provide a suitable adjustment to the data series with a minor loss of precision.

Nevertheless, the most complex part in the design of the models is the delimitation of the possible alternatives to be evaluated, starting from the related assumptions with the characteristics of the phenomenon under study. This process may fall into overfitting, which limits the applicability of the selected model to the forecast tasks;

this is where the Ockham's razor principle is widely used to overcome these challenges.

2.6. Time Series Modeling

The time series modeling process is an arduous and complex task that begins with the identification of significant factors within the behavior of the data. In the modeling process, the selection of the most convenient technique must be analyzed. The consideration of conventional mathematical models or computational intelligence techniques depend on the linear or nonlinear nature of the data series and the relationship among the variables that describe the phenomenon or process.

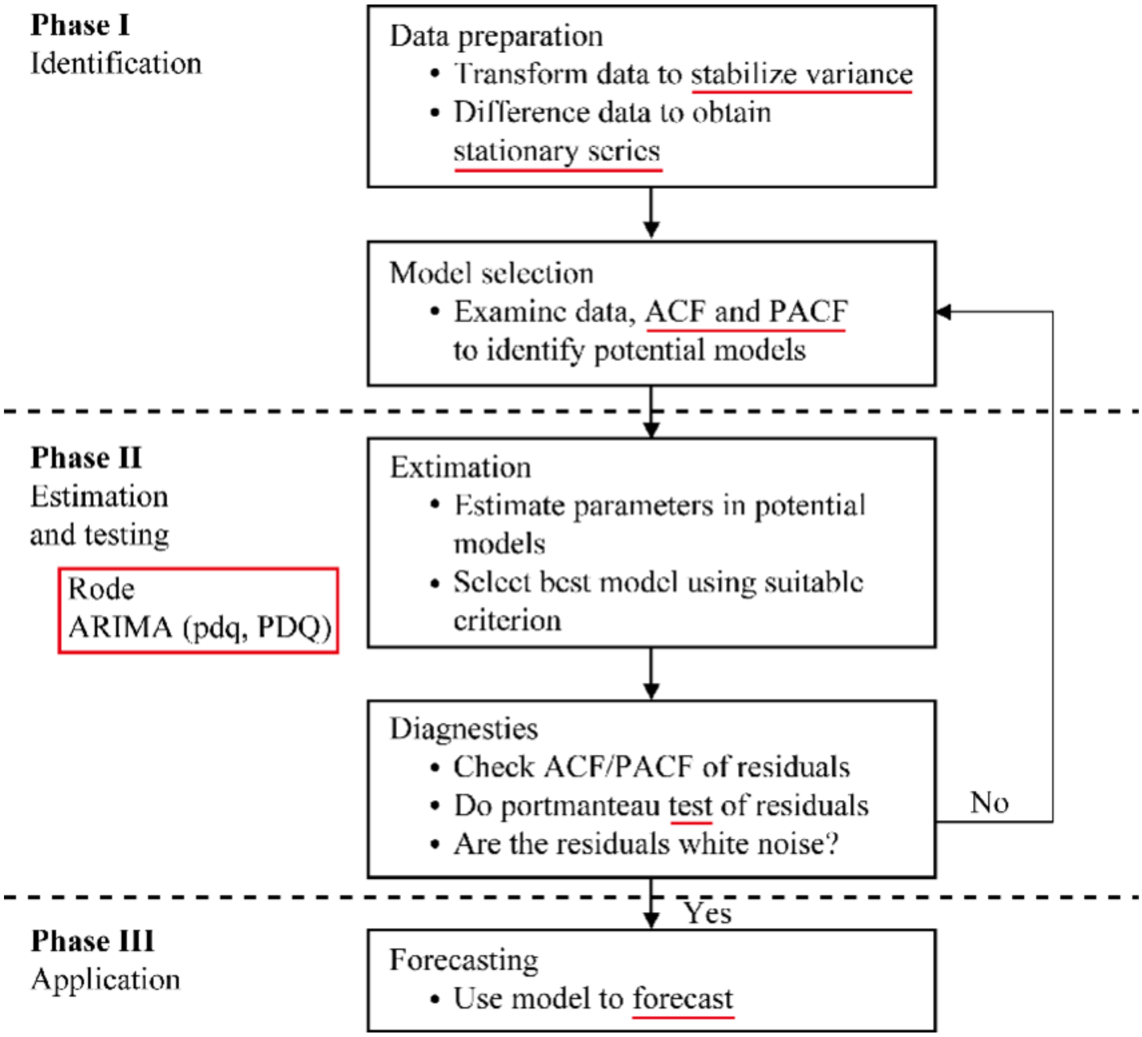


Fig. 2.6-1. Box-Jenkins Methodology [28].

In general, the time series model may have many shapes and may represent different stochastic processes. In literature, there are two kinds of linear models widely used; these are the autoregressive (AR) and the moving average (MA). The AR and MA models are the basis for the assembly of the autoregressive moving average (ARMA) and integrated autoregressive moving average (ARIMA) models. The ARIMA models are used in those time series with non-stationary characteristics in the mean that require a transformation that cancels the trend effect. There is a variation of the ARIMA for stationary time series known as SARIMA. For the design and building of ARIMA models and all their configurations, it is recommended to apply the Box-Jenkins principle (see Fig. 2.6.1).

In a practical level, the application of linear models may be very attractive for its simplicity and easy interpretation, nevertheless, many of the exposed patterns by the time series are nonlinear, for instance, econometrics time series that describe market volatility and the energy demand affected by weather conditions. Under this consideration, it is possible to find different alternatives in the state of art, among the most populars are Autoregressive Conditional Heteroskedasticity (ARCH) models [29] and its variations like Generalized ARCH (GARCH) [30], [31], Exponential Generalized ARCH (EGARCH) [32], the Threshold Autoregressive (TAR) model [33], the Non-linear Autoregressive (NAR) model [34], the Nonlinear Moving Average (NMA) model [35].

It is necessary to guarantee that the stationary assumption in the series is fulfilled at least in a weak sense [36], according to the definition of the autoregressive models. Below it will see a route sheet for the application of the Box-Jenkins methodology [25], [26].

2.1.1. Stationary in mean

In the first stage, it is necessary to determine if the time series is stationary in mean, e.g., in the Fig. 2.62, it is possible to highlight the presence of an increasing trend due to a variation of the mean over time.

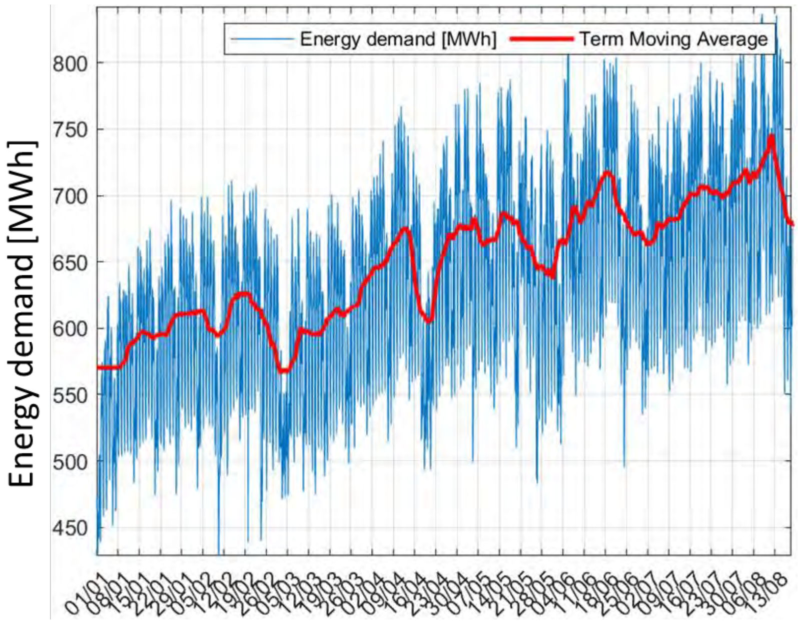


Fig. 2.6-2. Time series of an electrical demand.

- *Deterministic trend*

The main characteristic of the trend component versus the irregular one is to present permanent effects over the time series y_t. To all those series that do not show a trending component, a pure autoregressive model (AR(1)) would be enough to describe its behavior. This is due to the stationary assumption being fulfilled in a weak sense. The Equation 2.6-1 describes a process with those characteristics:

$$y_t = a_0 + a_1 y_{t-1} + \varepsilon_t, \quad to\ |a_1 < 1| \qquad (2.6\text{-}1)$$

Where,

ε_t: White noise.

From the Equation 2.6-2, it is easy to include the trending component in the series:

$$y_t = a_0 + \underbrace{a_1 t}_{trend} + \varepsilon_t, \quad to\ |a_1 < 1| \qquad (2.6\text{-}2)$$

The trend component can be modeled through linear, quadratics, exponentials, logarithms, among other terms. However, it is clear that it would be assuming a zero level of uncertainty because it would have certainty in the value of the future trending.

Similarly, it is possible to combine the autoregressive model with the deterministic trend to model another kind of process called stationary process over a trend (see Equation 2.6-3).

$$y_t = a_0 + a_1 y_{t-1} + \underbrace{bt}_{trend} + \varepsilon_t, \;\; to \; |a_1 < 1| \qquad (2.6\text{-}3)$$

In this case, the process is dominated by the y_t variant, so it is almost impossible to determine, in a graph, the temporal evolution of a pure deterministic trending model [25], [26].

- *Stochastic trend*

The concept of trend is scalable from a stochastic perspective except that its statistical properties do not keep themselves invariants and constants over time. In this case, it is necessary, like in its deterministic counterpart, to define the trending component but of a stochastic nature [25], [26].

In order to model a process with a stochastic trend, a simple random step process is used:

$$y_t = y_{t-1} + \epsilon_t \rightarrow \Delta y_t = \varepsilon_t \qquad (2.6\text{-}4)$$

The solution of the Equation (2.6-4) is:

$$y_t = y_0 + \textstyle\sum_{i=1}^{t} \varepsilon_i \qquad (2.6\text{-}5)$$

The Equation 2.6-5 allows showing a stationary process in mean by definition:

$$E[y_t] = E[y_0 + \textstyle\sum_{i=1}^{t} \varepsilon_i] = E[y_0] = y_0. \qquad (2.6\text{-}6)$$

However, the variance is not constant since its expression is:

$$V[y_t] = E\big[y_t - E[y_t]\big]^2 = E[y_0 + \textstyle\sum_{i=1}^{t} \varepsilon_i - y_0]^2 = E[\sum_{i=1}^{t} \varepsilon_i]^2 = E[\varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_1 \varepsilon_2 + \cdots + \varepsilon_t^2] = E[\varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_t^2] = t\sigma_\varepsilon^2$$

It is clear that the variance increases over time. However, the most interesting of this approach is the permanent effect and trending that have every one of the shocks (ε) over y_t.

A process with a deterministic and stochastic nature called random walk (derivative component) is described through the Equation 2.6-7 [25], [26]:

$$y_t = a_0 + y_{t-1} + \varepsilon_t \rightarrow \Delta y_t = a_0 + \varepsilon_t \quad (2.6\text{-}7)$$

The described process by the above expression will experiment a constant variation defined by the term a_0 since the generic solution to this is the Equation 2.6-8:

$$y_t = y_0 + a_0 t + \sum_{i=1}^{t} \varepsilon_i \quad (2.6\text{-}8)$$

The serie y_t is influenced by the deterministic term $a_0 t$ and the random permanent effect $\sum_{i=1}^{t} \varepsilon_i$. In this case, a process which behavior is adjusted to the above expression will not be stationary neither in mean nor in variance.

$$E[y_t] = E\left[y_0 + \sum_{i=1}^{t} \varepsilon_i\right] = E[y_0] = y_0.$$

However, the variance is not a constant since its expression is:

$$V[y_t] = E\big[y_t - E[y_t]\big]^2 = E[y_0 + a_0 t + \sum_{i=1}^{t} \varepsilon_i - y_0 - a_0 t]^2 = E[\sum_{i=1}^{t} \varepsilon_i]^2 = E[\varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_1\varepsilon_2 + \varepsilon_1\varepsilon_3 + \varepsilon_t^2] = E[\varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_t^2] = t\sigma_\varepsilon^2 \quad (2.6\text{-}9)$$

Equation 2.6-9 allows to show that a random model with derivative results is similar to a deterministic trending for a small number of samples. However, for large samples, a random process with a derivative trend can present noticeable variations over the trending line. This makes the stationary validations in a graph level generate confusions hence a test to detect must be used.

2.1.2. Stationarity in variance

Now, in order to verify the stationarity in variance is not enough a graphic analysis since it may fall in misjudjustments. For a higher level of confidence, it is commonly used the Dickey-Fuller (*DF Test*) or its “extended” version Dickey-Fuller extended (*DFE Test*). It is about a no “stationarity” contrast, it means, the null hypothesis is precisely the presence of a unitary root in the process of generating data from the analysed series [25], [26].

The Dickey-Fuller test is used to verify if a series y_t follows a non-stationary random walk or alternatively a stationary autoregressive process of first order.

$$H_0: a_1 = 1 \rightarrow y_t = a_0 + y_{t-1} + \varepsilon_t, \qquad (y_t \; non \; stationarity \; in \; variance)$$

$$H_1: a_1 = 1 \rightarrow y_t = a_0 + a_t y_{t-1} + \varepsilon_t, \qquad (y_t \;\; stationarity \; in \; variance)$$

It is possible to notice how the series variance increases over time, allowing to justify the selection of a stochastic trending model with derivative, it means, it is a process that describes a deterministic trend with random oscillations over the trending line. In this case, it is easier to notice that the trend describes a linear behavior.

In Fig. 2.6-3, it is shown that the presence of the trending and stationary components are inherent to the time series taken as an example. The stationarity assumption is not fulfilled due to the evolution of the characteristic over time.

Through the *Dickey-Fuller test*, it is possible to contrast the null hypothesis from the presence of the unitary root (see Fig. 2.6-4). Fig. 2.6-4 shows an example where it is not possible to reject this hypothesis because the time series is not stationary.

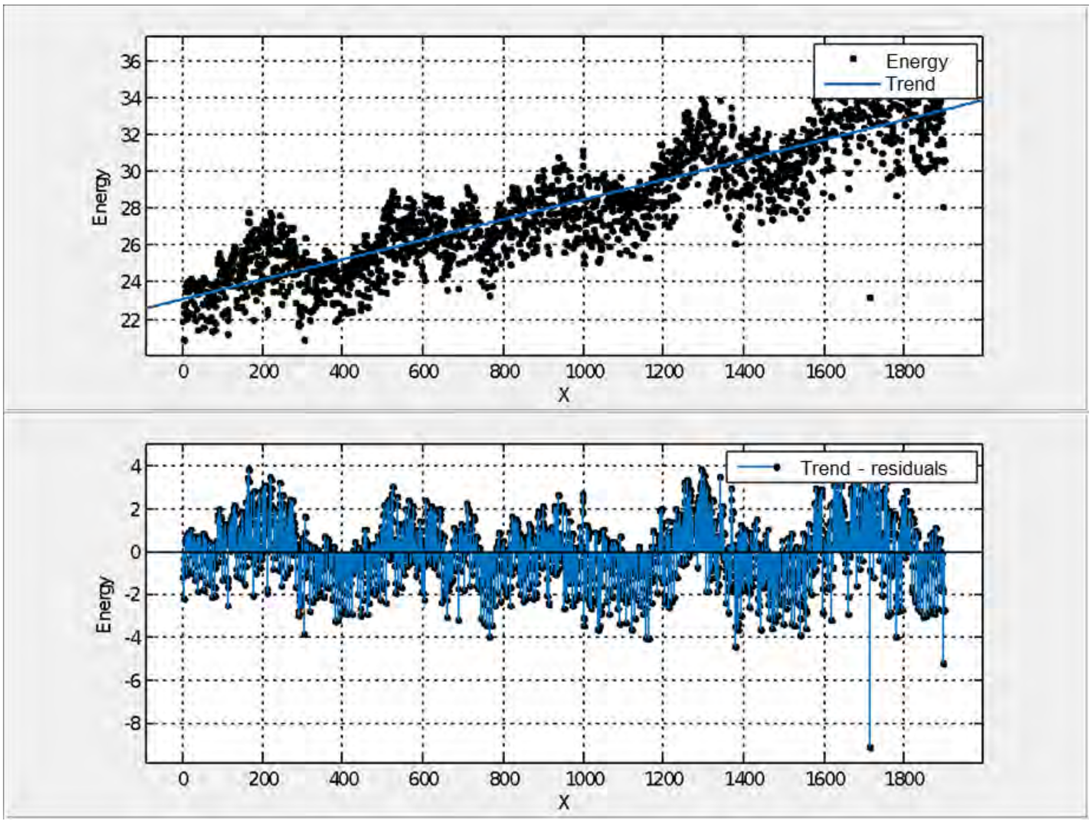


Fig. 2.6-3 Trend component in a time series.

```
D-F increased contrast for Demand_Aj
including 21 delays from (1 - L) Demand_Aj
(the maximum was 25, the criterion AIC) sample size
1878
unit root null hypothesis: a = 1

Contrast with a constant
model: (1-L) y = b0 + (a-1) *y(-1) * ··· + e
estimated value of (a - 1) : -0,0114579
contrast statistic: tau_c(1) = -1,6045
value p asymptotic 0, 4803
first order self correction coefficient of e: 0,001
delayed differences: F(21, 1855) = 62,429 [0,0000]
```

Fig. 2.6-4 Dickey-Fuller test to a time series.

Exponential logarithmic transformation is recommended to stabilize the variance or simply the first differences to fulfill the assumption of stationarity in a weak sense, in order to reverse the non-stationarity in the media and the variances of the time series. Fig. 2.6-5 shows as the first difference of the series remove the trending component.

The external values that are shown are atypical situations related to the phenomenon to be modeled. The *DF* test is used to reinforce a conclusion in which the *p-value* under 0.05 allows to reject the null hypothesis from the presence of the unit root reinforcing again the stationarity of the transformed series. The results from this test are shown in Fig. 2.6-6.

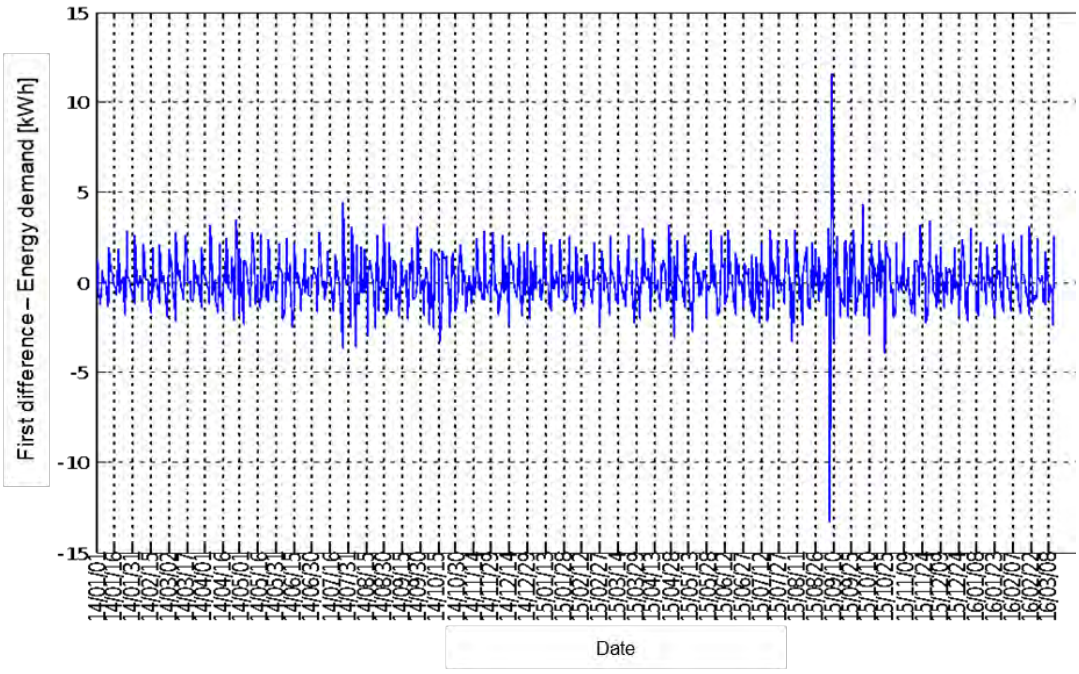


Fig. 2.6-5 First difference in a time series.

```
D-F increased contrast for d_Demand_Aj
including 25 delays from (1 - L) d_Demand_Aj
(the maximum was 25, the criterion AIC)
 sample size 1873
unit root null hypothesis: a = 1

Contras with a constant
model: (1-L) y = b0 + (a-1) *y(-1) + ... + e
estimated value of (a - 1) : -5,52034
contrast statistic: tau_c(1) = -12,9258
value p asymptotic 1,042e-028
first order self correction coefficient of e:-
0,007
delayed differences: F(25, 1846) = 55,812 [0,0000]
chart 11. Dickey- Fuller test for the first
differences of the daily energy demand
```

Fig. 2.6-6 Dickey-Fuller test to a time series.

It is necessary to apply stationary models that allow the transformation and model the time series in the right way, given the non-stationary behavior of the time series. The selection of the parameters can be established by making use of the graphic method, which involves an analysis of the total auto-correlation function (*ACF*) and the partial auto-correlation function (*PACF*).

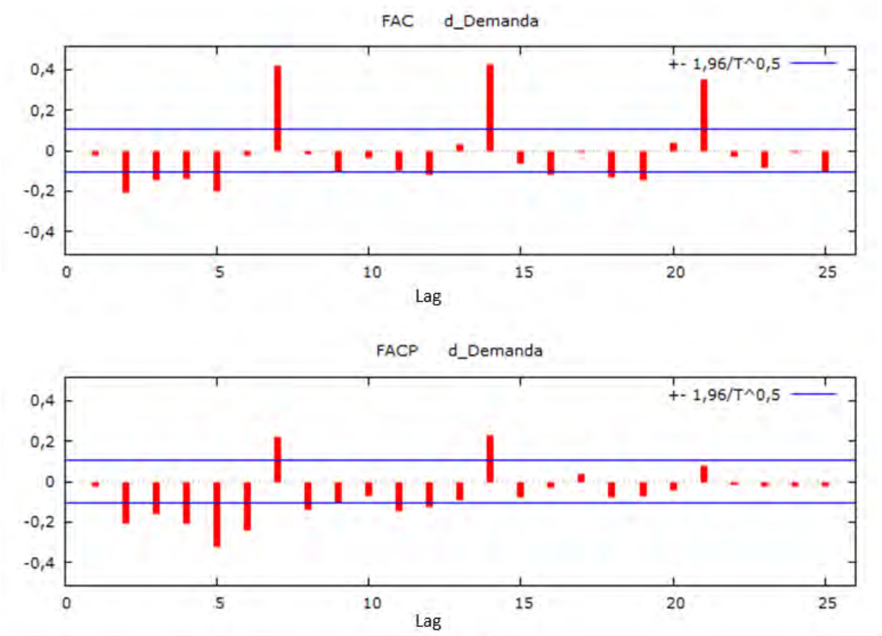


Fig. 2.6-7 Correlation analysis of a time series.

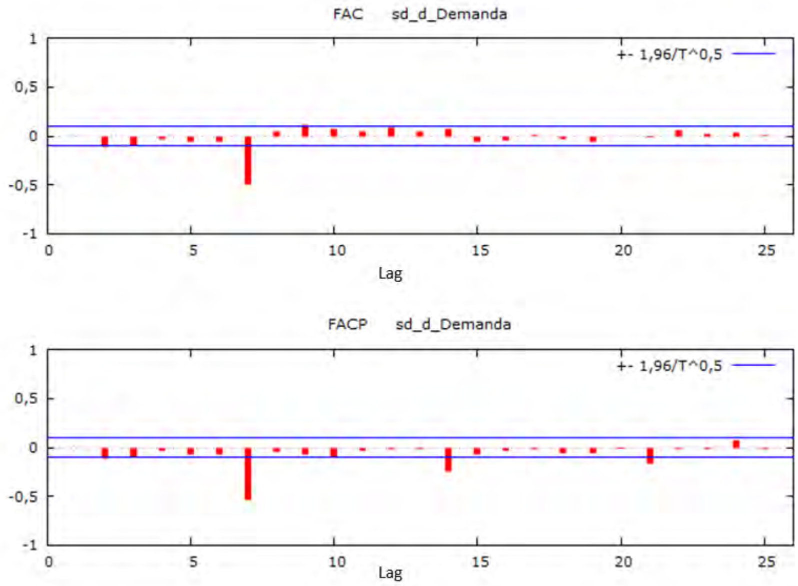


Fig. 2.6-8 Correlation analysis after applying the first difference to the time series.

One of the main characteristics of the Box-Jenkins methodology is that the model order *AR* and *AM* can be determined through the *AFC* and *PAFC* [37].

Fig. 2.6-7 shows the total and partial autocorrelation function of the time series, in this case, a clear seasonal effect independent of recent values is evidenced. In Fig. 2.6-8, the same analysis is applied to the first differences of the time series. The slow decline and the number of values out of the critical limits of the *PAFC* reaffirm the need of one or more *MA* non-stationary components [25], [26]. It is observed that the *AFC* is canceled in lag 7 that allows detecting the presence of a stationary component (SMA).

Considering the possible configurations of the ARIMA models, it must validate each one of the alternatives and evaluate its performance through the *Akaike criterion* (AIC) [1] and the *Schwartz criterion*[2].

2.7. Computational Intelligent System

From a broader view, it can be defined as a system which proposes solutions to complex cases where conventional algorithms are unable to solve [38]. This concept

[1] Akaike information criterion (CIA): when this value is lower then the model has a better performance.
[2] Schwartz information criterion (BIC): when picking from several models, the one with the lowest BIC is preferred.

may be subject to ambiguities, and therefore, it is necessary to clarify that a computational intelligent system must fulfill the following criteria [39].

– Only deals with the numerical data (low level).

– It has a pattern recognition component.

– Activities of the human being as problem solving, decision making and learning.

This type of system combines elements of learning, adaptation, evolution and fuzzy logic to design programs with some level of intelligence. It is based on experience and provides the right decisions despite limitations [40].

Computational intelligence applied in time series modeling has allowed developing several applications oriented to time series forecasting, risk analysis, customer profiling, smart grids, etc.

2.7.1. Artificial Neural Networks (ANN)

This type of soft-computing technique is based on biological neural networks performance of the human brain. Therefore, they are units called neurons and are able to learn from experience, generalize from previous examples and to abstract the main characteristics of a data series. Thus, ANN, change their behavior depending on the environment, provide correct answers to inputs with small variations, among other advantages [41], [42].

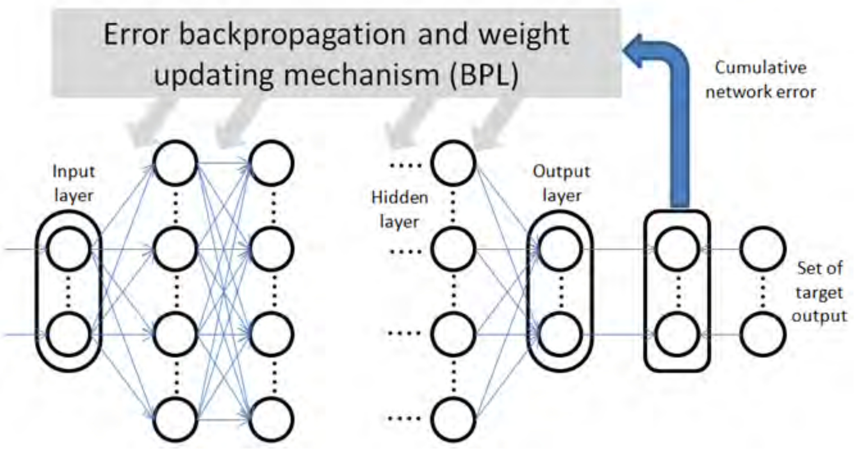


Fig. 2.7-1 Neural network topology.

Fig. 2.7-1 shows the basic structure of a neural network to train with a back propagation algorithm [43]. Back propagation is a method of training the neural network using three or more layers, commonly called: topology of three layers. The input layer is the only processing unit which receives inputs from the outside, therefore, it serves as distributors and not to perform a calculation operation. The layer (in the case of a topology that is more than three layers) between the lower and the upper layer is called hidden layer, the processor units are interconnected with adjacent layers and to carry out the operations calculation. Finally, the output layer represents the response of the network. This topology is supervised, i.e., input patterns matched and target outputs must be known [41], [42].

- *Recurrent Neural Network (RNN)*

Recurrent neural networks, unlike feedforward neural, have connections among the neurons in each one of the layers that allow to get information from past data and the network outcome (feedback).

Fig. 2.7-2 shows the simplest structure of a recurrent neural network where it stands out how the input produces an outcome that is a feedback to the same neuron.

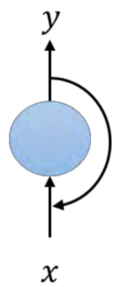


Fig. 2.7-2 Simple structure of a recurrent neural network.

In a neural network case with two or more neurons, each neuron receives at every instant of time, the input of the previous layer as well as the output of the previous instant of time from the same layer.

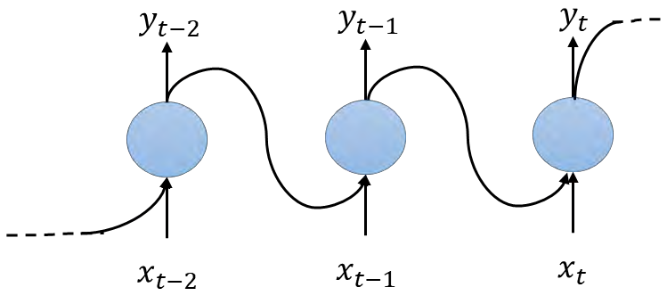


Fig. 2.7-3 Neural networks with two or more neurons.

Like this, each layer of recurrent neurons has two sets of parameters at its input. The input data received from the previous layer and the output vector of the previous instant (see Equation 2.7-1).

$$y_t = f(wx_t + uy_{t-1} + b) \qquad (2.7\text{-}1)$$

Where,

x: Previous layer sequence input.

w:Matrix of weights at the instant of time t.

u: Matrix of weights of the state of the network at the previous instant of time y_{t-1}.

b: bias.

Given the characteristic of retention of previous data or the instants of time by any of the neurons belonging to the recurrent neural network, it is possible to say that this kind of networks have a memory cell. This characteristic makes them attractive for sequential data modeling since it allows to evaluate the value to forecast considering the data or/and recent changes in the time series. The "internal memory" of these recurrent networks allows data from the current context, which is not allowed in other kinds of neural networks where their forecast is only based on the obtained weights during the training process. A valuable example of this kind of networks is commonly found in automatic translation tasks [44], the modeling of language [45], speech recognition [46], displacement of people [47], electrical energy consumption forecasting [48], vehicular trajectory forecasting [49].

- *Long-Short Term Memory*

The architecture of LSTMs is composed of units calledmemory blocks. Memory block contains memory cells withself-connections storing (remembering) the temporal state ofthe network in addition to special multiplicative units calledgates to control the flow of information. Each memory blockcontains an input gate to control the flow of input activationsinto the memory cell, an output gate to control the output flowof cell activations into the rest of the network and a forget gate (Fig. 2.7-4).

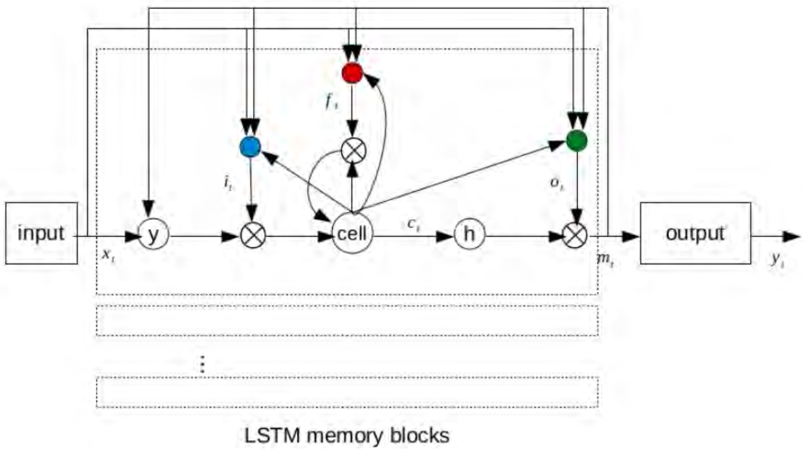


Fig. 2.7-4 LSTM node [50].

The forget gate scales the internal state of the cell before adding it back to the cell as input through self recurrent connection, therefore adaptively forgetting or resetting the cells memory. The STM architecture also contains peephole connections from its internal cells to the gates in the same cell to learn precise timing of the outputs [50].

- *Selection of a Proper Network Architecture*

After selecting the neural network structure to be used, the challenge is in determining the optimal quantity of parameters to be applied. A suitable model must provide a reasonable error in the training, validation and testing stages without overfitting. As an initial designing stage, it must carry out an exploratory statistical analysis that allows identifying the significant variances that would improve the model performance.

Selecting the best performance network is not an easy task since there is not a formal theory that allows simplifying this task. In the related work, it is recommended to apply theories related to the design of experiments and cross-validation to go through the experimentation zone that includes the greatest variation of the possible parameters for choosing the neuronal network with the best performance.

Another aspect to take into account is the quantity of available parameters and their affectation with the complexity of the model; this is why it is recommended to consider the *Akaike information criterion* in order to penalize the addition of extra parameters. In summary, the selection for the configuration and suitable structure of the neural network is important to the forecast process. The data conditioning, e.g., normalization, re-scaling or transformation also provide an improvement in the model performance based on neuronal networks.

2.7.2. Support Vector Machines (*SVM*)

This technique was initially defined for the classification of problems such as pattern recognition, text characters' identification and facial recognition. However, the authors of this technique found some applications oriented to the function approximation over the following years. *SVM* is based on the *SRM* principle (Structural Risk Minimization) that has as an objective to find a decision rule with a generalization capability from selecting a particular subset of training data called support vectors (see Fig. 2.7-5). In this method, an optimal separating hyperplane is built after non-linearly mapping the input space into a feature space of a larger dimension characteristic [51].

Unlike the neural network, the training of the support vector machine is equivalent to a quadratic programming problem with a linear constraint that guarantees a unique solution and a global optimal. The main disadvantage from this technique is the required computational cost over the training process for a big data set of training that increases the temporal complexity of the solution.

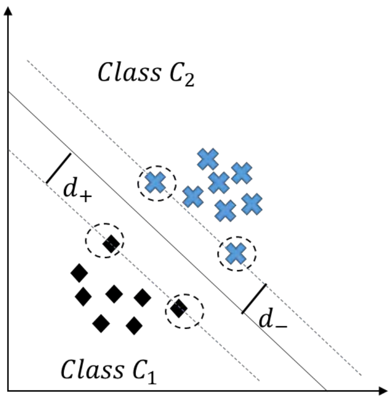


Fig. 2.7-5 General mapping through *SVM*.

- *Binary classification*

The process of binary classification requires the approach of a hyperplane that maximizes the separation of the two classes for the training set. In order to understand the operating principle for this technique, it takes two sets of linearly separable training data, which will be classified in one of the two classes using a linear hyperplane [51].

To obtain the most suitable linear hyperplane to separate the training data into the corresponding classes, the possible infinite solutions of separation are proposed and the calculated total margin is evaluated from the expression $M = d_+ + d_-$. To get a precise alternative with a good generalization capacity, the hyperplane that mazimizes the total margin is selected. This is known as a maximum margin hyperplane and it is obtain when $d_+ = d_-$. The training points for both classes that are closer to the selected separation hyperplane are called support vectors.

- *SVM for regression*

SVM must only be adjusted in the input, to forecast the time series data, passing from dichotomic data to definite data within the set of real numbers, keeping the same characteristics that define the classification algorithm (maximum margin). Since the output is a real number, it provides itself with a greater complexity because the algorithm for optimizing and obtaining the maximum margin has infinite possibilities. Thus, it is necessary to implement a margin of tolerance (epsilon) that allows limiting the solution. Despite, the aforementioned variations, the main idea is always the same: to minimize the error, select the hyperplane that minimizes the margin and establish what error is tolerated.

2.7.3. Random forest

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks) [52].

A tree-based model involves recursively partitioning the given dataset into two groups based on a certain criterion until a predetermined stopping condition is met. At the bottom of decision trees are so-called leaf nodes or leaves.

Fig. 2.7-6 shows a recursive partitioning of a two-dimensional input space with axis-aligned boundaries—that is, each time the input space is partitioned in a direction parallel to one of the axes. Here the first split occurred on $x_2 \geq a_2$. Then, the two subspaces were again partitioned: The left branch was split on $x_1 \geq a_4$. The right branch was first split on $x_1 \geq a_1$, and one of its subbranches was split on $x_2 > a_3$. Fig. 2.7-7 is a graphical representation of the subspaces partitioned in Fig. 2.7-5.

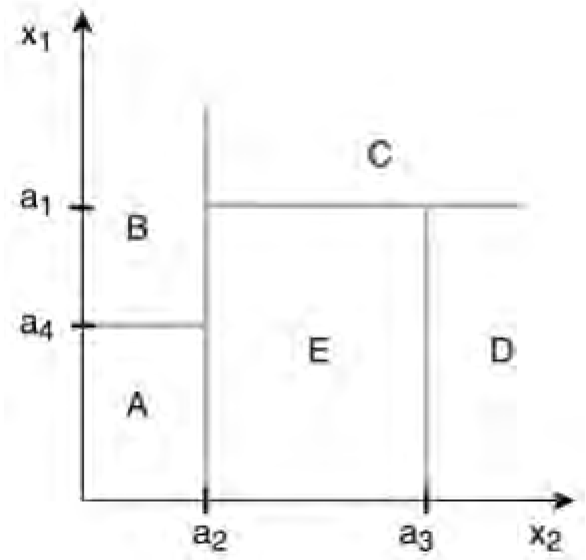


Fig. 2.7-6 Recursive binary partition of a two-dimensional subspaces.

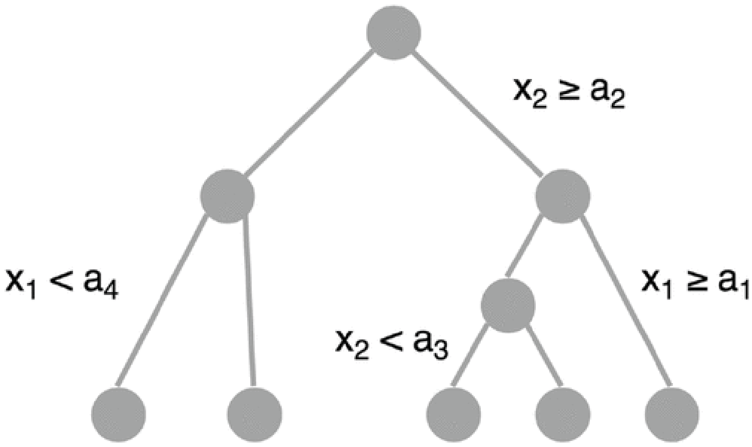


Fig. 2.7-7 A graphical representation of the decision tree in Fig

Depending on how the partition and stopping criteria are set, decision trees can be designed for both classification tasks (categorical outcome, for example, logistic regression) and regression tasks (continuous outcome).

2.8. Forecast Performance Measures

The building and selection stage of the forecast models requires a suitable training and evaluation process of the possible configuration for each technique. To build a particular model of an actual or simulated series is necessary to split the data of the time series in two parts: training set or validation set.

The data located within the training set will be the base to the construction of the desired model. Meanwhile, the validation set can be divided into two: a validation set used to evaluate the model performance obtained over training and a test set that is used to evaluate the performance of the selected model to finalize the training process.

Depending on the set of variables to consider, it is necessary to apply preprocess and transformation techniques on a common scale to all, e.g., normalization, logarithmic transformation, differentiation. Once the result is obtained from the model, the inverse transformation must be applied to the one performed on the variable input to measure the performance on the same units of the original time series.

Commonly used performance metric are described in Equation (2.8-1):

$$e_i = y_i - \hat{y}_i \qquad (2.8\text{-}1)$$

where,

y_i: current value.

$\hat{y}_i$: forecast value.

e_i: error for n data of time series.

In this chapter we shall describe few important performance measures which are frequently used by researchers, with their salient features [53].

2.8.1. Mean Forecast Error (MFE)

This metric is defined by the Equation (2.8-2):

$$MFE = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{2.8-2}$$

Some of the properties are:

- It is a measure of the average deviation of the forecasted values from the actual values.
- It shows the error direction, which is why it is known as forecast bias.
- The effect of the positive and negative errors cancels each other out, which prevent the exact values to be known.
- An error equal to zero does not mean that the forecast is perfect, instead they fit the validation data correctly.
- Extreme values are not penalized.
- It is affected by the data scale and the transformation to be performed.
- It is desired that this value will be close to zero to get a better forecast.

2.8.2. Mean Absolute Error (MAE)

This metric is defined by the Equation (2.8-3):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{2.8-3}$$

Some of the properties are:

- It shows total error magnitude.

- It does not show the direction of the error.
- Depends on the scale and variables transformations.
- The one with the lowest error is preferred.
- Extreme values are not penalized in this kind of error.

2.8.3. Mean Absolute Percentage Error (MAPE)

This metric is defined by the Equation (2.8-4):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{e_i}{y_i}\right| x100 \qquad (2.8\text{-}4)$$

Some of the properties are:

- It is independent of the scale but it is affected by the transformations performed.
- It does not show the direction of the error.
- Extreme values are not penalized in this kind of error.

2.8.4. Mean Percentage Error (MPE)

This metric is defined by the Equation (2.8-5):

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{e_i}{y_i}\right) x100 \qquad (2.8\text{-}5)$$

Some of the properties are similar to those of the *MAPE* except that:

- It shows the direction of the error.
- An error close to zero does not necessarily mean that the forecast is perfect.
- The one with the lowest error is preferred.

2.8.5. Mean Square Error (MSE)

This metric is defined by the Equation (2.8-6):

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \qquad (2.8\text{-}6)$$

Some of the properties are:

- Because positive and negative errors do not cancel each other out, *MSE* give a general idea of the error forecast.

- It penalized the values of extreme errors.
- It is based on the idea that large individual error values are costly than smaller individual error values.
- It does not provide an idea about the direction of the error.
- It is sensible to the changes in the scale and transformations carried out.
- It provides an idea of the forecast error but it is not sensitive or easily interpretable as above measures.

2.8.6. Sum of Square Error (SSE)

This metric is defined by the Equation (2.8-7):

$$SSE = \sum_{i=1}^{n} e_t^2 \qquad (2.8\text{-}7)$$

Its properties are the same as *MSE.*

2.8.7. Signed Mean Squared Error (SMSE)

This metric is defined by the Equation (2.8-8):

$$SMSE = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{e_i}{|e_i|} \right) e_i^2 \qquad (2.8\text{-}8)$$

Some of the properties are:

- Similar to *MSE* with the difference that it keeps the original sign of error.
- It penalizes the large error values presented over the forecast.
- It shows the direction of the total error.
- Positive and negative errors offset each other.
- It is sensible to changes in the scale and data transformation.

2.8.8. Root Mean Square Error (RMSE)

This metric is defined by the Equation (2.8-9):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} e_i^2} \qquad (2.8\text{-}9)$$

Its properties are the same as *MSE.*

2.8.9. Normalized Mean Square Error (NMSE)

This metric is defined by the Equation (2.8-10):

$$NMSE = \frac{MSE}{\sigma^2} = \frac{1}{\sigma^2 n}\sum_{i=1}^{n} e_i^2 \qquad (2.8\text{-}10)$$

Some of the properties are:

- It is a balanced error metric to evaluate the performance of the forecast model.
- The one with the lowest error is preferred.
- Other properties of this kind are similar to those on *MSE*.

Chapter 3

Related Work

This chapter introduces an overview of the main works focused on the topics addressed in this dissertation.

3.1. Preprocessing data

The time series modeling requires the available data fidelity and representability. Alterations in the natural behavior of the time series can induce noise or errors in the mathematical modeling building process. Many authors agree and show that this stage is important to the selection and assembly process of a reliable model. The data in the time series can present gaps, inconsistencies and tendencies with exponential behavior. The actions most commonly implemented by the authors to clean the time series data in order to improve the performance of the obtained models will be shown:

A method for econometric time series forecasting has been proposed in a work by [54]. Four necessary stages have been defined in order to build each one of the mathematical ARIMA models based on the Box-Jenkins methodology. In the first instance, to detect and remove errors and inconsistencies to upgrade the data quality, data pre-processing is added. Besides, all the gaps in the data are identified and filled in with average of the data closest. In the second instance, the detection and correction of the outlier data are integrated as a confidence interval defined by $[\mu - t * \sigma, \mu + t * \sigma]$, where μ: series mean, t: threshold to be determined, σ: standard deviation of the series. In the last stage, those cases where exponential trend and heteroscedasticity are shown, which makes it necessary to perform a logarithmic transformation.

The authors of the work presented by [55] perform a modeling of two time series related to rainfall in the Daning and Zhenshui rivers. To carry out the data conditioning before building the models, four kinds of techniques are proposed and compared: 1) Moving-Average Model (*MA*), 2) Principal Components Analysis (*PCA*), 3) Singular Spectrum Analysis (*SSA*), 4) Diagonal Averaging (*DA*). This preprocessing stage is integrated to a model based on neural networks. The diagonal averaging is to find equal elements in the resultant matrix and then to generate a new element by averaging over them.The obtained results showed the following:

- The effect of the *MA* technique in the performance of the forecasting is not perceptible.
- Applied *PCA* for noise elimination produces a slight improvement in the model performance.
- *SSA* produces a considerable improvement in the forecast model performance.

In the work proposed by [56] arises an alternative based on neural networks and two preprocessing data techniques for the quarterly modeling of the time series considering 48 possible neural models with different historical data, weather variables, trigonometric variables and time indicators as inputs. Parametric and non-parametric analysis are carried out to evaluate and select the best model for each case. In this work, two transformation strategies are proposed: 1) “log”: it consists of applying a logarithmic transformation to the data and 2) “full”: it consists of applying the logarithmic transformation along with a moving average. To remove the data series trend an approximation is made to a linear trend and this component is subtracted from the data. To remove the seasonal component a moving average is applied following a classical additive decomposition model. The parameters to decompose the time series are obtained from the own time series data. Another method to seasonality adjust the series is the one proposed by [57].

Some authors include more sophisticated transformations to the data conditioning and preprocessing such as one proposed by [58] in which they use Empirical Wavelet Transform enhanced with the Artificial Bee Colonial (ABC) algorithm. *EWT* is an algorithm that integrates the formalization of the *EWT* along with the adaptability of the Empirical Mode Decomposition (*EMD*). In this work, *EWT* is used to remove the impact of outliers, making the data more suitable for the *ARIMA* and Extreme Learning Model (*ELM*) forecasting models. To test their proposal, the authors use three datasets of the financial sector. A comparison of the performance (*MAPE*) models (*LSTM, ARIMA, ANN* and the proposed hybrid model) is carried out. A decrease in the *MAPE* is shown because of the data preprocessing. The lowest *MAPE* obtained by the proposal raised here was 0.61% of the proposed model.

The implementation of *ARIMA* models on univariate time series is limited by those with stationary characteristics, the authors [21] propose a logarithm transformation to get the own characteristics of time series in order to build a forecast model optimally. A statistical set of tests are implemented to solve the problem of imprecise and inadequate identification of the non-stationary characteristics of the time series. The proposal is composed by the following stages:

1) Data quality enhancement. At this stage, replacement of missing or outlier data is done using previous and/or late average data.
2) Variance stabilization. In this stage, natural logarithm is applied to the data when the time series is heteroscedastic.
3) Application of the roots unit test to the periodicity identification.
4) Auto-adaptation and processing algorithm of non-stationary time series.

Test data related to monthly electric power generation in China from January 1990 to December 2015 were used. The results obtained showed a better performance of the proposed model with a 2.5% *MAPE* respect to the 4.45% of the *Holt-Winters* model.

3.2. Selection of variables

One of the most important aspects in time series modeling is the selection of the variables to be considered not only by their relevance in the performance of the forecasting, but they also affect directly the reduction of costs associated with the availability of the different variables to be considered. Depending on the interaction among the variables and the structures of the models, it is possible to establish two approaches for variable selection: model-free (it does not depend neither on the structure or the model calibration) and model-based (depends on the model structure) [59], [60]. In the model-free approach, the *Pearson coefficient* is used to determine the existence of the correlation between the input and output of the model. On the other hand, the partial mutual information (*PMI*) brings information about the existence of a linear or non-linear correlation between the input and output data of the model. The model-based approach considers candidates input variables as significant if they contribute to one or more modeling performance indicators. The performance indicators commonly used are the mean square error between the observed and forecasted data and the coefficient of determination (R^2).

In the work presented by [61] three methods for the selection of the most significant input variables for four series of the synthetic data and two series of actual data are explained. The techniques used to carry out the significant analysis are: 1) partial linear correlation (*PLC*), 2) partial mutual information (*PMI*), and 3) a technique based on genetic programming (*GP*). The partial linear correlation provides information on how two covariant variables in a linear way can be analyzed in a scatter plot. The mutual information technique evaluates the dependency between two variables starting from their joint distribution and marginal density. In other words, partial mutual information measures the decrease in the uncertainty of one of the variables as the acknowledgement. Finally, there is the technique based on genetic programming that shares similar characteristics to the genetic algorithm but this one is based on the generation and variation of the tree structures. In this method, an input variable is considered significant if its numbers of occurrences in the generated equations is greater than the established threshold. The results showed that the *PLC* model (due to its low computational cost) and the *GP* model (due to its ability to detect a nonlinear relation) are recommended for the selection of the variables.

In the health sector, time series modeling contributes to the improvement and optimization process. The authors in [62] propose a model based on a neural network to determine the hospitalization rate. For the selection of the variables the *Box-Jenkins* methods has been taken as a reference for auto-regressive processes, in which the partial auto-regressive function chart is used to determine how much information from the past is necessary to be able to explain the future. The partial auto-correlation function helps to identify the highest order of an auto-regressive process. This analysis is carried out in four hospitals obtaining a minimum 1.17% and a maximum 3.95% *MAPE* (Mean Absolute Percentage Error).

The selection of the variables is a fundamental process within the performance of forecasting models. In some time series, e.g., in energy demand, it is possible to find many variables that affect the behavior of power consumption. The authors of [63] take two alternative solutions that help with the selection of variables with the aim of increasing efficiency in the number of variables to be used. The energy and temperature data are used for seven states of US from 2007 to 2011 and 2001 to 2010, respectively. The first method is a holistic one based on the selection from the probability measure error that is consistent with the error measure used for the final evaluation of the probabilistic forecast. The second method is based on the heuristic strategy that takes a shortcut trusting in a measurement error from the selection of variables. The results showed that even though the holistic method slightly exceeds does not dominate the heuristic method. A minimum 2.82% and a maximum 3.19% *MAPE* is obtained.

Energy trading companies require the short-term time series forecasting in order to carry out planning task and electrical studies. The work developed by the authors [64] proposes a hybrid demand forecasting model to forecast using weather, social variables and indicators of business days or holidays, time of day and season of the year. Data from the Australian electric power operator is used along with the data from urban region in Houston, Texas. The analysis of the variables is carried out through the empirical mode of decomposition and correlation analysis. To build the model, time series decomposition is carried out at low frequencies. To compensate for the loss of information during the decomposition process, the implementation of the T-Copula correlation analysis is proposed to estimate the dependence of the demand peaks and the exogenous variables considered. The results showed an enhancement on the model performance when the correlation T-Copula algorithm is incorporated reaching a *MAPE* with minimum of 3.98% and a maximum 4.62%.

3.3. Commonly used techniques for time series modeling

The time series modeling requires a set of actions that allow debugging, characterizing, selecting, and simplifying the data necessary for the building of the different models. Since there is not a unique technique that guarantees the best performance, many authors have presented different works in which the comparison among the different modeling techniques (e.g. *ARIMA*, *ES*, *ANN, SVM, LSTM, RTE*, among others). In other words, the hybrid models have been used in order to take advantage of the strengths and mitigate the weakness of the different techniques. Next, a summary of the works related to these topics will be shown:

The work proposed by [65] focuses on two challenges and proposals for a forecasting method based on a decomposition assembly framework called adaptive sub-series clustering-stacked residual LSTMs-multi-level attention mechanism (ASC-SRLSTMs-MLAttn). The method consists on three stages: decomposition, forecasting and assembly. In the decomposition stage, the original time series is divided into multiple sub-series by using the empirical assembly mode to reduce the complexity and time required for forecasting. In the forecasting stage, the correlated sub-series are grouped for building the models based on neural networks. The model is based on the encoder-decoder architecture integrated to the *LSTM* networks such as the encoder, given its adaptability to capture the dependency between multiple variables and temporal characteristics of the different time series to be analyzed. A multilevel attention mechanism (MLAttn) was added to make full use of the encoded information by the encoder with the purpose to enhance the forecast performance. In the assembly stage, each one of the subseries forecasts is added to obtain the original forecast for the series. To carry out the performance validation of the proposed model, a comparison is made with thirteen other forecasting methods. (*ARIMA*, *SVR*, four models based on feedforward neural networks, Single-layer Gated Recurrent Unit –GRU–, five variant of the model based on Long-Short Term Memory –*LSTM*– neural network, a model based on forecasting framework decomposition-assembly that uses Extreme Learning Machine –*ELM*– technique for the forecasting model). Four different time series from Beijing city were used to evaluate the performance of each one of the models (1- Air quality, 2-

Traffic, 3- Electric power consumption, 4- Solar energy generation). The results showed a better performance of the proposed model compared to the other forecast models. A performance (Root Mean Square Error –*RMSE*–) for each one of the four-time series analyzed in each one of the forecast windows (3h, 6h, 9h and 12h) were tested.

Neural networks have also been explored through univariate approaches, such as the one presented by [66] in which neural networks and an extension of support vector machines are used for nonlinear regression. A collection of consumption data that goes over a period of forty-one (41) years is used for the training of the proposed neural network and the regression adjustment. The input data consist in derivative variables of the consumption data, these are: time, month and seasonality index, consumption data one or two years in advance. The most important aspects to consider within what is presented by [66]. The results (*MAPE*) showed that the performance of the *SVR* model (3.3%) exceeded the one shown by the neural networks (3.9%).

In telecommunication, the network traffic is an important aspect in the service quality. The authors [67] propose the design of a methodology for forecasting 4G network traffic with hourly data. The different curves are initially classified by using the clustering fuzzy c-means technique to enhance the precision of each of the available models. A smoothing exponential model is proposed to enhance the performance of the *LSTM (Long-Short Time Model)* and *ANFIS (Adaptive Neuro-Fuzzy Inference System)*. Two real traffic data series are taken to try out the builded models. The authors use the *Sperman* (R) correlation coefficient to compare the performance of the models, showing a value R = 97.95% for the *LSTM* model while the *ANFIS* model got a value of R = 96.78%.

In the time series modeling, there is not a unique technique that could always guarantee the best performance, hence some authors choose to carry out a comparative analysis among several statistical or intelligent techniques to solve these problems. The author in [68] proposed neural networks for the decomposition and forecasting of the time series. A feedforward topology with sinusoidal activation function and other non-periodic activation function to fit the periodic and non-periodic characteristic of the time series is proposed. To carry out the validation and comparison of the considered models (*ARIMA, SARIMA, SVR, LSTM*, echo-state Network, neural decomposition) four-time series are used: 1) the USA monthly unemployment rate, 2) USA monthly international passengers rate, 3) Ozone concentration in Los Angeles city, USA and 4) Oxygen isotope measurement in India. This proposal showed a better performance regarding traditional methods. MAPE equal to 1.09% for the unemployment series, 45.03% for the passenger's monthly rate, 0.99% for the ozone concentration and 0.214% for the oxygen isotope measurement in India.

In the health sector, information related to the events that could affect the care capacity of the hospital network is of interest to avoid collapses in the hospital network. The work developed by [69] addresses this type of problem with the modeling of a time series related to the patient volume in a Hong Kong medical center. The data are collected over a period from July 1, 2009 to June 30, 2011. The proposed model strategy try to take advantage of the forecasting power of the Deep Neural Networks (*DNNs*) and the classic regression model such as Generalized Linear Model (*GLM*), Seasonal AutoRegressive Integrated Moving Average model (*SARIMA*) and AutoRegressive Integrated Moving Average with eXplanatory variable (*ARIMAX*) method. Three different hybrid strategies are designed in order to merge their results with *DNNs*, namely, Zhang's method, Khashei's method, and moving average filter-based method [69]. The achieved results showed that the hybrid model had a better performance than the individual ones. The tests are performed for four categories of patients depending on the complexity. The Khashei´s method kept an under 10% *MAPE* being the one with the best performance out of the three available methods.

Table 3.3-1 shows a summary of techniques commonly used in time series modeling.

Author		Modeling technique				
	LSTM	MLP	ARIMA	SVM	ANFIS	FL
[65]	✓	✓	✓	✓		
[67]	✓			✓	✓	
[70]		✓	✓	✓		
[71]	✓					
[72]	✓	✓		✓		
[69]	✓		✓			
[68]	✓		✓	✓		
[73]				✓		
[6]		✓	✓		✓	
[74]		✓	✓			
[75]		✓	✓			
[76]		✓				
[66]				✓		
[74]		✓	✓			
[3]		✓				
[10]				✓		
[9]		✓	✓	✓		
[14]		✓				
[2]			✓			
[77]			✓			
[11]						✓
[78]		✓			✓	
[79]		✓	✓			
[80]		✓				
[81]		✓		✓		
[82], [83]		✓				
[84]						✓
[85]		✓				
[86]		✓				
[87]		✓				
[88]		✓				

Table 3.3-1 Summary of techniques commonly used in time series modeling.

3.4.Training process

The training process based on Radial Basis Function (*RBF*) [82] is the most efficient when used in neural network models due to the need for an efficient, adaptive and versatile architecture in computational time as well as the precision of the result. A neural multi-level architecture composed by *RBF-SOM* (Serially Operating Multipliers) algorithms executed in parallel in a programming method known as *CUDA* (Compute Unified Device Architecture) is proposed by [89] to improve the accuracy and timing of the *RBF* model with the same amount of data. The validation for such a structure consisted in estimating ecological variables with information on the environment through the *CUDA-RBF-SOM* structure that showed an improvement in the time and precision in the training process compared to using the *RBF* network in a 0.1154% for the same estimated variable. As a conclusion, they came to a new training process (*CUDA-RBF-SOM*) that reduces execution time by 99.8846% in comparison to a *RBF-SOM* model.

The neural network model proposed by [90] focused on reducing the number of input data needed in the training process. A radial-basis neural network is applied to regression problems; the new structure consists on a multi-stage training process which it matches the orthogonal least squares (*OLS*) with an optimization gradient. The validation of the model is carried out on data and prognosis of stress and force in the knee for the prevention of injuries in different scenarios (task) in which the results and estimated times are obtained with other models such as orthogonal least squares (*OLS*) and feed -forward. The backpropagation network (*FFN*) is compared with its real values. The proposed structure named *Opt-RBN* showed, in some scenarios, favorable results in the training data, although the training process time is a little longer, authors considered it a good time because it is less than one minute.

The Stochastic Gradient Descent (*SGD*) model updates the convolutional neural network (*CNN*) during the training process with a noisy gradient calculated from a random batch, and each batch uniformly updates the network, which leads to loss problems in the batches. A model to solve this problem is proposed by modeling it as a stochastic process and automatically select the largest loss batch to accelerate its training and is called Inconsistent Stochastic Gradient Descent (ISGD) by [91]. The key concept is the inconsistent training that dynamically adjusts the training effort without loss. *ISGD* gradually reduces the average batch loss and uses a dynamic upper control limit to identify a large losing batch on the fly. *ISGD* remains in the identified batch to speed up training with additional gradient updates. The tests for validating the *ISGD* are based on data such as ImageNet, *MNIST* y *CIFAR*-10. The result of those tests had a convergence of the *ISGD* 14.94% faster than *SGD* for ImageNet data base, *ISGD* showed a convergence 23.57% faster than *SGD* in the *CIFAT*-10 test. It also showed a convergence 28.57% faster than *SGD* with the *MNIST* database.

The Back Propagation Neural Network (*BPNN*) model is used in the training process to determine the number of neurons needed in the two hidden layers of a neural network to forecast the magnitude on the Richter scale of the earthquakes in a region of the Philippines Sea [92]. With the use of register data from the earthquakes in the region from 1990 to 2014, several series of the *BPNN* models are built in order to forecast the magnitude of the earthquake and compare it to the actual data. From the data, it is concluded that a number of 10 neurons per hidden layer is the ideal number for the forecasting model. Since, forecasting errors of the *BPNN* model with 10 neurons in each hidden layer are very similar to the models that use more than 10 neurons per layer, which cause a longer time of computation for similar results. The results are compared with the actual values of the magnitudes of earthquakes that have already occurred and authors concluded that the *BPNN* model forecasted a reliable Richter scale earthquake magnitude result.

To solve the test categorization problem, Neural Network model for categorization are used. One of the proposed models is Improved Back-Propagation Neural Network (*IBPNN*) by (Li, C. et al., 2014) that with parallel computational process speed up the neural network training for text categorization. The *BPNN* algorithm uses a Sun Cluster with 34 nodes (processors). The parallel *IBPNN* is integrated with the *SVD* (Singular Value Decomposition) technique where the neural network input is represented as a low dimensional feature vector. The validation is performed by using different data bases where the number of processor are modified from 1 to 32 which produced an improvement in the execution times without diminishing the accuracy of the categorized text. The results showed that the parallel *IBPNN* along with the *SVD* technique achieves a faster, adaptive, and reliable training process in text categorization.

Table 3.4-1 shows a summary of techniques commonly used in training of time series models.

Author	Training techniques				
	RBF	**OLS**	**BPNN**	**FFN**	**SGD**
[93]	✓				
[90]		✓		✓	
[91]					✓
[92]			✓		
[94]			✓		
[95]	✓				
[96]				✓	
[97]				✓	
[98]	✓				
[99]	✓	✓		✓	

Table 3.4-1 Summary of techniques commonly used in training of time series models.

3.5. Performance comparison for the time series modeling

A comparison and model selection study is to be applied in the forecast of the steel fiber reinforce concrete (water absorption, divided tensile strength and compressive strength) based on a neural networks architecture. The results of the *ANN* models [100] (Incremental Back Propagation (*IBP*), Batch Back Propagation (*BBP*), Quick Propagation (*QP*), Levenberg Marquardt (*LM*) and Genetic Algorithm (*GA*)) are compared. To analyze the structure for the optimal and reliable results, the performance of the ANN is statistically measure by the root media square error (RMSE) and the media percentage error (MPE). The results showed that the best average precision in forecasting water absorption and divided tensile strength is obtained by the *GA* when compare with *LM* (0.531741 versus 0.549935) and (1.872817 versus a 2.200841) respectively. Even though the *MPE* of *GA* is greater than that of *LM* (0.238413 versus 0.203847) and (2.012567 versus 1.806573). Regarding compressive strength, IBP presented a better performance than BBP based on the *RMSE* values obtained at the end of the test (5.882287 versus a 6.430472), however, the *MPE* of *IBP* is greater than that of *BBP* (1.916267 versus a 1.1804). When evaluating *RMSE* for flexural strength, authors concluded that *BBP* works better in comparison to QP (0.981202 versus 1.000345) however, the MPE of IBP is slightly greater than that of QP (0.30676 versus 0.299273). From the analysis authors concluded that Genetic Algorithm (GA) is the best model to forecast the properties of the steel fiber reinforce concrete, and they stablished an order in terms of general predictive efficiency of all the model as follows: $GA > IBP > LM > BBP > QP$.

An indirect adaptative control method using a Neural Network (NN) based on a Variable Learning Rate (VLR) combined with the Taylor Development (TD) by [101] is proposed for the control of non-linear dynamic systems. The proposed architecture used two neural networks blocks as identifier and controller to ensure the convergence of control systems. The effectiveness of the proposed algorithm is shown through experimentation and simulation and for its performance comparison authors used the Media Square Error (MSE). The simulation time reached the next results:

- Simple Neural Controller Approach ($SNCA$): MSE equal to 8.67×10^{-3} and time of 33.41 seconds.
- NN-TD: MSE equal to 7.86×10^{-3} and time of 30.10 seconds.
- *NN-TD-VLR: MSE* equal to 5.65×10^{-3} and time of 24.32 seconds.

From the above results authors concluded that the application of Taylor Development along with the Variable Learning Rate to a controller neural network improve the training time giving as a result optimal time simulation and a lower MSE.

A multi-layer feedforward neural network ($FFNN$) model by [102] with 11 different training algorithms is developed for the estimation of the Media Number (MN) balanced Molecular Weight (MW) and polydispersity index (PDI) of the nonlinear multivariable bio polymerization of the policaprolactone (PCL). The compared metric in the performance of the algorithm are the Media Absolute Error (MAE), Root Media Square Error ($RMSE$) and Media Absolute Percentage Error ($MAPE$). In the PLC bio polymerization process for 11 different training algorithms with an architecture of six different types which are: Additive Momentum (AM), Self-Adaptive Learning Rate ($SALR$), Resilient BackPropagation (RBP), Conjugate

Gradient BackPropagation (*CGBP*), Quasi-Newton (*LM*) and Bayesian Regulation Propagation (*BRP*). Once the tests are performed and analyzed, it is demonstrated that the data Quasi-Newton (*LM*) is the more suitable training algorithm for estimating due to its low error index, which are a *MAPE* of 4.512% for *MN*, a 5.31% for *MW* and a 3.21% for *PDI*. A MAE of 20.76 g/mol for *MN*, of 26.55 g/mol for *MW* and of 0.45 g/mol for *PDI*. A *RMSE* of 15.84 g/mol for *MN*, of 33.95 g/mol for *MW* and of 1.485 g/mol for *PDI*.

The estimation of crude oil prices in the world is an important factor for companies and governments. It is very common to find the use of an *ANN* model for this variable estimation. To estimate such variable a Multi-layer perceptron (*MLP*) neural network model it is proposed by [90]. The *MLP* model is trained with the oil price data from 1980 to 2014. The estimations are compared with a Vector autoregressive (*VAR*) model by using the Media Square Error (*MSE*). Once the tests were performed and analyzed, the authors concluded that the *MLP* architecture estimates with a very low *MSE* (0.0897) compared to the *MSE* (0.4816) of the *VAR*, so the *ANN-MLP* forecast tends to be more accurate.

Estimating the price of uranium is an important factor that directly affects the operating cost of nuclear power plants. The implementation of neural networks to estimate this type of process is suitable given the non-stationary characteristics of the time series. The *FFNN* tend to suffer from overfittings (unbalance between memorization and generalization). In the work developed in [103] a comparative analysis of the performance of this technique with respect to Support Vector Machines (SVM), Radial Basic Network Functions (RBN) and Regression Tree Ensembles (RTE) is proposed. Once the tests were carried out, it was concluded that the *FFNN* method achieved better precision in estimating the price of uranium (*CDF* of 0.99 and *AMRE* of 0.0533).

Table 3.4-1 shows a summary of techniques commonly used in performance comparison of time series model.

Author	Performance comparison						
	MSE	*RMSE*	*MPE*	*MAPE*	*MAE*	*AMRE*	*CDF*
[100]		✓	✓				
[101]	✓						
[102]		✓		✓	✓		
[104]	✓						
[103]						✓	✓
[105]	✓						
[106]		✓					
[107]		✓				✓	
[108]				✓			
[109]		✓					

Table 3.5-1 Summary of techniques commonly used in performance comparison of time series models.

3.6. Final Remarks

A time series corresponds to realizations of a stochastic process whose statistical properties can be static or dynamic over time. In cases where the phenomena or processes have variations in their statistical properties is more complex to model over time. Besides, the time series dependence with exogenous variables may mean a support for the description of the phenomena and the minimization of the uncertainty within the autoregressive processes. Different authors propose alternatives that allow debugging the irregularities in the data before the building of model. The procedures commonly used in the state of the art are based on averaging, replacing or adding missing data/outliers. In the case of non-stationary time series, some transformations are used to stabilize the behavior of the data. Finally, some filters are introduced to minimize the presence of *outliers* and achieve a more reliable behavior of the phenomena or processes.

To analyze and identify the variables that could bring information about the phenomena of interest could be helpful in the building of the models, because the complexity can be reduced and increase the accuracy in those cases where the univariate model is not relevant. Therefore, several authors propose different strategies to select the variables for the model. The selection of variables can be based on the correlation with the variables of interest or considering their contribution in increasing the performance of the model. Among the most useful procedures are the total/partial correlation functions and the analysis of the variance error.

Once the variables have been conditioned and selected, it is necessary the selection and evaluation of the modeling technique that better fits to the conditions of the time series. There are different classical statistical techniques and some other based on intelligent systems that are used in the modeling of stochastic processes (econometric series, power consumption, population, transportation). Models based on computational intelligence have been widely used for their adaptability in linear and nonlinear processes. *MLP*, *LSTM*, *ARIMA* and *SVM* techniques are commonly used in modeling.

The time series modeling process is not unique, however it is possible to evaluate the performance of various techniques to select the one that best fits the time series. Since there is no formal theory for the selection of the best model, in the case of techniques based on computational intelligence, it is common to find authors who base their selection on iterative processes in which the training parameters change. The *ARIMA* models is based on the Box-Jenkins methodology, which propose a comparative principle based on an observer criterion.

The review of the state of the art presented above has allowed us to identify the contribution of this thesis. In this work, the development of an adaptive methodology for the modeling of stationary and non-stationary time series is proposed. In this case, the proposal incorporates an automatic system for the selection, filtering, transformation and characterization of the variables related to the time series automatically. The main idea is that from the data of the time series and the information of the variables, the system will be able to optimally assemble the most suitable model. Once the representative variables within the process have been identified, the system evaluates the different modeling alternatives included in the knowledge base to build the hybrid model that best fits the time series. In this thesis, an automatic training process is proposed based on a design of experiments that allows identifying the most significant parameters within the structure of the model. Training iterative process can be costly in time and computational resources due to wide experimentation zone. The *Hyperparameters* are considered in order to minimize the time required for the training process of each of the available models. In addition, a maintenance process (*Auditor*) for the assembled model is included in order to provide stable performance considering the dynamic characteristics of the time series. Finally, a performance metric capable of providing information on the deviation in the shape and magnitude of the forecast of the time series is proposed.

PART II

PROPOSED APPROACH

Chapter 4

Adaptive Methodology Approach for Time Series Modeling

This chapter introduces the adaptive methodology approach for time series modeling proposed in this dissertation. The main definitions, general considerations and the algorithms for data analysis, model assembling and performance supervision in this work are introduced in this chapter.

4.1. Problem Statement

The time series modeling based on computational intelligence techniques is a research line of interest for several economic sectors. Many statistical techniques of conventional and non-conventional modeling have been used (e.g. *ARIMA*, *ANN*, *SVM*, *RTE*, etc.) to carry out time series forecasting [19]. However, the combinations of different modeling techniques does not guarantee that models can not lose their reliability, which generates a detriment in the performance of the forecasting process [110]. Based on the above, in this work, a design and implementation of a methodology capable of evaluating, estimating and fitting the model parameters for forecasting is proposed.

In this thesis, a metric of performance to recommend when a modeling must be re-trained is defined. It is necessary to evaluate the behavior and deviation of the model to anticipate the loss of reliability without affecting the forecast. Likewise, it is established a comparison metric that allows selecting of the best available model within the knowledge base.

Time series models based on the integration of conventional and non-conventional statistical techniques have been approached by different authors in order to forecast variables related to different phenomenons (e.g., energy, transport, education, marketing, among others)[19], [111]–[113]. However, the proposed approaches in the state of the art show that it is common for models to lose their reliability and therefore require retraining [20], [114]–[117]. *SVM*, *ANN*, *RTE* and some combinations with classification techniques such as k-means, self-organizing maps and decision trees are commonly used techniques in time series modeling process. The re-training of the modeling has the following challenges and opportunities:

Challenge	Opportunity
The debugging of the time series data to be used in training.	The time series methodology proposed by [19] is used, in which the steps to characterize and model the stationary and non-stationary time series are explained.
Selection of a suitable experimentation area to fit the model parameters.	Hyperparameters are used in order to optimize the training process of the available models.
The definition of a performance metric that allows defining the conditions that maximize the degree of generalization without falling into overfitting.	The relation correlation/MAPE is used as a performance metric to get information about the reliability of the time series forecasting.
Expert support in the model training process in order to select the best model.	The training process of the selected model is automatized.

Table 4.1-1 Challenges and opportunities related to this methodology.

Some works have been carried out where adaptive strategies for ARIMA and neural network models have been addressed [20], [118]. In this thesis, an auditor module capable of indicating when a model must be retrained has been included.

Time series can be non-stationary hence their statistical characteristics are changing over time. This causes that the parameters of the models loss validity or there is the possibility that other techniques could have a better performance [20], [21]. A comparative performance metric that helps to select the suitable modeling technique considering a knowledge base is proposed. The statistical modeling technique based on intelligent systems (e.g., *RTE, SVM, ANN*) are included.

In this thesis, three scenarios related to power consumption, vehicular traffic and energy market sectors are used.

4.2. Proposed Methodology

The time series modeling process requires the characterization and selection of significant variables for the phenomenon of interest. This process allows identifying and integrating variables that provide information about the time series. Once the variables to be used have been selected within the modeling process, it is necessary to define which methodology or statistical technique fits to the features of the time series. However, commonly the models reduced their performance over time. Fig. 4.2-1 shows an outline of the proposed approach. Each of the proposed stages will be analyzed and organized in the following subsections: subsection 4.2.1. Time series data, subsection 4.2.2. Model assembling, subsection 4.2.3. Training process, subsection, subsection 4.2.4. Auditor, subsection 4.2.5. Selection criterion.

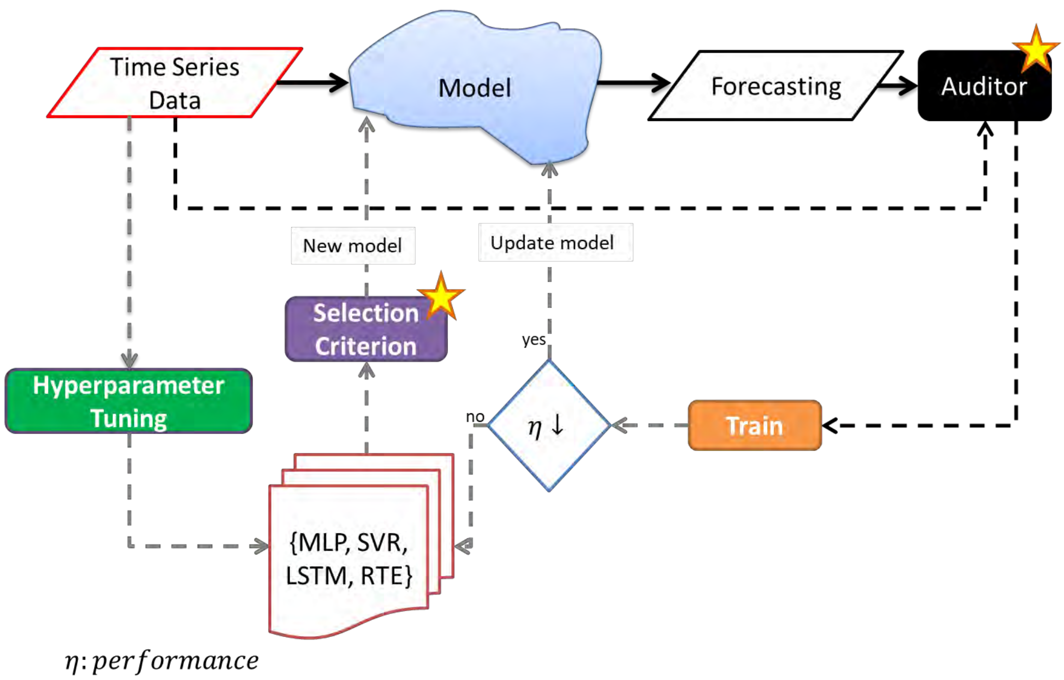


Fig. 4.2-1 Adaptive methodology for time series modeling.

4.2.1.Time series data

Time series modeling requires characterization of the data to identify statistical properties that guide the selection of modeling techniques. One of the most important properties is stationarity[3], since not all statistical modeling techniques are suitable for this type of time series. In case of changes in statistical properties, such as mean and variance, transformation-related adjustments such as differentiation and application of logarithms must be implemented. The constant trend and variance –homoscedasticity– are desired properties in any modeling process because they guarantee stable behavior of the model. The extraction of the characteristics for the time series is necessary for the explanation of the phenomenon. Each of the stages are detailed below:

1) *Interdependence of the time series (previously transformed to stationary time series)*: It includes the evaluation of the historical information required to be able to describe the behavior of the phenomenon to be modeled. A question to be solved is: How many historical information is necessary to forecast future behavior? At this stage, a total and partial autocorrelation analysis is carried out to establish whether the previous data provide information on the future behavior of the time series.

2) *Identification of exogenous variables:* It is necessary to identify which variables minimize uncertainty and help in the description of the phenomenon of interest. In this case, it is important to solve some questions such as: How many historical data of these variables should be considered to answer these questions? The correlation matrix is proposed to identify variables that provide significant information on the behavior of the time series.

3) *Simplification of the considered variables:* the simplicity of a model is the key to evaluating its performance. Many variables do not guarantee a better description of the time series, but it can increase its complexity. Therefore, the

[3] Stationarity in the time series (weak sense) is a desired property since it guarantees that the statistical properties do not change among periods. Thus, the mean and variance are constant regardless of position of the random variable within the stochastic process.

proposed methodology incorporates principal components analysis and factorial analysis for reducing and grouping the variables.

4) *Identification of similarities within the data of the time series:* The number of models depends on the relation between the observed variability and the complexity of the time series data. The data similarity is evaluated through the *Tukey-Kramer* test in order to identify the need for multiple models.

5) *Evidence of periodicity within the time series:* the seasonality of the data provides information about the forecast of the data. The detection of this characteristic within the time series leads to consider historical information related to data with similar conditions.

6) *Patterns identification in time series:* In the same time series, it is possible to detect a diversity of behaviors that can be grouped into small groups. The grouping of time series with similar characteristics (subgroups) allows the calculation of aggregated series that provide information on the most representative patterns. The classification of data is carried out through the decision trees. Once the data are grouped, the calculation of each aggregated series is carried out by adding the curves that belong to the same subgroup.

4.2.2. Model assembling

The proposed methodology includes three (3) stages for obtaining a model that provides a stable forecasting with low variability (see Fig. 4.2-2). The methodology based on stages allows the combination of conventional statistical and computational-intelligence techniques to provide better robustness and enhance model functionality.

The first stage consists of the implementation of an intelligent clustering technique to obtain the shape of a typical load curve for time series data with hourly granulatity (weekdays, weekends, and holidays). The second stage presents the base curve (i.e., the reference curve) for each day that will be forecasted. This base curve is done by implementing statistical techniques that use historical data of time series and the typical curves figured out in the first stage. Finally, the third stage allows correcting the base curves considering external variables (e.g., weather conditions).

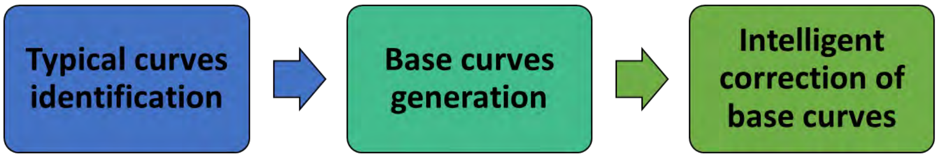


Fig. 4.2-2 The proposed methodology to time series modeling.

1. *Typical curves identification (profiling):* The goal of this stage is to generate an algorithm that considers a set of curves associated with the same day of a week, clusters those curves that represent the typical behavior of that day and rejects the ones that display different or atypical behavior. To identify the typical curves, their normalized shapes are analyzed.

2. *Base curves generation:* stage 2 obtains the base curve for forecasting the time series for each of the considered days. The goal is the generation of base curves as accurate as possible, so that, the possible corrections can be minimized. Thus, the base curve is a preliminary forecasting, which has been obtained using statistical techniques, and it will be corrected at the final stage of the proposed methodology.

3. *Intelligent correction for the base curve:* stage 2 corresponds to a statistical model that allows obtaining preliminary forecasting curves (i.e., base curve), which represent mostly the expected forecast. However, they have to be slightly corrected for performance improvement. The difference between a base and actual curve is due to the effects of external variables (e.g., temperature, humidity, and wind speed). Thus, stage 3 includes computational intelligence techniques for adjusting the forecasting base curve according to the external variables.

   Each of the stages considered within the intelligent correction of the base curve is detailed below.

- *Smoothing of the corrected curves:* In the output of stage 3 is possible to observe a small ripple in the time series forecasting because there exists a certain level of randomness in the models. Therefore, a smoothing technique is implemented for further enhancement of the performance of the proposed model. An inverse clustering technique based on median is implemented to smooth the curves. This technique is a particular case of the centroid method in which the size of all clusters is considered equal [119]. This process considers the four most similar curves to the obtained curve after the intelligent correction. Such curves become a reference to fit the trace of the corrected curve and reduce the ripple to obtain the required smoothing.

- *Extreme value suppressors:* It is possible to obtain extreme values of the time series in each period of the generated curves, since no model is perfect, to correct these atypical values, if they appear. An extreme value suppressor is applied; the suppressor allows better robustness as well as reliability increase. The extreme value suppressor is based on an absolute median deviation elimination using a sliding window.

4.2.3. Trainning Process

In this thesis, two strategies of training and optimization process of each modeling techniques were considered (*LSTM, SVR y RTE*). First, hyperparameter optimization, which it is the process of finding the optimal hyperparameters in machine learning (Bayesian optimization) is integrated with iterative process to train and select the best performance in each kind of intelligent system analyzed [120]. Second, iterative process based on experimental design to identify the most significant parameters within the performance of the neural network (MLP) was developed. Below, a description of these proposals are explained:

a. *Training Optimization –Hyperparameters– (SVR, LSTM and RTE):* The models based on machine learning techniques turned out to be suitable and adjustable due to multiple parameters definition and their interaction with an objective function associated with the precision of the model. For the learning process, the implementation of the backpropagation algorithm is essential to figure out the gradient of an objective function for each parameter of the model. The training process is based on making the right decisions that allow the functions of the multidimensional space to be suitably integrated with an evaluation metric that guarantees optimal learning. Once the functions and the learning algorithm have been established according to experience in the problem, it is important to define a starting point or initial values of the model parameters. The initialization of the parameters before the training process is known as hyperparameters optimization.

b. *Training Optimization –MLP–:* An experimental design methodology that allows providing the highest level of generalization of the forecasting models is proposed. The methodology allows identifying which factors are significant in the training and configuration process of neural network in terms of error (*MAPE*).

4.2.4. Auditor

The modeling of the time series establishes a set of guidelines for the analysis, debugging, and selection of the variables for the training model. However, this set of steps allows the definition of the structure and parameters for the model. The structure of the model depends on the statistical characteristics of the time series data. Commonly, the time series have non-stationary characteristics hence it is normal for the model parameters to lose precision. The design of a system based on computational intelligence techniques is proposed to avoid detriment in the performance of the forecasting models.

4.2.5. Selection criterion

A module to identify which of the available models is appropriate for the characteristics of the time series is defined. This module provides a metric that indicates when the current model can not maintain the desired performance levels. Now, the training of each of the models available in the knowledge base will be evaluated using the performance metric *rMAPE,* which is proposed in this thesis and is defined as:

$$rMAPE = \frac{MAPE}{r_{xy}} \qquad (4.2\text{-}1)$$

Where,

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum {x_i}^2 - (\sum x_i)^2}\sqrt{n\sum {y_i}^2 - (\sum y_i)^2}}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| x100$$

r_{xy}: Sample statistic of Pearson´s correlation coefficient.

$MAPE$: Mean Absolute Percentage Error.

x_i: actual data time series.

y_i: time series forecasting.

n: number of time series data.

This metric is suitable to evaluate the capabilities of forecasting performance considering the shape and deviation of time series. The performance metric that have been proposed here will be define by the interval $0 \leq rMAPE < \infty$. The above is because $0 \leq MAPE < \infty$ and $-1 \leq r_{xy} \leq 1$. In accordance with the previously intervals, it is possible to build the different scenarios through this metric.

Since the ideal is a $MAPE$ closer to zero while the Pearson´s coefficient be closer to 1 (total positive correlation), it is possible to indicate that the lower the value of $rMAPE$ is the better the performance of the model. Next, it will be carried out the comparison for the other scenarios.

5) $rMAPE$ close to zero (0) is the result of a very little MAPE while the correlation (r_{xy}) is close to one (1), that is, the forecasted time series is closer to an actual magnitude and shape.
6) $rMAPE$ with tends to infinite is the result of low correlation (r_{xy}) closer to zero (0) no matter $MAPE$ behavior, that is, the forecasted time series is closer to the actual magnitude but its shape does not correspond with reality. This happens when due to the compensatory effect of error above or lower, a “good” performance is reached but shape of the forecasted time series does not correctly match the actual magnitude.

4.3. General Considerations

Time series models *(TSM)* are ensembled taking into account the suitable modeling techniques in the knowledge base. With the aim to build a suitable model for each one of the available time series *(TS),* significant variables (V_i) and transformations (Tr_i) must be identified until to reach the stationarity.

$$\forall TSM_i \in M \; where \; M = \{TSM_1, TSM_2, \dots, TSM_n\} \quad (4.3\text{-}1)$$

$$i = 1,2,3, \dots, n \; time \; series$$

Where M is the forecast of the selected model for each one of the time series *i.* For each time series, it is necessary to perform a set of data conditioning actions, e.g., the validation of the stationarity test to perform the necessary mathematical transformations Tr. At the same time, the variables to be considered are evaluated within the analysis process and their contribution to the explanation of the phenomena to be modeled.

4.3.1. Adaptive methodology

It is necessary to identify and establish the significant characteristics for each time series (TS_i) and the exogenous variables (VE_i). For the time series TS_i, the Dickey-Fuller test that evaluates the hypothesis about the existence of a unit root DFT, i. e., the stationarity of the time series. If the time series is not stationary, it is necessary to carry out of transformations in mean (Du^k) and variance (Lv^k), allowing to get an equivalent time series TSt_i. The process of selecting the variables requires of a multivariate statistical analysis: 1) correlation analysis (ACF_i), 2) correlation matrix (MC_i), 3) reduction and simplification of the number of variables to be considered by using main components analysis (AF_i),4) identification of subseries or classes with similar characteristics through the *Tukey-Kramer* (TKt_i) multicomparison test, 5) spectral power density for the identification of seasonality in the time series (PSD_i).

It is possible to find *j* patterns in the time series data *i* to simplify the number of models. The different patterns are obtained by using the hierarchical clustering technique with the time series data (TS_i) and the more significant exogenous variables (VE_{ij}). For the representative patterns of the identified subgroups (CTS_{ij}), it is proposed that in addition to the time series (TS_i) data, the concatenation of the data of the significant (VE_{ij}) exogenous variables. The result of this stage is the distribution of the time series data in different groups with similar characteristics and/or shapes. The number of groups j depend on the data distribution of the time series i. In Fig. 4.3-1, the implementation scheme proposed in this thesis is shown.

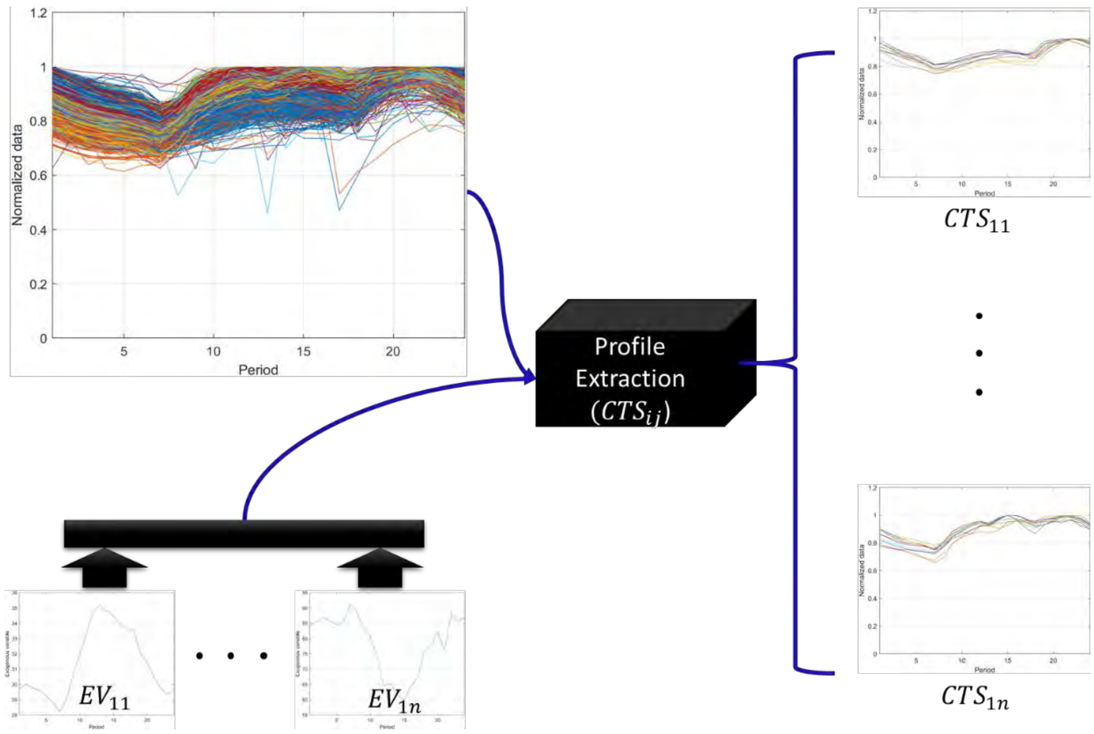


Fig. 4.3-1 Profile extraction module.

It is necessary to determine the level or value not explained by the profiles (CTS_{ij}) for the building of each one of the time series ($M_{ij}{}^{k}$) models. The uncertainty degree given by delta value ($d_{ij}{}^{k}$) for the time series i, the j group and the kind of selected k model. In this stage, the different models are trained taking the input values of each of the representative exogenous variables to explain variations in the behavior of the time series i within each one of the k groups. Fig. 4.3-2 shows the proposed forecasting structure.

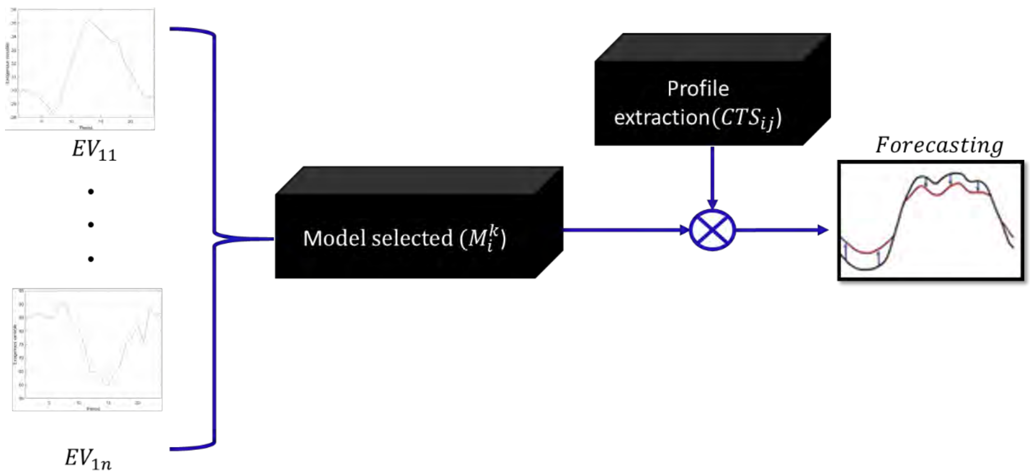


Fig. 4.3-2 Proposed forecasting structure.

The training process of each of the models based on neural networks is carried out by varying each of the significant parameters within the performance of the models. The determination of the parameters of the structure to be varied are obtained through an experimental design, which the following factor are taken into account: 1) network type, 2) number of layers, 3) the number of neurons in the hidden layer, 4) activation function type. Additionally, the factors defined in the training and validation stages are: 5) initial learning coefficient, 6) number of data to consider, 7) the percentage of data for training, validation, and testing, 8) training algorithm, 9) training epochs, 10) training time and 11) presentation data order for training. Akaike´s criterion is used as a metric to evaluate the level of complexity of the evaluated model and its error to determine the performance for each case. Once the relevant factors are determined within the performance of the model, these parameters are evaluated in an iterative way until evaluating the largest possible experimentation area. For each configuration and possible values, the results obtained by using the *rMAPE* here proposed are compared. Fig. 4.3-3 shows the training algorithm scheme for the models based on neural networks.

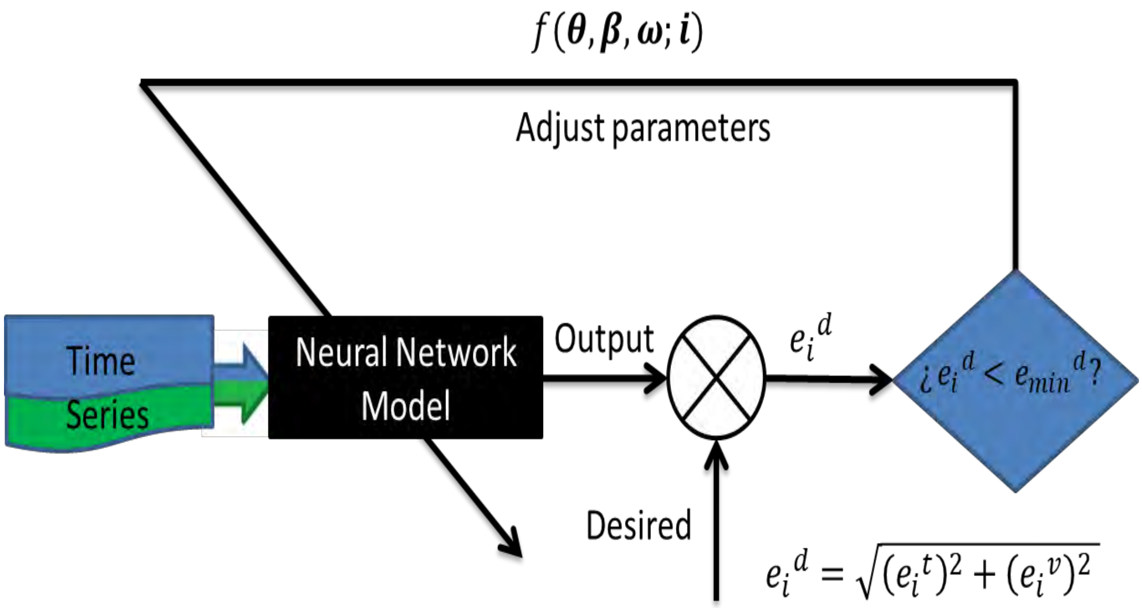


Fig. 4.3-3 Proposed training process.

The training process for each one of the models $(M_{ij}{}^k)$ requires of a total path of the experimentation zone (EZ_i^k), which can represent a high computational cost. Therefore, the definition of hyperparameters is proposed with an a *priori* function that is based on experience to establish a starting point for each parameter $(Mp_i{}^k)$ to optimize the training process. It is possible to reach the desired values by using the Bayesian optimization algorithm. Fig. 4.3-4 shows the estimating process of the data distribution function corresponding to the parameters that best adjust to the desired performance.

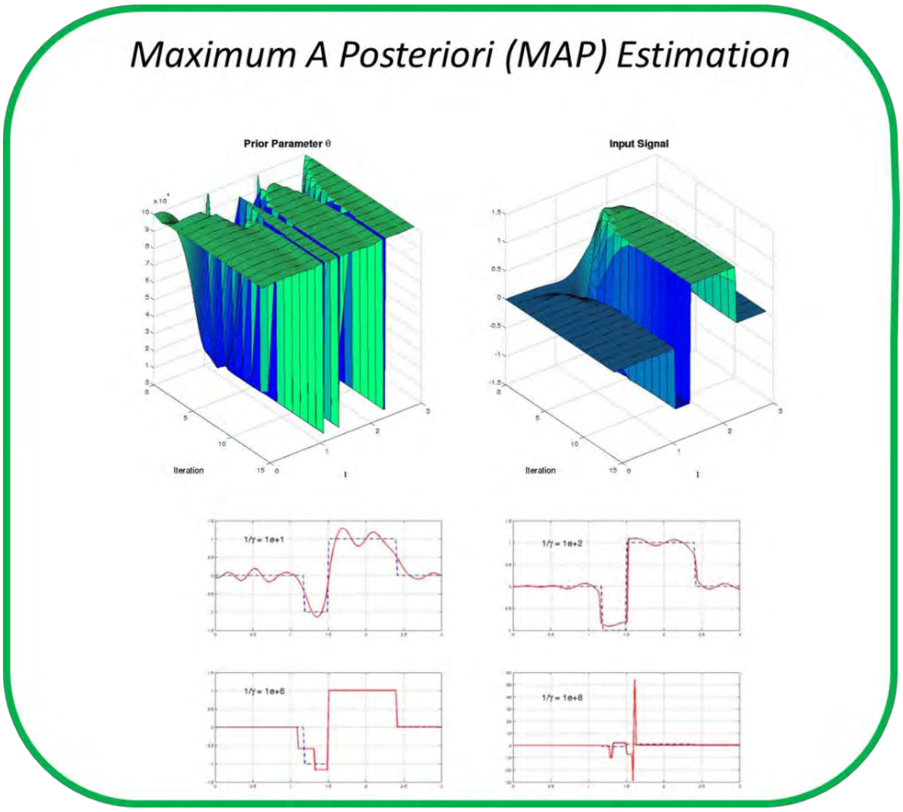


Fig. 4.3-4 Estimation process of data distribution function.

Once each model has been implemented (${M_{ij}}^k$), the auditor (A_i) evaluates in a continous way the performance to indicate when a retraining or updating of the implemented modeling technique are required. The auditor (A_i) is designed by using a *random forest* classification technique, considering: 1) deviation of error of the forecasting models (${M_{ij}}^k$) during the training and inference stages ($\Delta e = {e_{train(i)}}^k - {e_{test(i)}}^k$), 2) variation in the behavior of the probability distribution function on each one of the data during the training and implementation stages ($\Delta pdf = {pdf_{train(i)}}^k - {pdf_{test(i)}}^k$), 3) deviation of the training error trend from the forecasting ($\Delta te = {te_{train(i)}}^k - {te_{test(i)}}^k$), 4) number of periods that can be found outside the confidence interval of the training error $\begin{cases} 1, IC_{inf} \leq {M_{ij}}^k \leq IC_{sup} \\ 0, otherwise \end{cases}$. The confidence interval (IC) is determined while considering the limits by the 95% of the obtained error during the training stage, $[\mu_{error} - 1.96\sigma_{error}, \ \mu_{error} + 1.96\sigma_{error}]$. According to the considered input, the model is selected when it is necessary to update or modify the current forecasting.

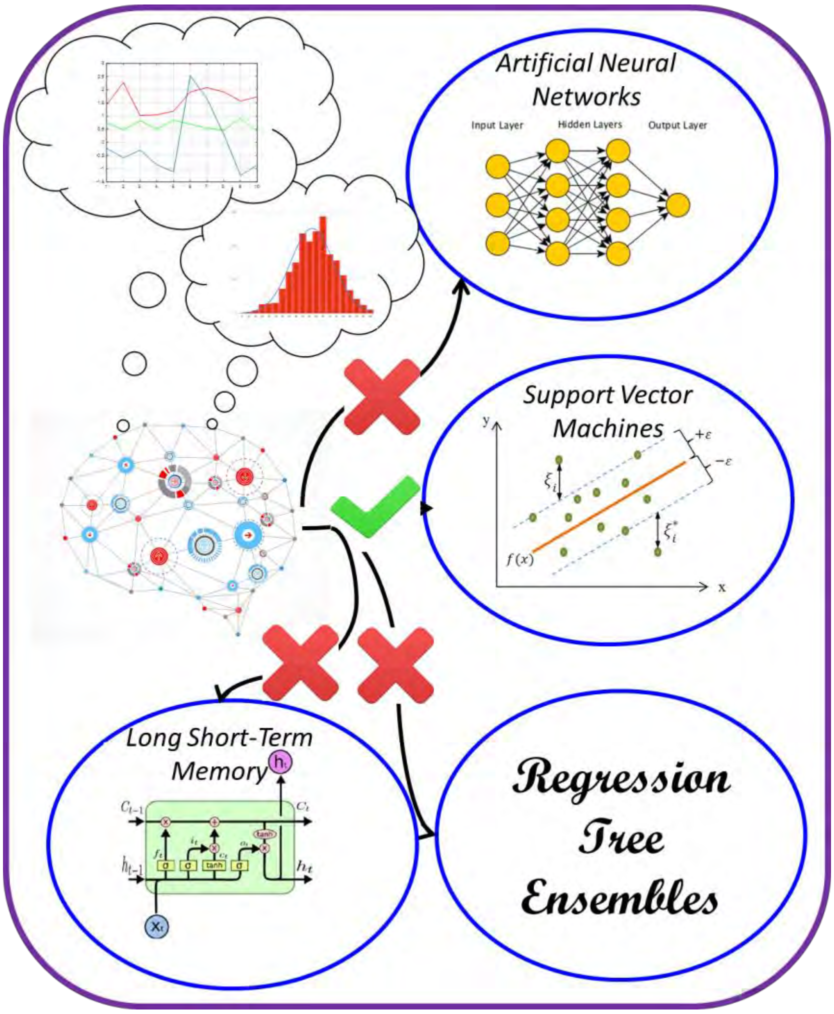


Fig. 4.3-5 Selection schema of available models.

Once the *Auditor* (A_i) provides the updated recommendation or replaces the model ($M_{ij}{}^{k}$) for each subgroups *j* that belongs to the time series *i,* the training of each one of the available k models is carried out. Once the training for every model has been done, the comparison of the obtained performance is carried out by using the *rMAPE* metric here proposed. Fig. 4.3-5 shows the selection process scheme.

The proposed methodology can help in the modeling process of each kind of time series. This proposal guarantees a high and constant performance in an automatic way with the selection of each model according to the time series features. A novel strategy based on time series classification to building a low quantity of models depending on each group assembled in the stage of typical curves identification is proposed. In general, this methodology allows a support to time series models thanks to the *Auditor* that it takes over of maintenance of each model. With *hyperparameters* in the training process, it is possible a reduction of time required for this task. Finally, the new metric is helpful when the *Auditor* must be selecting a new model with a better generalization performance of the time series.

Chapter 5

Implementation

This chapter introduces the application of the proposed approach considering three kind of time series. It describes some considerations about the data and the simulation software used to develop the experiments. A new proposal in a time series modeling to forecast the behavior of several phenomena as power consumption, vehicular traffic and electrical market is presented. Likewise, intelligent and non-intelligent techniques proposed in the literature and in this proposal for time series modeling are analyzed.

5.1. Developed Computational Tool

It is necessary the development of a computational tool in order to test the adaptive methodology that has been proposed. The tool allows the handling of the time series TS_i while considering the different exogenous variables VE_{ij} to carry out the analysis, debugging, selection and assembling of the more suitable models M_i. Weather information, taxonomic variable of the day, calendar type and the same data of the time series are requested prior to the building model. The data of the time series are hourly.

5.1.1. Interface

The proposed tool has a user interface and the building process of the model is automatic with minimal human intervention. Fig. 5.1-1 shows the main interface of the tools. To build each model, the user must upload the corresponding data to the time series by using the “Load” button. Once the data is uploaded, the assembled model is added to the available time series list. For each time series, a set of models are built for every kind of day and the different weather seasons of the year. The user could validate the assembled models and the data related to the development, weather correlated variables, and lags that are considered for the autoregressive process and the transformation for those non-stationary time series.

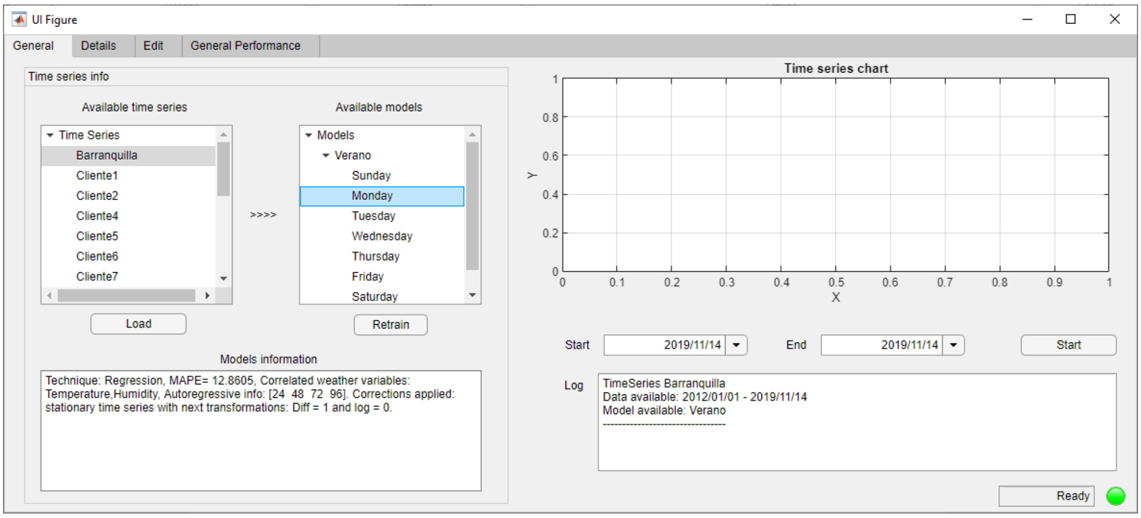


Fig. 5.1-1 Main interface to time series modeling.

Once the models are implemented, the *Auditor* carries out the continuous monitoring of the model performance; in case that retraining is detected, the "retrain" button will turn green. Through the button, the user could carry out the retraining tasks for each models.

The user must indicate the desire start date ("Start") and the end date ("End") to forecast the data for the selected time series. Once the forecast is indicated, the "Start" button must be pressed to get the forecasted data for each available models.

The historical and forecast data curves will be display to compare the precision of the used models in the "Time series chart" section. In the text panel, "log" actions carried out by the user during the execution of the tool are registered.

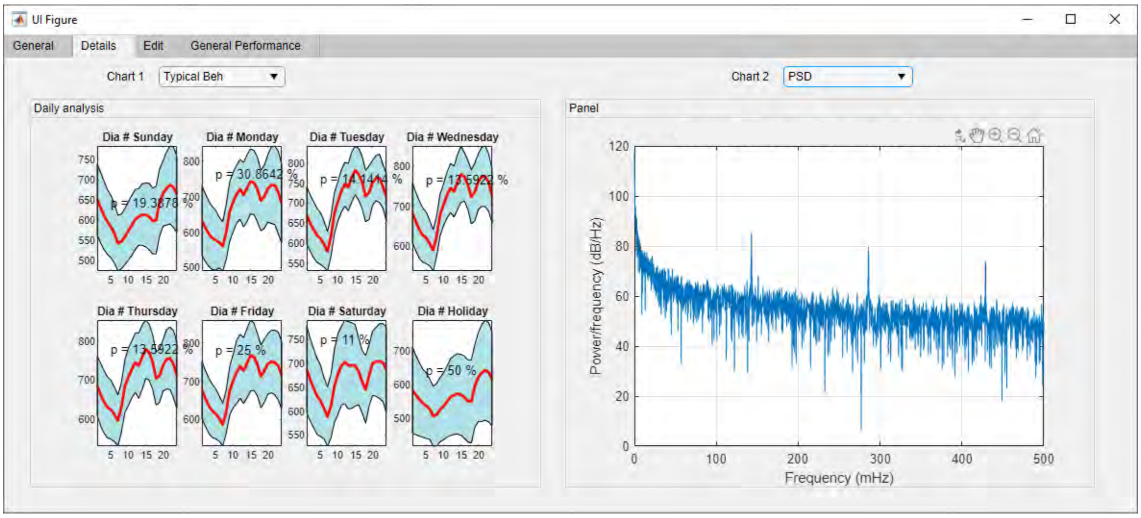


Fig. 5.1-2 Details module to data analysis.

Since the proposed methodology does not require of a human intervention, a detail panel is incorporated to compare and analyze some characteristic of the time series by using the graphic statistical techniques (see Fig. 5.1-2).

Chart # 1: Profiles, typical behavior, partial autocorrelation function and correlation matrix.

Chart # 2: Multicomparison through *Tukey-Kramer* test and power spectral density.

This set of graphical analysis techniques gives the user the tools to understand the grouping of the different models, the detection of the stationary patterns, the identification of the amount of prior data (Lag) to generate the autoregressive process and the selection of the exogenous variables to explain the behavior of the time series.

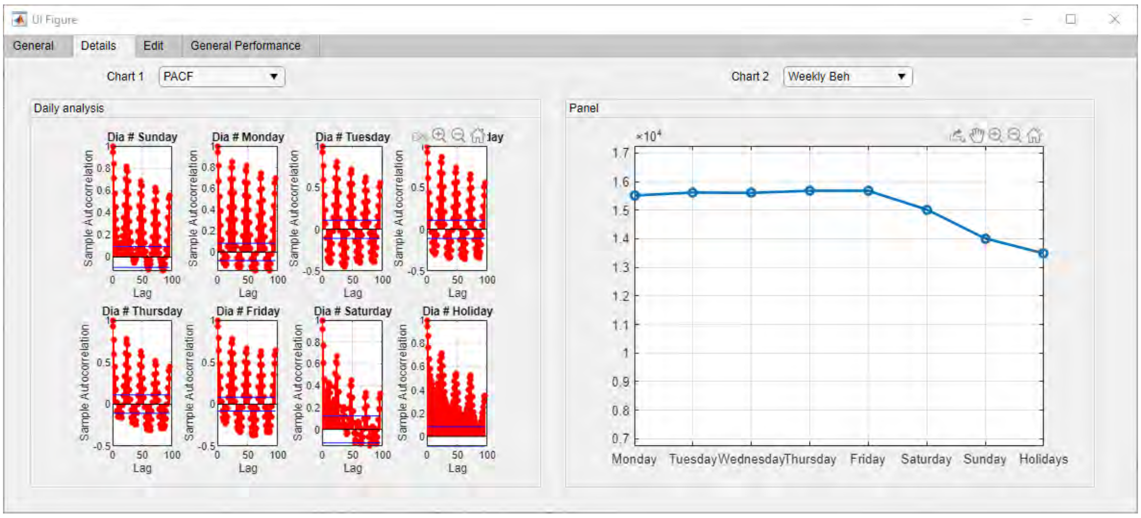


Fig. 5.1-3 Debug module to typical behavior identification.

It is important for the user to establish, which curves do not represent a typical behavior as a preliminary step to retraining and characterization of the time series data. Above, it can help to avoid the addition of values that do not correspond with reality or adding noise in the modeling process. Fig. 5.1-3 shows the "Debug" panel in which the user could filter manually the atypical curves that are found within the data base.

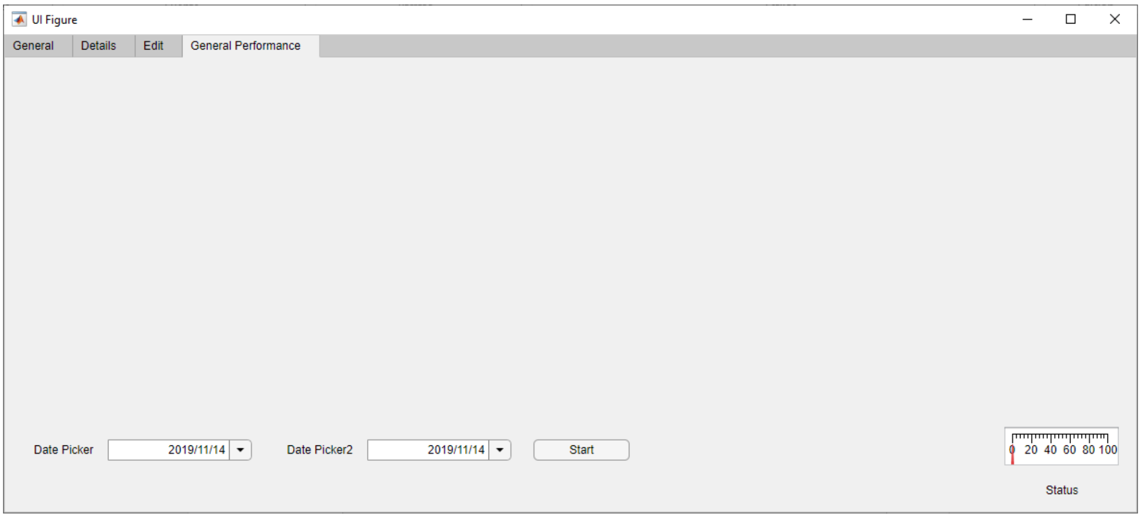


Fig. 5.1-4 General performance module.

Fig. 5.1-4 shows the "General Performance" panel that is used to carry out the test of the available model for the selected time series.

5.1.2. Model implementation

A description will be given about the implementation of the identification stages for atypical curves, generation of base curves and intelligent correction of the base curves.

1. *Typical curves identification (profiling):* The goal of this stage is to build an algorithm that considers a set of curves associated with the same day of a week. To identify the typical curves, their normalized shapes are analyzed. For the implementation of this stage, the following actions are required:
    - Normalizing the time series by dividing all the data per curve by the maximum individual value. In this way, the curves range between 0 and 1, and it is possible to focus on the curve shape.
    - Obtaining the clustering by employing the hierarchical clustering technique. This method divides the data into sets considering how similar they are.
    - Measuring the similarity using *DTW (Dynamic Time Warping)*. This technique is considered the best metric for the comparison of time series. Due to the temporal characteristic in the curves of the time series, they are part of the set of problems in which this technique presents excellent results.

Fig. 5.1.5 shows the obtained results for this stage. The typical curve is observed with a thick solid line.

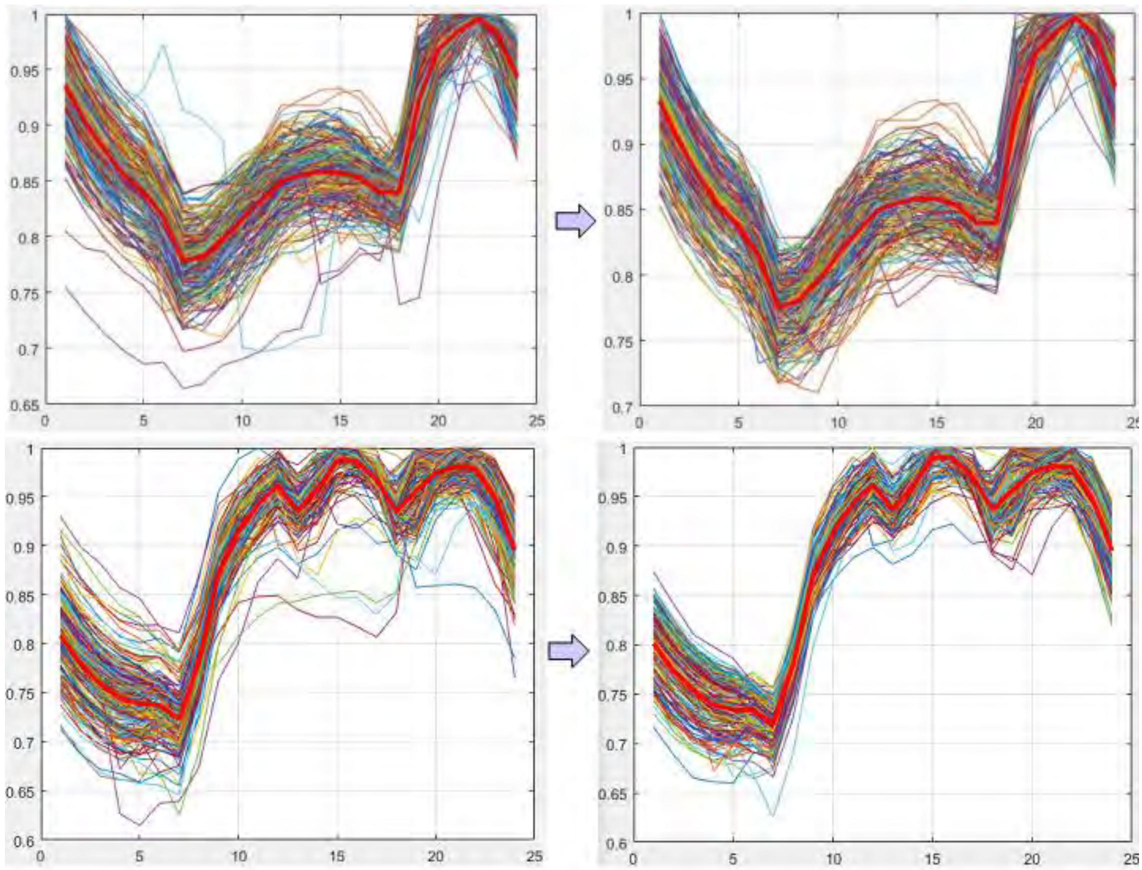


Fig. 5.1-5 Identification of typical curves for the sunday (top row) and monday (bottom row) for a time serie.

2. *Base curves generation:* The process to obtain the base curve is described as follows:
    - Once the day to be forecasted is identified, the typical curve obtained in stage 1 is used to start the process. Such a curve is normalized but it represents the typical shape of the time series analyzed.
    - The typical curve of the day is located at the current average level of time series. This is achieved by verifying the average level of the time series data from the previous days for weekdays, and from the average level of the time series of the same day in the last week for saturdays, sundays, and holidays.
    - The obtained curve is adjusted using the last four curves for the same day, obtaining in this way the base curve.

Fig. 5.1.6. shows the building of a base curve for the time series.

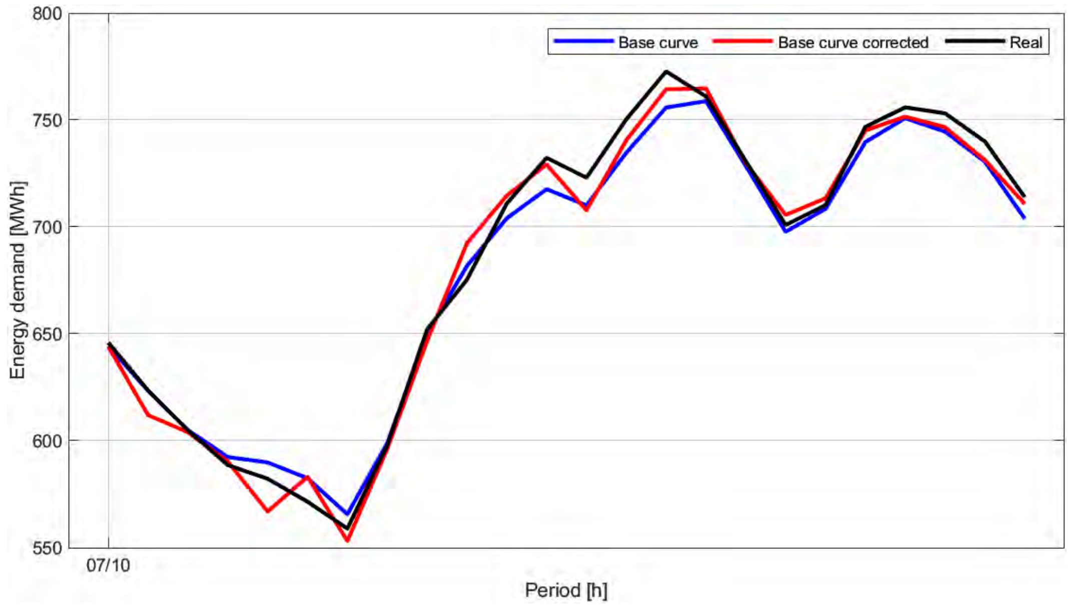


Fig. 5.1-6 Sample for a base curve of time series data.

3. *Intelligent correction for the base curve:* The required steps for this stage are as follows:
   - The computational intelligence technique (*ANN*, *SVM*, *RTE* or *LSTM*) is used to adjust the base curve. Each technique has to adjust only one type of day (i.e., Sundays to Saturdays, taking the holidays as independent days).
   - The input for the computational intelligence technique is a function of the external variables, while the output is the adjustment that must be applied to each period (increase or reduction) to obtain a better forecasting.
   - The computational intelligence technique is trained with the historical register.

Each of the phases considered within the intelligent correction of the base curve is detailed below.

7) *Smoothing of the corrected curves:* In the output of stage 3 is possible to observe a small ripple in the forecasting curves because there exists a certain level of randomness in computational intelligence-based models. Therefore, a smoothing technique is implemented for further enhancement of the performance of the proposed model. An inverse clustering technique based on median is implemented to smooth the curves. This process considers the most similar curves to the obtained curve after the intelligent correction. Such curves

become a reference to adjust the trace of the corrected curve and reduce the ripple to obtain the required smoothing (see Fig. 5.1.7).

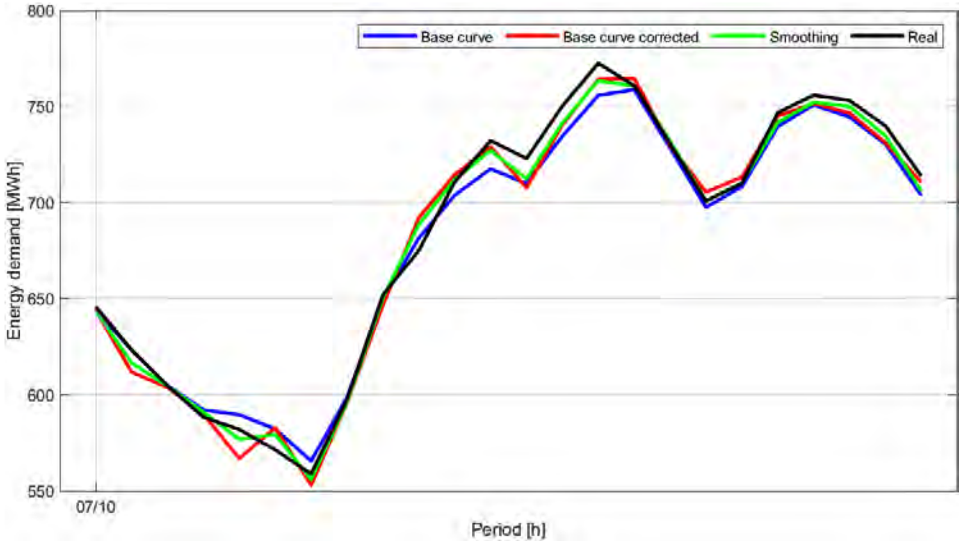


Fig. 5.1-7 Sample smooth of the corrected base curve obtained previously for the Sunday in time series data.

8) *Extreme value suppressors:* It is possible to obtain extreme values of the time series in each period of the generated curves. Since any model is not perfect, an extreme value suppressor is applied to correct these atypical values, if they appear. The suppressor allows better robustness as well as reliability increase. The extreme value suppressor is based on an absolute median deviation elimination using a sliding window (see Fig. 5.1.8).

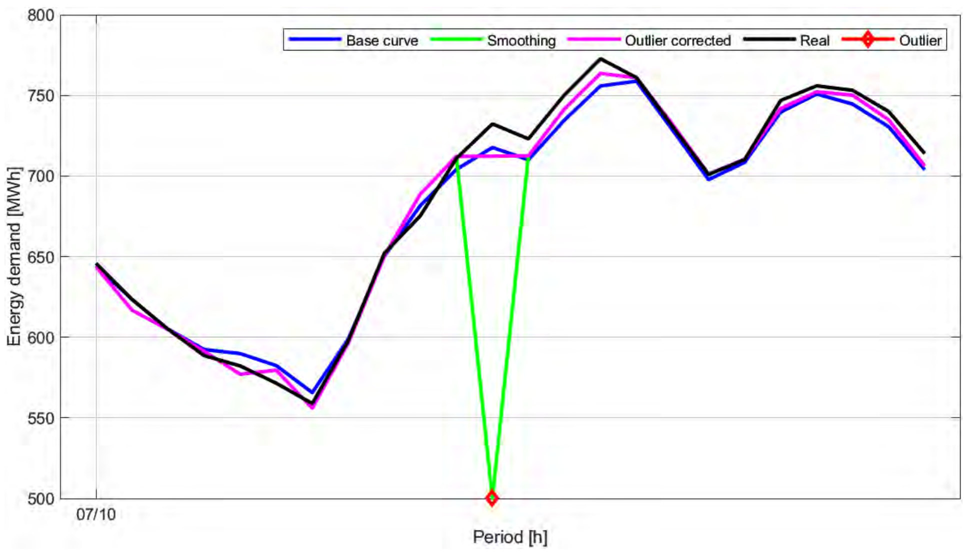


Fig. 5.1-8 Implementation of the extreme value suppressor for an ordinary day of the time series data.

5.1.3. Trainning Process

a. MLP trainning

To train this kind of feedforward neural network, an experimental design is setted to evaluate and select the model with the best performance. Those factors that influence adaptability and generalization in the process of training, validation and selection of neural networks are identified. The factors that modify the configuration of the neural network are: 1) types of network, 2) number of hidden layers, 3) number of neurons in the hidden layers, 4) type of activation function. Besides, it is possible to define factors in the training and validation process stages: 5) learning rate, 6) number of data to be considered, 7) the percentage of data for training, validation and testing, 8) training algorithm, 9) training epochs, 10) training time, 11) presentation data order for training.

The factors 1, 4, 9, 10 and 11 must be kept constants during the iterations. The activation functions of the neural networks to be considered for the modeling process for the time series are the sigmoid in the hidden layer and the lineal in the outer layer [121]. A feedforward multilayer perceptron neural network (*MLP*) is selected due to its characteristics of generalization.

The factors 2, 3, 6, 7 and 8 must be considered as designing factors to be manipulated to verify their significance within the *MAPE* error level. Since the aim is to select those significant factors in the design of a neural network for the analyzed time series, the definition of two levels (high and low) is chosen.

Table. 5.1-1 shows the factors to be considered during the training and validation processes.

Factors	**Levels**	
	Low	**High**
Training algorithm	Levenberg-Marquardt (-1)	Resilient Backpropagation (1)
Number of hidden layers	2	6
Number of neuron in the hidden layer	1	50
Number of data	60	120
Validation percentage	10	30

Table 5.1-1 Design factors of the experiments performed.

Since the objective of this study is to identify those significant factors in the performance of a neural network for forecasting purposes, the response variables such as the index Akaike information Criterion (*AIC*) and the error performance level (*MAPE*) are chosen.

b. SVM, LSTM and RTE trainning

To train *SVM, LSTM* and *RTE* models, it is proposed an optimization process based on Bayesian theory. The Bayesian optimization theory is used to carry out the global optimization for multimodal functions, such as models based on computational intelligence systems. A Bayesian hypermodel represents the data behavior and the qualitative information through the distribution of probabilities. Table. 5.1-2 shows the factors to be considered during the training and validation processes.

ML Model	Hyperparameter	Value
LSTM	HiddenLayerSize	[1 50]
	Learning rate	[1e-3 1]
SVR	C	linspace(0.01, 5, 20)[4]
	gamma	range(0.01, 0.5, 0.05)
	kernel	{linear, poly, rbf}
RTE	max_depth	range(8,15)

Table 5.1-2 Initial ML models and their hyperparameters settings. All models were trained using statistics and machine learning toolbox of MATLAB.

Next, the Bayesian optimization process algorithm is explained:

1. It is necessary to include a *prior* that is a believe about a phenomenon before any evidence is presented. When it has not any knowledge about a parameter, it is treated as a random variable (e.g., Gaussian multivariable function).
2. Combining the *prior* distribution with some observations given by (tested configuration) to get *posterior* distribution (*MAP*, i.e., Maximun A Posteriori) on the target (the estimation where it has the actual function).

3. Using the *posterior* distribution to evaluate the next configuration, according to an acquisition function, it is posible testing the next configuration with the greatest amount of information.
4. Evaluating the selected configuration in the third step.

Steps 2-4 must be iterated until the values converge [120] once all the steps have been completed. Fig. 5.1-11 shows an example of the execution of the prior algorithm.

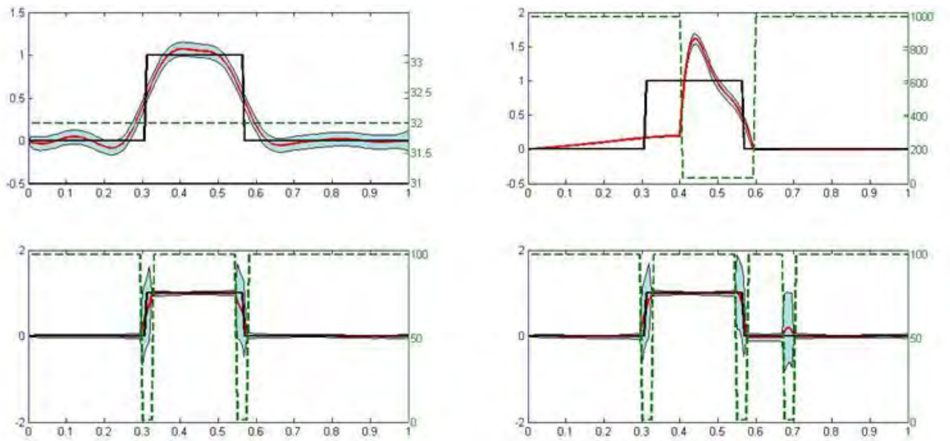


Fig. 5.1-9 *MAP* solution with different variance behaviors. *MAP* Estimation (red line) and its deviation (blue fill). Original signal (black line).

5.1.4. Auditor

There are several reasons that can lead to loss of precision of a model. Next, a list of facts that could motivate to an update or modify the model is presented:

1) Errors obtained during the training or validation process. Here, the building of the typical profiling is used in the training stage through the *clustering* process. It is normal that the non-stationary time series varies its statistical properties over time, this is due to an increase or a change in the consumption patterns, modifications in mobility policies, marketing campaigns, etc. The profiles may vary with respect to those obtained during the training process causing the model to lose validity. It is necessary to use a metric that will characterize this kind of cases. The dynamic time warping algorithm to measure the degree of similarity between the time series is used.

2) Variations in the probability distribution functions as a result of the training process. This variable works as an indicator of the data frequency and its typical behavior in normal conditions. A variation of this metric between the training and the implementation stages also provides information of the time series dynamics.
3) A deviation from the training error trend with respect to the forecasting. The growth rate of error is compared.
4) Numbers of periods outside of the confidence interval of the training error. This provides information about the model behavior during the implementation stage.

Once the variables to be considered are characterized by the *Auditor*, expert knowledge is used to build a supervised model by using *random forest.* Depending on the behavior of the variable the *Auditor* will be train with a logic variable that will indicate if the model should be re-trained. A graphical description of this process is shown in Fig. 5.1-12.

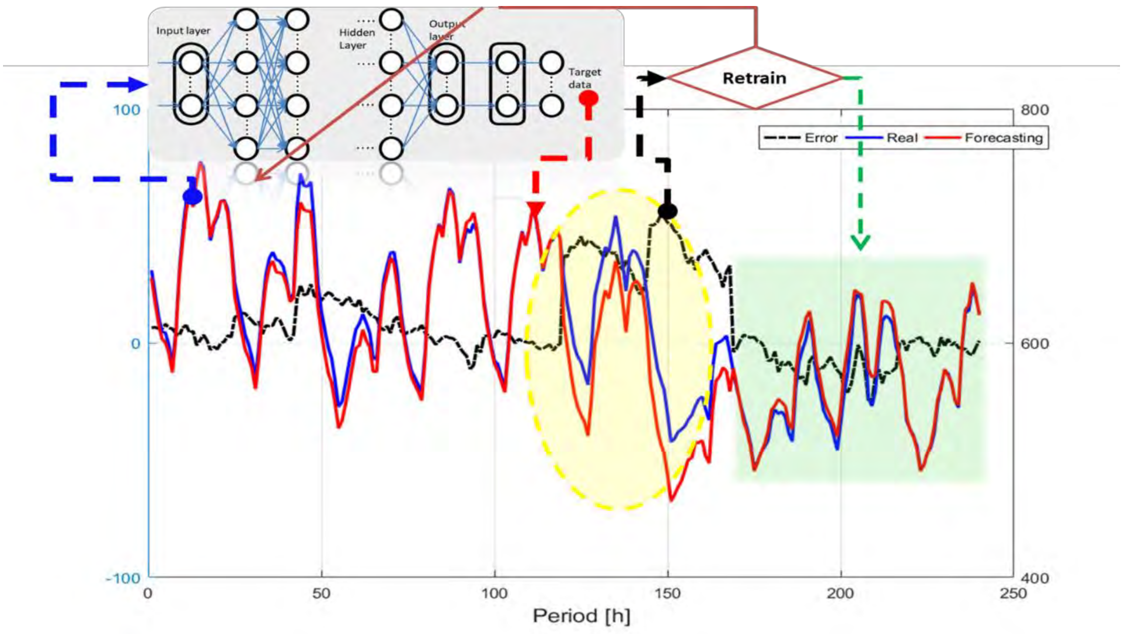


Fig. 5.1-10 Auditor structure.

5.1.5. Selection criterion

It is necessary to train and evaluate the performance of each models by using the metric *rMAPE* proposed here to carry out the implementation of this stage. For each case, a model with the lowest value will be selected to characterize the data from each subgroup.

PART III

EXPERIMENTS RESULTS AND CONCLUSIONS

Chapter 6

Experimental Results

This chapter introduces the discussion and analysis of the empirical experiments and testing that have been carried out for the proposed test beds. The results depicted in this chapter demonstrate the utility, feasibility and reliability of the overall proposed approach presented in the previous chapters.

6.1. Data Description

In this section, the comparison of the proposed methodology with other methods about three different time series (power consumption, vehicular traffic and energy market) is carried out to demostrate superiority in the time series forecasting. The contribution of each stage in the performance of the forecasting will be shown. Table 6.1-1 show a summary of datasets used in this section to evaluate the performance of this thesis.

No.	Datasets	Number of instances	Number of attributes	Area
1	Electrical demand in a building.	2542	107	Energy
2	Energy demand in electrical market # 1.	730	107	Energy
3	Energy demand in electrical market # 2.	730	107	Energy
4	Vehicular traffic	440	107	Transport
5	Electricity hourly-ahead market price.	365	107	Energy/Economic

Table 6.1-1 Data sets chosen to evaluate *AM*.

6.1.1.Energy demand

Data of several time series related to energy demand were used in order to asses the performance of this proposal. Follow some details are explained:

Case # 1. Energy demand in a building: Hourly data related to energy demand in a building are used in training, validation and test process for a total data equal to 2542 days x 24 periods. Zeros and outliers´ data are cleaned because these add noise in the assembling model process. Dataset has non-stationarity properties and weather seasons that modify the times series behavior in time. Data are provided by E-LAND project (see Fig. 6.1-1) [122].

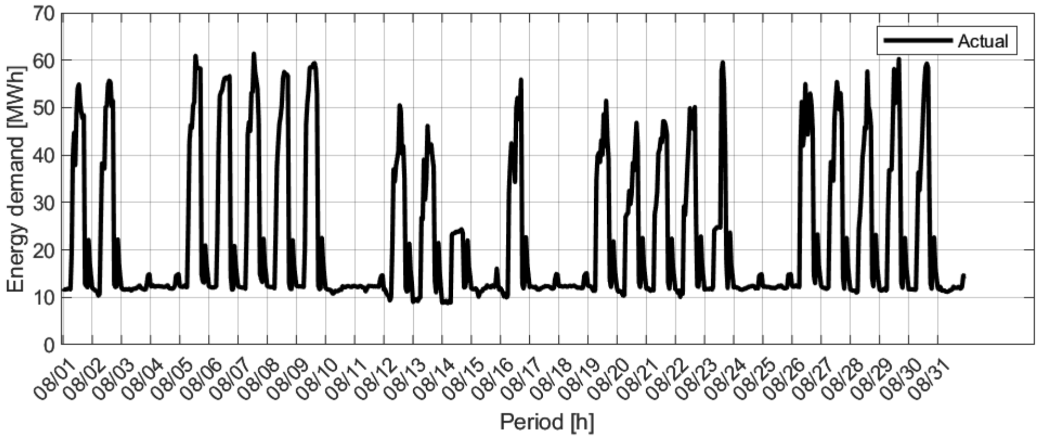


Fig. 6.1-1 Energy demand of a building in E-LAND Project.

Case # 2. Energy demand in electrical markets: Energy demand data are normalized during the building process of the forecasting models for each commercialization market (see Fig. 6.1-2 and Fig. 6.1-3). To analyze the energy demand, historical energy demand data of two (2) energy commercialization markets in Colombia (Market 1: Atlántico and Market 2: Antioquia). Dataset has non-stationarity properties that modify the times series behavior in time. In addition, the time series has changing characteristics in shape and trend due to energy consumption behavior. Weather data are consulted through the website www.accuweather.com.

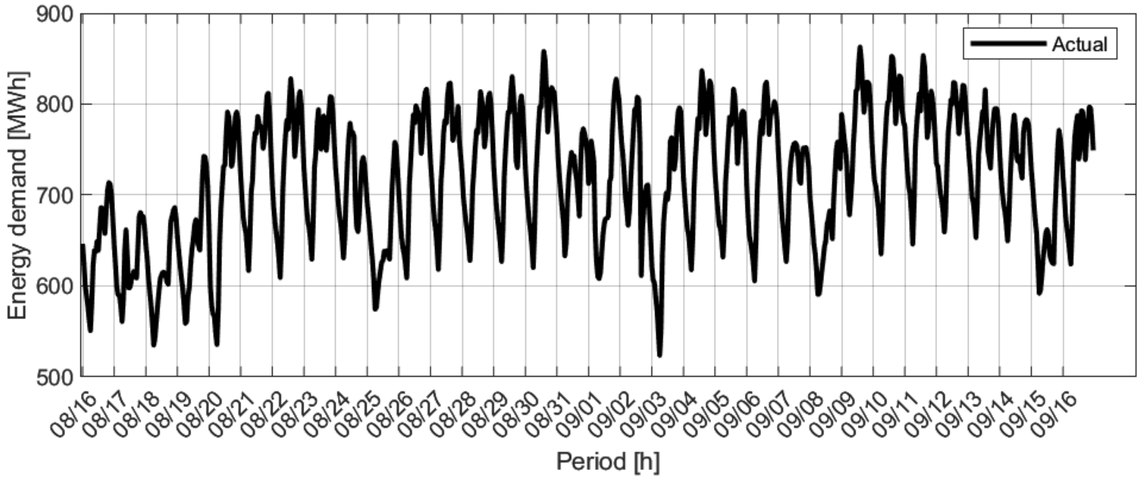


Fig. 6.1-2 Energy demand of electrical market # 1.

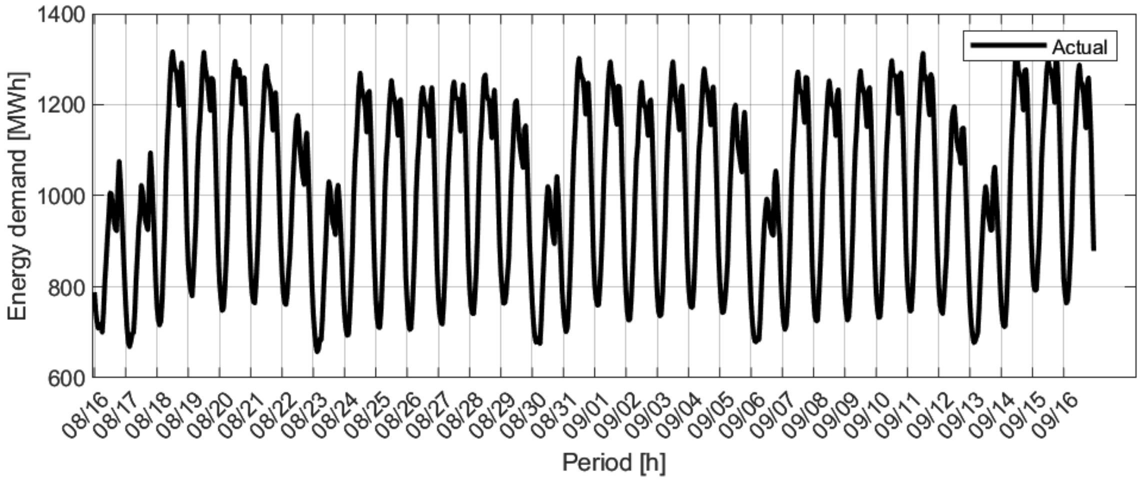


Fig. 6.1-3 Energy demand of electrical market # 2.

6.1.2. Vehicular traffic

Road traffic data will be used to evaluate the performance of the group phase for the construction of the base curve in each one of the required models for the forecasting of the time series data [123]. Data describes the vehicular occupancy, between 0 and 1, from different highways of the San Francisco bay area. The database has 440 data with hourly granularity (440 days x 24 periods) for each time series. This dataset has stationarity features, i.e., the time series is stable in statistical terms. In addition, there is a label for each observed day of the time series, from Monday to Sunday; with numerical labels that ranging from 1 to 7, respectively.

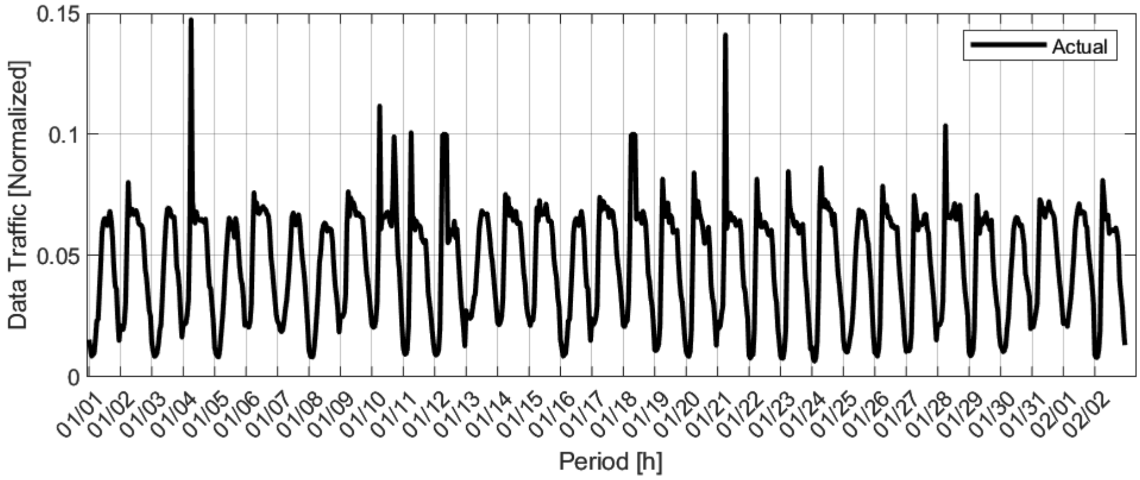


Fig. 6.1-4 Vehicular traffic testbed.

6.1.3. Energy market

The Hourly Energy Price (*HOEP*) is charged to local distribution companies (*LDCs*), in the IESO-administered market, other non-dispatchable loads and paid to self-scheduling generators. Businesses that use more than 250,000 kWh a year pay the hourly price. The *HOEP* is also the basis for regulated rates charged to residential and small business customers. The *HOEP* values are reported as $/MWh.

The models are trained on hourly data of an electricity market. Data detailed above are provided by [124].

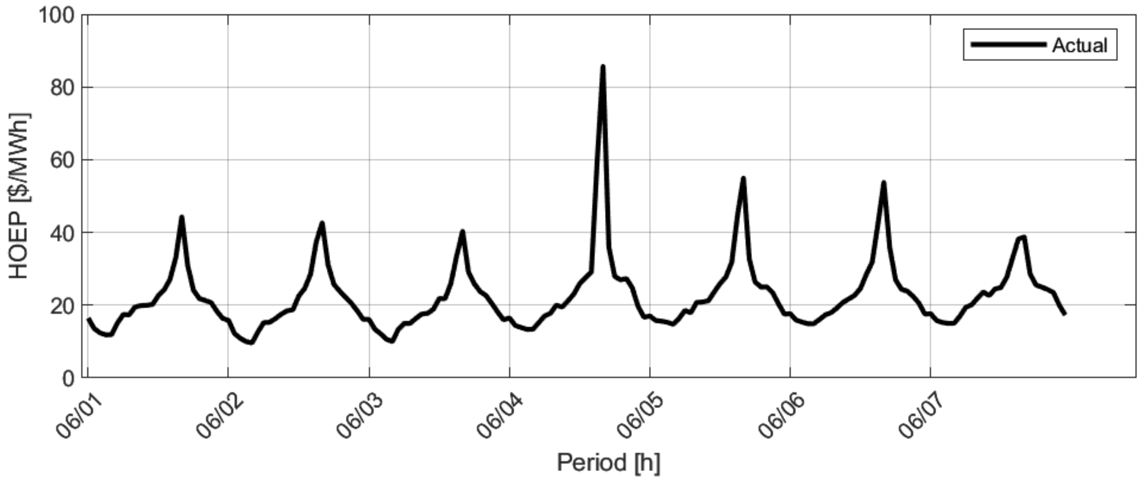


Fig. 6.1-5 Hourly energy price.

Data is normalized in order to avoid differences in data magnitudes that can affect the consistency of data distribution. The normalization process is carried out through the Equation (6.1-1) for each variable to consider for the time series forecasting. Data normalized is set in 0 to 1.

$$X_{normalized} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (6.1\text{-}1)$$

Where,

$X_{normalized}$: normalized time series.

X: original time series.

$\max(X)$: maximum value of time series.

$\min(X)$: minimum value of time series.

6.2. Experimental Design

The following experimental design is proposed to evaluate the performance of the proposed methodology. The comparison of the performance for the forecasting methodology is carried out by evaluating the *rMAPE* metric and the distribution of the selected models. At the same time, it will be evaluated if there is a difference in the performance obtained by each one of the available forecasting models. Data sets from a variety of domains have been conducted to demonstrate the generalizability of the designed scheme.

For each time series, 70% of the data is taken for training, 15% for validation and 15% for testing. The performance of the models is validated by using one of the rolling windows with a step k that depends on the numbers of models for each subset. For the validation process, the separation of data in small subsets has been proposed to avoid the overlap of data during the training procedure. This ensures that the models are unaware of all the validation data. Once training process for the model is done, the tests data are used to evaluate the performance of the model. The training data selection will take place randomly and following a uniform distribution.

Taking into account the above, it defines the following characteristics to the experiment design:

Response variable:

1) Time series data.

Independent variable

- Time series models:
  1. Artificial neural networks (*MLP*).
  2. Suppor vector regression (SVR).
  3. Long-Short term memory (*LSTM*).
  4. Regression Tree Ensembles (*RTE*).

- Time series data. Three (3) types of time series (electric demand, vehicular traffic and electrical market).

There are some charts bellow that show the results obtained by each response variable according to the design of experiments. The results describe the performance of each one of the available models and the suitability of each technique to be considered within the proposal.

Two kinds of graphical comparison are implemented to carry out the statistical analysis of the proposed methodology: 1) accumulated average and 2) *Tukey-Kramer* multicomparison means test. The analysis allows the identification of the error trend in a variable time window. Besides, the inclusion of the multicomparison test in the analysis brings information about the performance of each models in this thesis.

6.2.1.Experimental metrics

The *rMAPE* and *MAPE* metrics will be used in order to carry out the performance evaluation of each forecasting models in this work. The performance metrics *MAPE* and *rMAPE* were defined in sections (2.8-4) and (4.2-1), respectively.

6.3. Experimental results

The behavior will be evaluated in three different scenarios in order to evaluate the performance. Before, a comparative analysis of training process for MLP case will be reviewed in order to evaluate the contribution of this approach in the modeling performance.

6.3.1. Training of MLP

The 2^k factorial design is selected to evaluate each factor effect in the neural network's training and validation processes. This experiment is suitable when the goal is to analyze the significance of the factors with a minimum number of runs [125]. In this case, it is only considered two levels (low and high). This can be viewed as a weakness when the factors have significant interaction and a curvature in the experimental zone. When the curvature is detected, it is necessary to aggregate central and axial points in the experimental design. Another aspect to consider is the null quantity of degree of freedom for the error. Thus, it is recommended to apply the following two strategies: 1) to identify the significant effects through a normal probability plot, i.e., remove from the analysis those factors fitted to the normal probability plot because their behaviors are similar to the residues. In this case, the degree of freedom of each excluded effect is added to the error. 2) to aggregate central points to the 2^k experiment; these points are added in the center of the experimental zone, at points $x_i = 0\ (i = 1, 2, 3, \dots, k)$ [125]. This strategy allows adding a degree of freedom to the error and measuring the response variable experimental zone curvature. The experimental design carried out is a 2^5 with six central points. Three of them will be developed with a qualitative factor in low level (Levenberg-Marquardt algorithm) and the other three in high level (Resilient Backpropagation algorithm). If the linear model does not fit the data, adding an axial point locates at the central point will be necessary.

It is necessary to develop a script that allows to modify the architecture of the neural network keeping the factors according to what is shown in Table 5.1-1 to get the realizations of the experiment. Each sequence is executed to figure out the *MAPE*. A multifactorial ANalysis Of VAriance (*ANOVA*) is used to identify the factors that are significative to the analysis and adjustment process of the forecasting model. Fig. 6.3-1 shows the normality of effects.

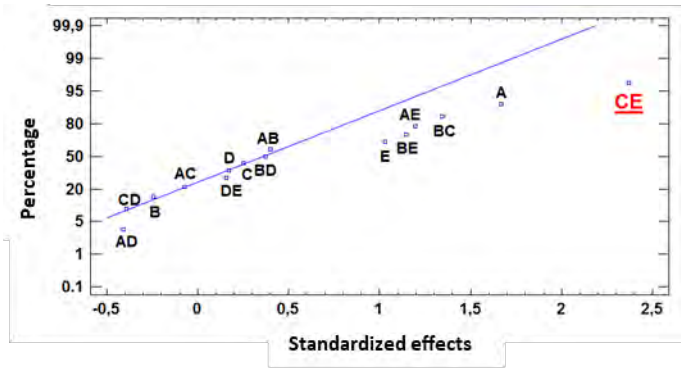


Fig. 6.3-1 Normality of effects graphic.

Fig. 6.3-1 and Table 6.3-1, the CE effect is the most relevant (the interaction among the number of neurons in the hidden layers and the percentage of data validation) and the error value (*MAPE*) is only affected by *CE* effect, since the *p-value* is less than 0.05. It is necessary to verify that the model fulfills the normality, homoscedasticity, and independent conditions to validate the result reached through the *ANOVA* analysis.

Source	Sum of squares	df	Mean square	F	p-Value
A:Train_Alg	0.871079	1	0.871079	2.78	0.1096
B:Hidd_Num	0.0182764	1	0.182764	0.06	0.8114
C:Neu_Num	0.0201734	1	0.0201734	0.06	0.8020
D:Data_Num	0.00900434	1	0.00900434	0.03	0.8669
E:Val_Per	0.331972	1	0.331972	1.06	0.3145
AB	0.0507545	1	0.0507545	0.16	0.6912
AC	0.00168203	1	0.00168203	0.01	0.9422
AD	0.0519588	1	0.0519588	0.17	0.6877
AE	0.446753	1	0.446753	1.43	0.2451
BC	0.567389	1	0.567389	1.81	0.1921
BD	0.0440187	1	0.0440187	0.14	0.7114
BE	0.412097	1	0.412097	1.32	0.2637
CD	0.048383	1	0.048383	0.15	0.6981
CE	1.75907	1	1.75907	5.62	0.0270
DE	0.00784171	1	0.00784171	0.03	0.8757
Total error	6.89167	22	0.313258		
Total (corr.)	11.5321	37			

Table 6.3-1 ANOVA of the experimental results.

Chi square is used to validate the normality condition. The result is shown in Table 6.3-2:

Test	p-Value
Chi-square	0.0269

Table 6.3-2 Normality test for AIC.

Data results for this stage are not fulfilled with the normality test. Since the factors have similar effects compared to the *AIC* response, it is considered to continue the optimization stage only with the last response variable.

According to Table 6.3-3 results, four factors fulfill with the homoscedasticity condition because the *p-values* are greater than 0.05.

Factors	Levene´s test	p-Value
B	0.327	0.722
C	0.305	0.738
D	0.366	0.685
E	0.876	0.425

Table 6.3-3 Homoscedasticity test.

Durbin-Watson statistic = 2.13134 (*p-value* = 0.6651), and Lag 1 residual autocorrelation = -0.158786.

Due to *Durbin-Watson* statistic has a *p-value* greater than 0.05, the null hypothesis is rejected; hence, the residuals have not linear correlation.

- *Regression Model*

According to the *ANOVA* analysis, the interaction of two factors (C: Neurons number and E: Validation percentage) are related significantly to the *AIC* behavior. Now, it is necessary to set an ideal configuration through an optimization process. To carry out this stage, the experimental design to be used is two factors with one replicate.

The results of the experiment are shown in Table 6.3-4.

Num_Neu	Val_Per	AIC
1	10	18.2656065
5,5	20	2.35135344
10	20	165.831393
5,5	10	53.4919072
1	20	7.9016126
10	10	200.675933
1	10	19.8895334

5,5	20	105.424653
10	20	175.661213
5,5	10	121.928836
1	20	29.8195033
10	10	159.69137

Table 6.3-4 Levels of model analysis.

	Coef	Typical error	to	p-value
Number of neurons	16.46	1.44	11.38	1.9E-07

Table 6.3-5 Coefficients test.

Equation (6.3-1) shows the fitted equation of the regression model for the *AIC* response. Coefficient test is taken according to the result shown in Table 6.3-5.

$$y_1 = 16.46C \tag{6.3-1}$$

Regression statistics	
R_pearson	0.96010775
R^2	0.92180688
Adjusted R^2	0.83089779
Typical error	33.1356306
Observations	12

Table 6.3-6 Grouped regression statistics of the model analysis.

Table 6.3-6 shows the grouped regression statistics of the adjusted model. The reached R^2 is a correlation coefficient. Furthermore, Table 6.3-7 shows that the p-value ($4.7736e-07$) is less than α used in the experiment, hence null hypothesis is rejected demonstrating that model fits to the actual data.

SoV	dof	SS	MS	Fo	p-value
Regression	1	142381.8	142381.8	129.677347	4.7736E-07
Residues	11	12077.6	1097.97		
Total	12	154459.5			

Table 6.3-7 ANOVA for model analysis.

Fig. 6.3-2 shows the residuals for the only relevant factor (number of neurons) discovered in the fitting curve for the *AIC* response. It has not a clear structure in the residuals data.

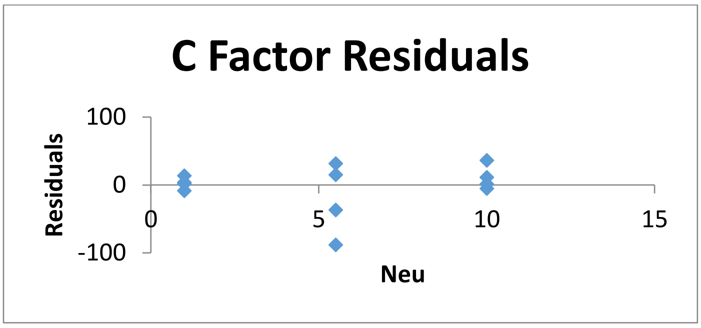


Fig. 6.3-2 Residuals of C factor.

- *Optimization Results*

According to the regression model analysis, only the *C* factor is required for fitting the data to the *AIC* response. It is proposed to evaluate the means to observe whether there is any statistical difference between levels. The results are shown in Table 6.3-8. In this case, the proposal is minimizing the *AIC* response with a minimum number of neurons.

Source	**Sum of squares**	**df**	**Mean square**	**F-Ratio-F**	**p-value**
Between groups	50734.7	2	25367.4	22.75	0.0003
Within groups	10036.5	9	1115.17		
Total (Corr.)	60771.3	11			

Table 6.3-8 ANOVA table for C factor (Number of neurons).

Neu_Number	**Cases**	**Mean**	**Homogeneous group**
1	4	18.9691	X
5.5	4	70.7992	X
10	4	175.309	X

Table 6.3-9 Fisher *LSD* test.

Table 6.3-8 and Table 6.3-9 show the results of *LSD* analysis. Results show that there is no statistical difference between 1 and 5.5 levels. However, it is notable the change when the number of neurons jump to 10 neurons.

The optimal point where the system reached a minimum *AIC* with a low number of neurons is less than five for one neuron in the hidden layer.

A comparison of performance of each *MLP* model considering this training methodology and other based on iterative process is presented in Table 6.3-9.

Data set	MAPE [%]	
	Iterative	Experimental design
Electrical demand in a building.	24.2	***22.5***
Energy demand in electrical market # 1.	3.5	***3.22***
Energy demand in electrical market # 2.	3.2	***2.8***
Vehicular traffic	26.5	***25.4***
Electricity hourly-ahead market price.	14.2	***12.3***

Table 6.3-10 Performance comparison of each MLP model.

6.3.2. Energy demand

Case # 1: A testing analysis is performed to time series data with MAPE equal to 14.49%. Fig. 6.3-3 shows a time series forecasting.

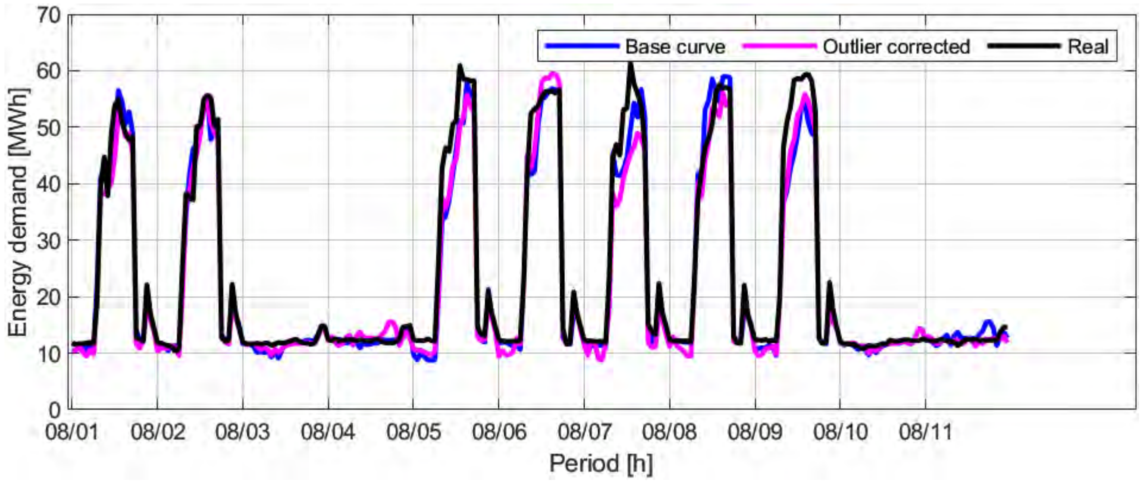


Fig. 6.3-3 Base curve, intelligent correction and actual data of time series of building.

Case # 2: Forecasting data published in the official XM web site [126], are taken as a reference to carry out a comparison of the result obtained by the proposal.

Fig. 6.3-4 shows the forecasting of the energy demand commercialization in the electrical market # 1.

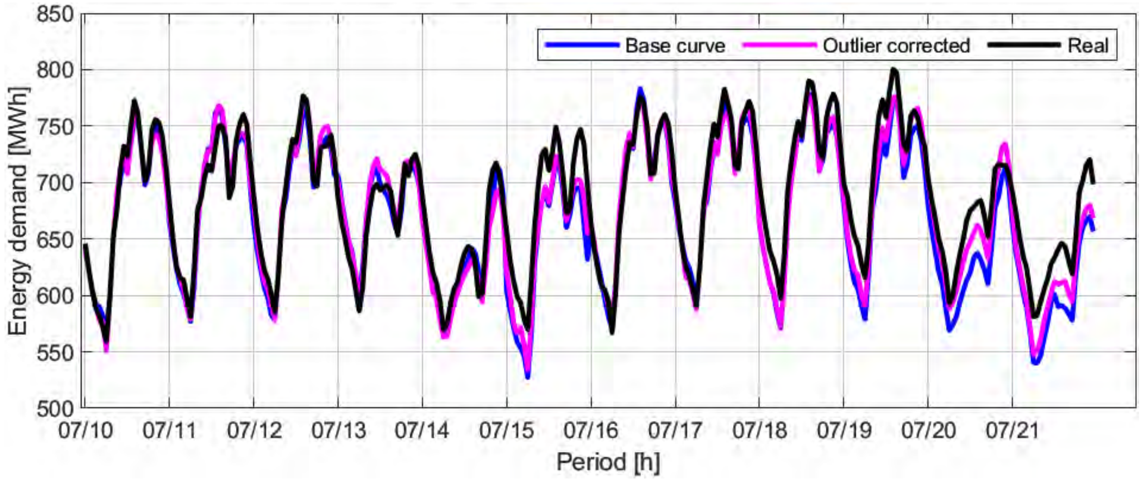


Fig. 6.3-4 Base curve, intelligent correction and actual data of time series of electrical market # 1.

Fig. 6.3-5 shows the forecasting of the energy demand commercialization in electrical market # 2.

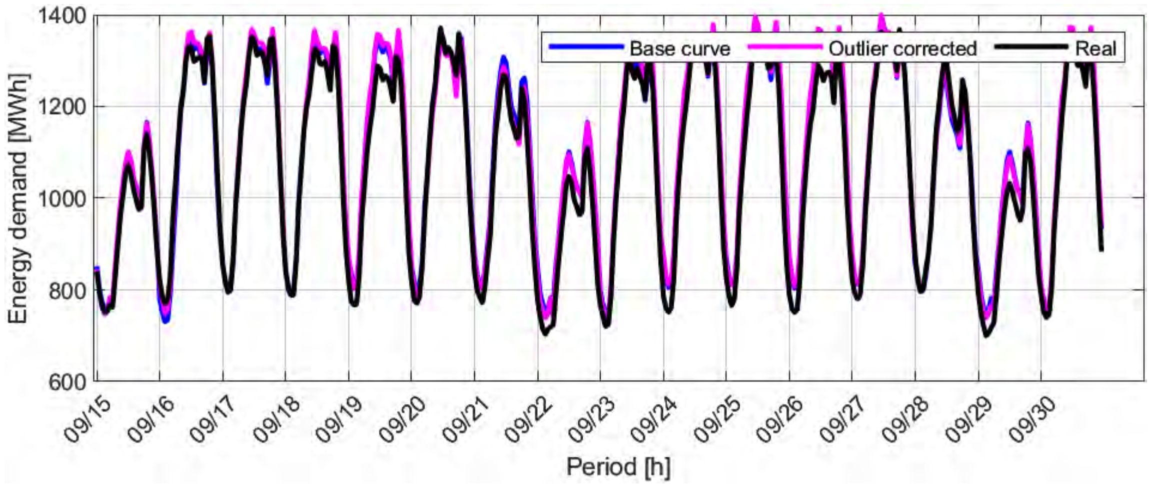


Fig. 6.3-5 Base curve, intelligent correction and actual data of time series of electrical market # 2.

Figs. 6.3-6 – 6.3-7 show the performance (*MAPE*) of the adaptive methodology (*AM*) for the forecasting process of commercialization markets 1 and 2.

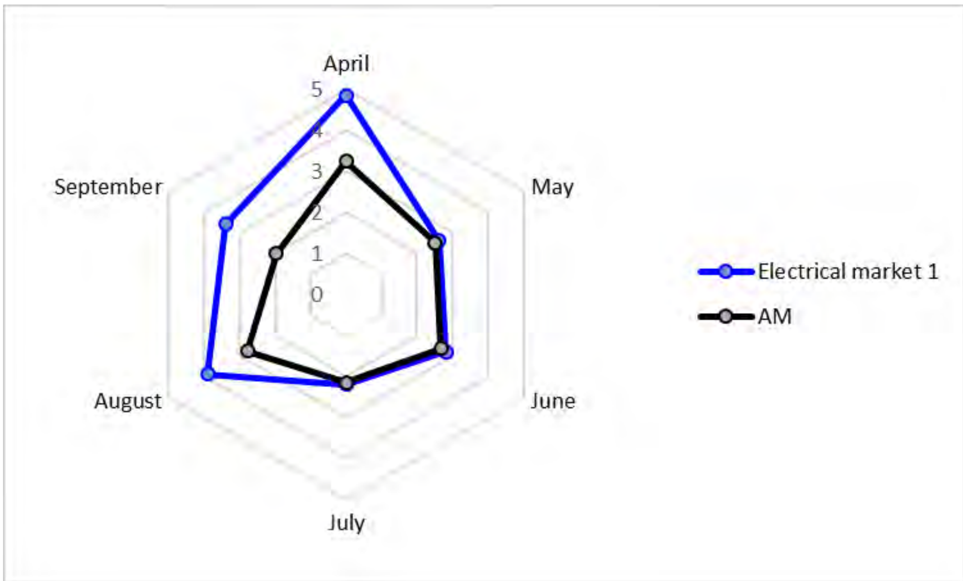


Fig. 6.3-6 comparison of the performance of the model used by the electrical market 1 operator and the *AM* Model.

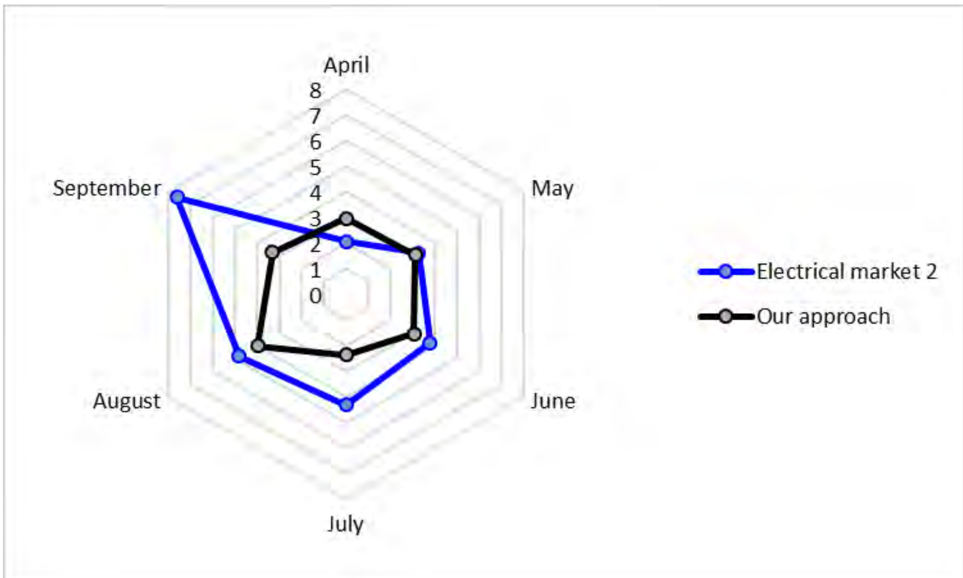


Fig. 6.3-7 Comparison of the performance of the model used by the electrical market 2 and the *AM* Model.

Figs. 6.3-8 shows the individual performance of each one of the modeling techniques and *AM* model used for the time series of the electrical market 2 for a forecasting window (7 days forward).

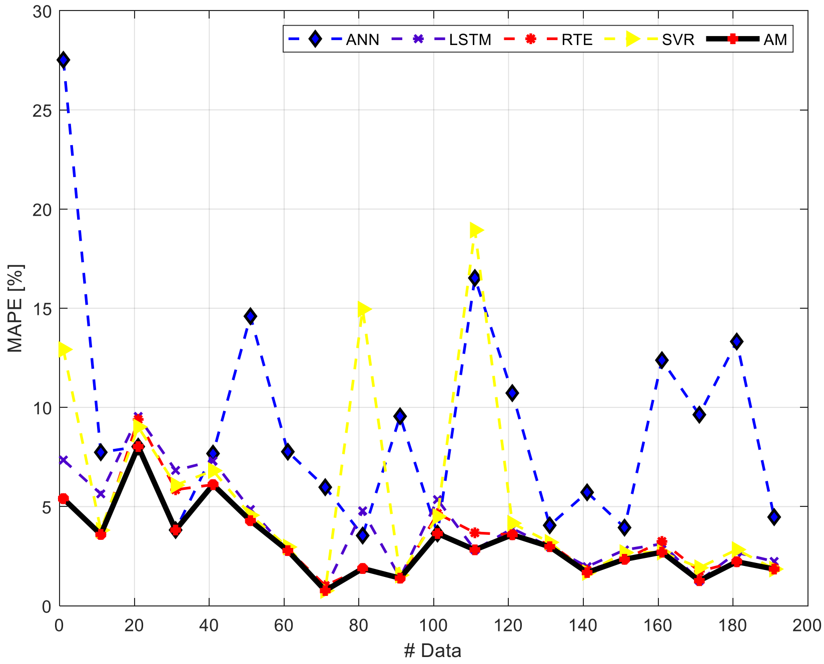


Fig. 6.3-8 Cumulative average *MAPE* for seven days ahead forecasting.

6.3.3. Vehicular traffic

The data used for this experiment are described in section 6.1.2. It is taked as a reference the article [127] which carries out the comparison of different Kernels for the classification of the different data sets including the one analyzed in this section. In order to obtain the results showed in Fig. 6.3-9 where 173 daily data arranged in [127] are selected to carry out the tests. The cumulated error rate is used to evaluate the evolution of the performance of the classification stage while the random presentation of the data progressed. Fig. 6.3-10 shows the forecasting obtained through the proposed methodology. The performance (*MAPE*) achieved is 17.35%. The results of the forecasting are used as reference since the author [127] used the data for the classifying process of the types of days (Monday-Sunday) depending on the hourly traffic detected.

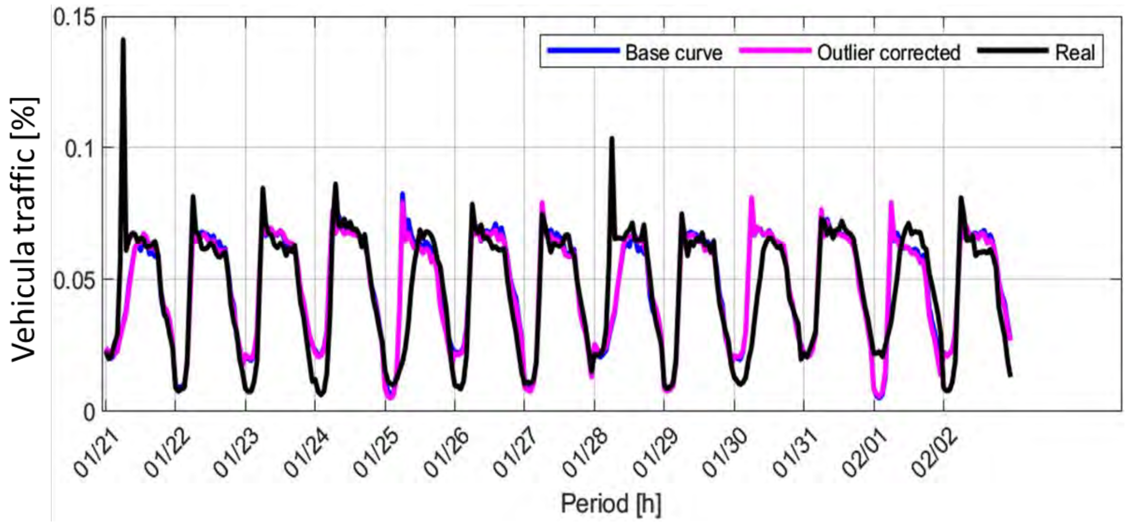


Fig. 6.3-9 Base curve intelligent correction and actual data of the traffic time series of vehicular traffic.

Fig. 6.3-10 show the average classification error rate in tests data proposed by [127].

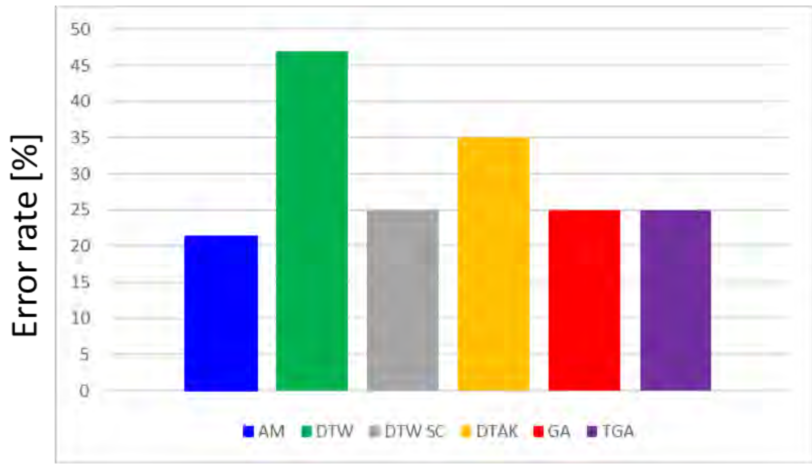


Fig. 6.3-10 Mean and standard deviations of classification error rates for each type of day depending on the traffic data.

6.3.4. Energy market

The data to be used for this experiment are described in section 6.1.3. It is taked as a reference the article [128] where a new artificial neural network (*ANN*) has been used to compute the forecasted price in the electricity market . In order to obtained the results shown in Fig. 6.3-11, six months are selected to carry out the tests. The performance (*MAPE*) achieved by the test period is 14.2%.

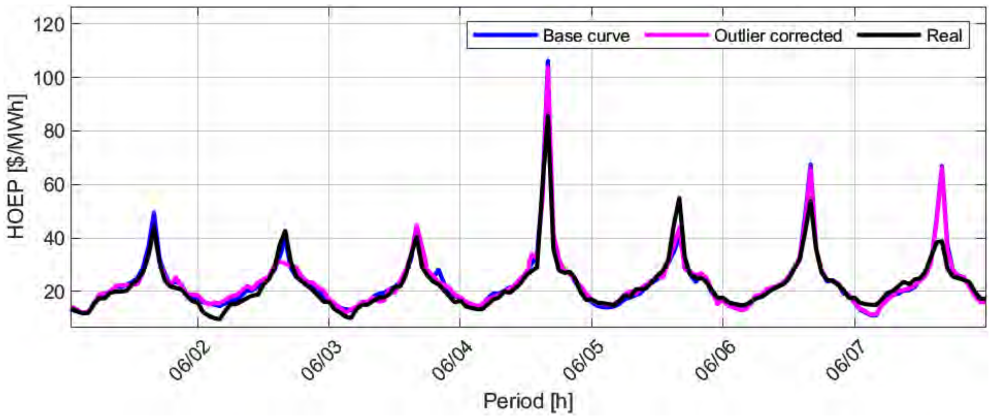


Fig. 6.3-11 Base curve, intelligent correction and real data of time series of the energy market.

In Table 6.3-11, a comparative analysis of the obtained results is shown by the proposal and the author´s work shown in [128].

	MAPE[%]	
Date	**Adaptive methodology**	[128]
June	**11.7**	13.66
July	**11.72**	21
August	25.2	**21.25**
September	**11.74**	20.35
October	**11.18**	566.35
November	**12.93**	18.34
December	**14.83**	16.14
Total	***14.22***	***96.72***

Table 6.3-11 Performance comparison (MAPE) of the proposed methodology.

It is possible to adjust the average *MAPE* from the author [128] without the month of October, in order to carry out a comparison in scenery without extreme data, which clearly shows a very large data error. Given this, the new *MAPE* would be equal to 18.07% showing a better performance by the proposed methodology.

6.4. Analysis of Results

Fig. 6.3-3 shows a case where the proposed methodology is applied for the forecasting of the energy demand in a building. The achieved performance index (*MAPE*) is suitable for the goals of the project.

Figs. 6.3-4 and Fig. 6.3-5 show the results obtained in the forecasting process for the energy demand of the electrical markets 1 and 2. The performance achieved by the *AM* methodology is superior for both markets in comparison to the models implemented by the author in [126] (see Fig. 6.3-6 and Fig. 6.3-7).

Fig. 6.3-8 shows the performance of each one of the modeling techniques included on the knowledge base for 7 days, the results show a better performance (accumulated average) for the recurrent neural networks (*LSTM*) and the Regression Tree Ensembles (*Tree*). Neural networks *(ANN)* present a poorer performance than regression vectors *(SVR)*.

Figs. 6.3-10 shows a better performance of the classification module and profiles construction (*profiling*) for identifying the type of day depending on the traffic profile.

Chapter 7

Conclusions and Future Works

This chapter summarizes the main conclusions arose by the analysis and discussion of the results reported in this work. The chapter also reviews the dissertation's scientific contributions and then discusses promising directions for future research and applications in certain topics in which the work of this thesis can continue. Finally, some concluding remarks are drawn.

7.1.Conclusions

The results and analysis carried out during this proposal have allowed highlighting the characteristics of this adaptive methodology taking into account the response variables discussed in section 6.1. The main objective, is to propose a methodology that will allow to emsamble models capables of fitting themselves into the characteristics of different time series (the lowest *MAPE*), without having their performance affected over time.

The building of consumption profiles proved to be a proper step towards forecasting time series. However, a preliminary base curve does not incorporate the effect of external variables such as weather conditions. Therefore, an intelligent correction (*SVR, ANN, LSTM* and *RTE)* is necessary. Such correction technique can be adjusted each period independently; however, it is possible to induce ripple in the output curve. Therefore, the intelligent correction agent is complemented with mechanisms of smoothing and outliers suppressing to guarantee the best performance. *LSTM* and *decision tree* showed a better performance than support vector regression (*SVR*) and the neural networks (*ANN*). However, adding several computational intelligence techniques within the knowledge base allow the consideration of multiples alternatives for other cases where the techniques may show a better performance.

The base curve (*profiling*) in the proposed methodology showed a better error rate in comparison with [127] in the classification task of typical traffic curves according to the type of day. This brings a support in the construction of the models since it allows simplifying the numbers of required models to characterize the behavior of the different time series to be considered.

The consideration of different modeling techniques allowed evaluating different alternatives for the same time series and the different profiles obtained in the *profiling* stage. The results show that the inclusion of more than one modeling technique helps in the process of characterization of the time series since it allows taking advantage of the strengths and minimizing the weakness in every case. The proposed adaptive methodology provides the possibility to build hybrids models in those cases where a unique technique does guarantee the better performance.

The implementation of the *Auditor* module and the retraining process help in the maintenance process of the models minimizing the decreasing rate in the performance of the assembly forecasting models. Once the need of retraining is identified by the *Auditor*, the system is able to evaluate the performance of each one of the available modeling techniques in the knowledge base for the window of the time series under study.

7.2. Main Contributions

This thesis presents an adaptive methodology for the characterization, assembly and automatic maintenance of time series models. The analysis, debugging, selection and transformation of the significant data related to the time series are proposed. The approach shows a better performance in the data classification and the forecasting processes of the time series compared in this work. Besides, it provided a recommendation capable of supporting maintenance of the available models. In the same way, considering diversity of the intelligent modeling techniques as: *ANN, SVR, LSTM* and *RTE* allowed the construction of hybrid models.

This thesis makes the following contributions in time series modeling problems related to energy demand, vehicular traffic and energy market data:

- *A computational analysis and characterization that integrate a multivariable stage for the identification, selection and transformations of significants variables needed for the construction of each models are developed.*
- *An adaptive and intelligent methodology for the modeling of the time series that integrates multiple computational intelligence techniques (ANN, SVR, LSTM and RTE) and an Auditor module to evaluate, adjust and retrain the selected models with the aim to keep the performance at the desired levels.*

7.3. Future Research and Directions

- ***Automatic data identification for atypical curves:*** the modeling process of the time series depending on the characterization of the data that results in a significant description of the time series. The atypical filtered data is a relevant work to avoid noise within the process of characterization and construction of the models. It is proposed to develop a module that is capable of detecting and filtering atypical data before the construction and the training processes of each models.

- **Time series modeling with variable granularity:** the modeling methodology allows characterizing data sets with hourly profiles (24 periods). However, other types of phenomena and processes occur with other granularities. Therefore, an extension of the present methodology that allows characterizing time series without regardless of their periodicity and granularity would provide wide applicability in different fields.

- **Time series depuration and data correction (Data augmentation):** the number of data is an inconvenient in the characterization of the time series. In some cases, the number of data becomes a limitation within the construction process of the models and even more when some of these are damage or incomplete. For this reason, it is suitable to propose of a module that allows correcting incomplete or atypical data with the objectives to increase the representative data within the analysis process of the time series.

References

[1] Q. H. Goulermas, J YWu, "Forecasting electric daily peak load based on local prediction," *2009 IEEE Power Energy Soc. Gen. Meet.*, pp. 1–6, Jul. 2009, doi: 10.1109/PES.2009.5275587.

[2] M. M. Dalvand, S. Bahram, Z. Azami, and H. Tarimoradi, "Long-term Load Forecasting of Iranian Power Grid Using Fuzzy and Artificial Neural Networks," in *Universities Power Engineering Conference, 2008.*, 2011.

[3] M. Beccali, M. Cellura, V. Lo Brano, and A. Marvuglia, "Forecasting daily urban electric load profiles using artificial neural networks," *Energy Convers. Manag.*, vol. 45, pp. 2879–2900, 2004, doi: 10.1016/j.enconman.2004.01.006.

[4] J. W. Z. Guo, W. Zhao, H. Lu, "Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model," *Renew. Energy*, pp. 241–249, 2012.

[5] M. Dai, Z. Yu, and R. Zeng, "A Neural Network Short-Term Load Forecasting Considering Human Comfort Index and its Accumulative Effect," in *2013 Ninth International Conference on Natural Computation (ICNC)*, 2013, pp. 262–266.

[6] L. C. Moreira de Andrade and I. Nunes da Silva, "Very Short-Term Load Forecasting Based on ARIMA Model and Intelligent Systems," *2009 15th Int. Conf. Intell. Syst. Appl. to Power Syst.*, pp. 1–6, Nov. 2009, doi: 10.1109/ISAP.2009.5352829.

[7] C. Garc, "Estimation and selection of ARIMA models," pp. 1–56, 2012.

[8] A. Amin, L. Grunske, and A. Colman, "An approach to software reliability prediction based on time series modeling," *J. Syst. Softw.*, vol. 86, no. 7, pp. 1923–1932, 2013, doi: 10.1016/j.jss.2013.03.045.

[9] P. Pai and W. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," *Electr. Power Syst. Res.*, vol. 74, pp. 417–425, 2005, doi: 10.1016/j.epsr.2005.01.006.

[10] S. Fan and L. Chen, "Short-Term Load Forecasting Based on an Adaptive Hybrid Method," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 392–401, 2006.

[11] K. Song, Y. Baek, D. H. Hong, and G. Jang, "Short-Term Load Forecasting for the Holidays Using," vol. 20, no. 1, pp. 96–101, 2005.

[12] K. Song, S. Ha, J. Park, D. Kweon, and K. Kim, “Hybrid Load Forecasting Method With Analysis of Temperature Sensitivities,” vol. 21, no. 2, pp. 869–876, 2006.

[13] X. Wang and M. Meng, “Forecasting Electricity Demand Grey-Markov Model,” no. July, pp. 12–15, 2008.

[14] T. Yalcinoz and U. Eminoglu, “Short term and medium term power distribution load forecasting by neural networks,” *Energy Convers. Manag.*, vol. 46, pp. 1393–1405, 2005, doi: 10.1016/j.enconman.2004.07.005.

[15] D. Liu, L. Zeng, C. Li, K. Ma, Y. Chen, and Y. Cao, “A Distributed Short-Term Load Forecasting Method Based on Local Weather Information,” *IEEE Syst. J.*, vol. 12, no. 1, pp. 208–215, 2018, doi: 10.1109/JSYST.2016.2594208.

[16] D. Fay and J. V. Ringwood, “On the influence of weather forecast errors in short-term load forecasting models,” *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1751–1758, 2010, doi: 10.1109/TPWRS.2009.2038704.

[17] J. Jimenez, A. Pertuz, C. Quintero, and J. Montana, “Multivariate statistical analysis based methodology for long-term demand forecasting,” *IEEE Lat. Am. Trans.*, vol. 17, no. 1, pp. 93–101, 2019, doi: 10.1109/TLA.2019.8826700.

[18] Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia, “An Ensemble Forecasting Method for the Aggregated Load with Subprofiles,” *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3906–3908, 2018, doi: 10.1109/TSG.2018.2807985.

[19] J. Jiménez, K. Donado, and C. G. Quintero, “A Methodology for Short-Term Load Forecasting,” vol. 15, no. 3, pp. 400–407, 2017.

[20] J. Wang, J. Hu, K. Ma, and Y. Zhang, “A self-adaptive hybrid approach for wind speed forecasting,” *Renew. Energy*, vol. 78, pp. 374–385, 2015, doi: 10.1016/j.renene.2014.12.074.

[21] S. Liu, S. Gu, and J. Peng, “Self-adaptive Processing and Forecasting Algorithm for Univariate Linear Time Series,” *Chinese J. Electron.*, vol. 26, no. 6, pp. 1147–1153, 2017, doi: 10.1049/cje.2017.09.027.

[22] U. Sarwar, M. B. Muhammad, and Z. A. Abdul Karim, “Time series method for machine performance prediction using condition monitoring data,” *I4CT 2014 - 1st Int. Conf. Comput. Commun. Control Technol. Proc.*, no. October, pp. 394–398, 2014, doi: 10.1109/I4CT.2014.6914212.

[23] A. Stepchenko, J. Chizhov, L. Aleksejeva, and J. Tolujew, “Nonlinear, Non-

stationary and Seasonal Time Series Forecasting Using Different Methods Coupled with Data Preprocessing," *Procedia Comput. Sci.*, vol. 104, no. December 2016, pp. 578–585, 2016, doi: 10.1016/j.procs.2017.01.175.

[24] R. Shumway and D. Stoffer, *Time Series Analysis and its Applications*, Third., vol. 19. Springer, 2017.

[25] J. H. Cochrane, "Time series for macroeconomics and finance," *Manuscript, University of Chicago*, no. 773. pp. 1–136, 2005, doi: 10.1111/j.1469-8137.2012.04105.x.

[26] K. W. Hipel and I. A. McLeod, *Time Series Modelling of Water Resources and Environmental Systems*. Elsevier Science, 1994.

[27] Q. MicroSoftware, "www.eviews.com," 2020. http://www.eviews.com/help/helpintro.html#page/content%2Fadvtimeser-Unit_Root_Testing.html%23ww184771.

[28] P. Rotela Junior, F. L. R. Salomon, and E. de Oliveira Pamplona, "ARIMA: An Applied Time Series Forecasting Model for the Bovespa Stock Index," *Appl. Math.*, vol. 05, no. 21, pp. 3383–3391, 2014, doi: 10.4236/am.2014.521315.

[29] B. Krithikaivasan, Y. Zeng, K. Deka, and D. Medhi, "ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 683–696, 2007, doi: 10.1109/TNET.2007.893217.

[30] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1054–1064, 2018, doi: 10.1109/TITS.2017.2711046.

[31] Y. Zhang, A. Haghani, and X. Zeng, "Component GARCH Models to Account for Seasonal Patterns and Uncertainties in Travel-Time Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 719–729, 2015, doi: 10.1109/TITS.2014.2339097.

[32] Y. Liu, B. F. Hseuh, Z. Gao, and Y. Yao, "Online flooding prognosis in packed columns by monitoring parameter change in EGARCH model," *2017 6th Int. Symp. Adv. Control Ind. Process. AdCONIP 2017*, pp. 300–305, 2017, doi: 10.1109/ADCONIP.2017.7983797.

[33] J. L. Aznarte and J. M. Benítez, "Algorithms for Regime-Switching Models," vol. 23, no. 11, pp. 1841–1847, 2012.

[34] A. Aliberti *et al.*, "A Multi-Patient Data-Driven Approach to Blood Glucose Prediction," *IEEE Access*, vol. 7, pp. 69311–69325, 2019, doi: 10.1109/ACCESS.2019.2919184.

[35] L. Xiao and Y. Pei, "Collaborative distribution knowledge service system based on task context aware," *Proc. - 2010 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IAT 2010*, pp. 146–151, 2010, doi: 10.1109/WI-IAT.2010.175.

[36] R. Mahía, "REVISIÓN DE LOS PROCEDIMIENTOS DE ANÁLISIS DE LA ESTACIONARIEDAD DE LAS SERIES TEMPORALES." UAM, 1999, [Online]. Available: https://www.uam.es/personal_pdi/economicas/rmc/doctorado/tendest.PDF.

[37] P. Gonzalez, *Análisis de series temporales : Modelos ARIMA*. Universidad del País Vasco, 2009.

[38] R. A. Jabbar *et al.*, "Neural Network (NN) based Demand Side Management (DSM)," *Univ. Power Eng. Conf. (AUPEC), 2010 20th Australas.*, 2010.

[39] R. Eberhart and Y. Shi, "Computational Intelligence concepts to implementations," 2007, pp. 17–25.

[40] A. Konar, "Computational Intelligence," Elsevier, 2005, pp. 1–8.

[41] E. S. de I. de Bilbao, "Redes Neuronales y Sus Aplicaciones," *EHU 2008/12*, 2008. http://ocw.ehu.es/ensenanzas-tecnicas/redes-neuronales-artificiales-y-sus- aplicaciones/contenidos/pdf/libro-del-curso.

[42] R. Flórez and J. M. Fernández, "Las Redes Neuronales Artificiales," Netbiblo, 2008, pp. 16–41.

[43] Fakhreddine O. Karray and C. de Silva, *Soft Computing and Intelligent Systems Design*. 2004.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "NEURAL MACHINE TRANSLATION," in *ICLR 2015*, 2015, pp. 1–15.

[45] W. Zaremba, "Recurrent Neural Network Regularization," in *ICLR 2015*, 2015, no. 2013, pp. 1–8.

[46] A. Graves, A. Mohamed, and G. Hinton, "SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS," in *ICLR 2015*, 2015, no. 3.

[47] B. Yang, K. Yin, S. Lacasse, and Z. Liu, "Time series analysis and long short-term memory neural network to predict landslide displacement," *Springer*, no. December 2018, pp. 677–694, 2019, doi: 10.1007/s10346-018-01127-x.

[48] Y. Du and M. Qi, "A Hybrid LSTM Neural Network for Energy Consumption Forecasting of Individual Households," *IEEE Access*, vol. 7, pp. 157633–157642, 2019, doi: 10.1109/ACCESS.2019.2949065.

[49] S. Dai, "Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction," *IEEE Access*, pp. 38287–38296, 2019, doi: 10.1109/ACCESS.2019.2907000.

[50] A. Azzouni and G. Pujolle, "A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction," *arXiv.org*, 2017, [Online]. Available: http://arxiv.org/abs/1705.05690.

[51] Vladimir Cherkassky and F. Mulier, "Learning From Data: Concepts, Theory, and Methods," in *Learning From Data: Concepts, Theory, and Methods*, 2nd ed., vol. 53, no. 9, I. John Wiley & Sons, Ed. John Wiley & Sons, Inc, 2007, pp. 1689–1699.

[52] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J.*, vol. 20, no. 1, pp. 3–29, 2020, doi: 10.1177/1536867X20909688.

[53] R. Adhikari K. and A. R.K., "An Introductory Study on Time Series Modeling and Forecasting Ratnadip Adhikari R. K. Agrawal," *arXiv Prepr. arXiv1302.6613*, 2013.

[54] S. Liu, S. Gu, and T. Bao, "An automatic forecasting method for time series," *Chinese J. Electron.*, vol. 26, no. 3, pp. 445–452, 2017, doi: 10.1049/cje.2017.01.011.

[55] G. C. Oniscu, A. Adair, and E. Hidalgo, "Severe preprocurement blunt trauma to the liver: Is there a need for back-table cutdown?," *Liver Transplant.*, vol. 15, no. 11, pp. 1647–1648, 2009, doi: 10.1002/lt.21880.

[56] G. P. Zhang and D. M. Kline, "Quarterly time-series forecasting with neural networks," *IEEE Trans. Neural Networks*, vol. 18, no. 6, pp. 1800–1814, 2007,

doi: 10.1109/TNN.2007.896859.

[57] A. F. Atiya, S. M. El-Shoura, S. I. Shaheen, and M. S. El-Sherif, “A comparison between neural-network forecasting techniques - Case study: River flow forecasting,” *IEEE Trans. Neural Networks*, vol. 10, no. 2, pp. 402–409, 1999, doi: 10.1109/72.750569.

[58] H. Yu, L. J. Ming, R. Sumei, and Z. Shuping, “A Hybrid Model for Financial Time Series Forecasting-Integration of EWT, ARIMA with the Improved ABC Optimized ELM,” *IEEE Access*, vol. 8, pp. 84501–84518, 2020, doi: 10.1109/ACCESS.2020.2987547.

[59] A. P. Alves da Silva, V. H. Ferreira, and R. M. G. Velasquez, “Input space to neural network based load forecasters,” *Int. J. Forecast.*, vol. 24, no. 4, pp. 616–629, 2008, doi: 10.1016/j.ijforecast.2008.07.006.

[60] T. M. K. G. Fernando, H. R. Maier, and G. C. Dandy, “Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach,” *J. Hydrol.*, vol. 367, no. 3–4, pp. 165–176, 2009, doi: 10.1016/j.jhydrol.2008.10.019.

[61] H. D. Tran, N. Muttil, and B. J. C. Perera, “Selection of significant input variables for time series forecasting,” *Environ. Model. Softw.*, vol. 64, pp. 156–163, 2015, doi: 10.1016/j.envsoft.2014.11.018.

[62] S. Rasouli, H. Tabesh, and K. Etminani, “A Study of Input Variable Selection to Artificial Neural Network for Predicting Hospital Inpatient Flows,” *Br. J. Appl. Sci. Technol.*, vol. 18, no. 4, pp. 1–8, 2016, doi: 10.9734/bjast/2016/30987.

[63] J. Xie and T. Hong, “Variable selection methods for probabilistic load forecasting: Empirical evidence from seven states of the United States,” *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6039–6046, 2018, doi: 10.1109/TSG.2017.2702751.

[64] M. R. Haq and Z. Ni, “A new hybrid model for short-term electricity load forecasting,” *IEEE Access*, vol. 7, pp. 125413–125423, 2019, doi: 10.1109/ACCESS.2019.2937222.

[65] F. Liu, Y. Lu, and M. Cai, “A hybrid method with adaptive sub-series clustering and attention-based stacked residual LSTMs for multivariate time series forecasting,” *IEEE Access*, vol. 8, pp. 62423–62438, 2020, doi:

10.1109/ACCESS.2020.2981506.

[66] G. Oğcu, O. F. Demirel, and S. Zaim, “Forecasting Electricity Consumption with Neural Networks and Support Vector Regression,” *Procedia - Soc. Behav. Sci.*, vol. 58, pp. 1576–1585, Oct. 2012, doi: 10.1016/j.sbspro.2012.09.1144.

[67] T. H. H. Aldhyani, M. Alrasheedi, A. A. Alqarni, M. Y. Alzahrani, and A. M. Bamhdi, “Intelligent Hybrid Model to Enhance Time Series Models for Predicting Network Traffic,” *IEEE Access*, vol. 8, pp. 130431–130451, 2020, doi: 10.1109/ACCESS.2020.3009169.

[68] L. B. Godfrey and M. S. Gashler, “Neural decomposition of time-series data,” *2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017*, vol. 2017-Janua, no. 7, pp. 2796–2801, 2017, doi: 10.1109/SMC.2017.8123050.

[69] S. Jiang *et al.*, “Combining Deep Neural Networks and Classical Time Series Regression Models for Forecasting Patient Flows in Hong Kong,” *IEEE Access*, vol. 7, pp. 118965–118974, 2019, doi: 10.1109/access.2019.2936550.

[70] A. Bauer, M. Zufle, N. Herbst, A. Zehe, A. Hotho, and S. Kounev, “Time Series Forecasting for Self-Aware Systems,” *Proc. IEEE*, vol. 108, no. 7, pp. 1068–1093, 2020, doi: 10.1109/JPROC.2020.2983857.

[71] Y. Yang and Y. Yang, “Hybrid method for short-term time series forecasting based on EEMD,” *IEEE Access*, vol. 8, pp. 61915–61928, 2020, doi: 10.1109/ACCESS.2020.2983588.

[72] K. Zheng, B. Qian, S. Li, Y. Xiao, W. Zhuang, and Q. Ma, “Long-Short Term Echo State Network for Time Series Prediction,” *IEEE Access*, vol. 8, pp. 91961–91974, 2020, doi: 10.1109/ACCESS.2020.2994773.

[73] J. Jaramillo, J. D. Velasquez, and C. J. Franco, “Research in Financial Time Series Forecasting with SVM: Contributions from Literature,” *IEEE Lat. Am. Trans.*, vol. 15, no. 1, pp. 145–153, 2017, doi: 10.1109/TLA.2017.7827918.

[74] S. Huang, S. Member, and K. Shih, “Short-Term Load Forecasting Via ARMA Model Identification Including Non-Gaussian,” *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 673–679, 2003.

[75] A. Ã. Azadeh, S. F. Ghaderi, and S. Sohrabkhani, “A simulated-based neural network algorithm for forecasting electrical energy consumption in Iran,” *Energy Policy*, vol. 36, pp. 2637–2644, 2008, doi:

10.1016/j.enpol.2008.02.035.

[76] T. Senjyu, H. Takara, K. Uezato, and T. Funabashi, “One-Hour-Ahead Load Forecasting,” *Trans. Eng. Comput. …*, vol. 17, no. 1, pp. 113–118, 2002, Accessed: Jan. 31, 2013. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:One-Hour-Ahead+Load+Forecasting#4.

[77] J. Olaya, “Un modelo spline para el pronóstico de la demanda de energía eléctrica,” *Rev. Colomb. Estadística*, no. 2, pp. 187–202, 2007.

[78] C. Pacheco *et al.*, “Electric load forecasting : evaluating the novel hierarchical neuro-fuzzy BSP model,” vol. 26, pp. 131–142, 2004, doi: 10.1016/S0142-0615(03)00060-7.

[79] G. A. Darbellay and M. Slama, “Forecasting the short-term demand for electricity Do neural networks stand a better chance ?,” vol. 16, pp. 71–83, 2000.

[80] OLANREWAJU, “Energy Efficiency Assessment Using Artificial Neural Network Combined with Data Envelopment Analysis,” 2011.

[81] Gabriel I. S. Ruas, “Electrical Energy Demand Prediction Using Artificial Neural Networks and Support Vector Regression,” 2008.

[82] D. C. Park, M. A. El-Sharkawi, R. J. Marks, L. E. Atlas, and M. J. Damborg, “Electric Load Forecasting Using an Artificial Neural Network,” *IEEE Trans. Power Syst.*, 1991.

[83] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, “Long-term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models,” in *IEEE Transactions on Energy Conversion*, 2006, pp. 273–284.

[84] H. S. G, M. Rastad, and J. Nazarzadeh, “A Hybrid Nonlinear Model for the Annual Maximum Simultaneous Electric Power Demand,” vol. 21, no. 3, pp. 1069–1078, 2006.

[85] K. Nagasaka and M. Al Mamun, “Long-term Peak Demand Prediction of 9 Japanese Power Utilities Using Radial Basis Function Networks,” pp. 1–8, 2004.

[86] A. Miguel, E. Gonza, and D. Carmona-ferna, “Forecasting of the electric energy

demand trend and monthly fluctuation with neural networks," vol. 52, pp. 336–343, 2007, doi: 10.1016/j.cie.2006.12.010.

[87] A. Mizutani, T. Yukawa, K. Numa, and Y. Kuze, "Improvement of Input-Output Correlations of Electric Power Load Forecasting By Scatter Search," pp. 429–433, 2005.

[88] R.-H. Liang and C.-C. Cheng, "Combined regression-fuzzy approach for short-term load forecasting," *IEE Proc. - Gener. Transm. Distrib.*, vol. 147, no. 4, p. 261, 2000, doi: 10.1049/ip-gtd:20000507.

[89] F. J. Martínez López, J. A. Torres Arriaza, S. Martínez Puertas, and M. M. Peralta López, "Multilevel neuronal architecture to resolve classification problems with large training sets: Parallelization of the training process," *J. Comput. Sci.*, vol. 16, pp. 59–64, 2016, doi: 10.1016/j.jocs.2016.04.002.

[90] M. Bataineh and T. Marler, "Neural network for regression problems with reduced training sets," *Neural Networks*, vol. 95, pp. 1–9, 2017, doi: 10.1016/j.neunet.2017.07.018.

[91] L. Wang, Y. Yang, R. Min, and S. Chakradhar, "Accelerating deep neural network training with inconsistent stochastic gradient descent," *Neural Networks*, vol. 93, pp. 219–229, 2017, doi: 10.1016/j.neunet.2017.06.003.

[92] J. W. Lin, C. T. Chao, and J. S. Chiou, "Determining Neuronal Number in Each Hidden Layer Using Earthquake Catalogues as Training Data in Training an Embedded Back Propagation Neural Network for Predicting Earthquake Magnitude," *IEEE Access*, vol. 6, pp. 52582–52597, 2018, doi: 10.1109/ACCESS.2018.2870189.

[93] A. M. Gracia, A. V. Delgado, A. A. Usón, I. Z. Bribián, and S. Scarpellini, *Disminución de Costes Energéticos en la Empresa*. 2006.

[94] C. H. Li, L. T. Yang, and M. Lin, "Parallel training of an improved neural network for text categorization," *Int. J. Parallel Program.*, vol. 42, no. 3, pp. 505–523, 2014, doi: 10.1007/s10766-013-0245-x.

[95] L. Gu, D. K. S. Tok, and D. L. Yu, "Development of adaptive p-step RBF network model with recursive orthogonal least squares training," *Neural Comput. Appl.*, vol. 29, no. 5, pp. 1445–1454, 2018, doi: 10.1007/s00521-016-2669-x.

[96] Y. Liang, J. Jiang, Y. Chen, R. Zhu, C. Lu, and Z. Wang, "Optimized Feedforward

Neural Network Training for Efficient Brillouin Frequency Shift Retrieval in Fiber," *IEEE Access*, vol. 7, pp. 68034–68042, 2019, doi: 10.1109/ACCESS.2019.2919138.

[97] M. Hacibeyoglu and M. H. Ibrahim, "A Novel Multimean Particle Swarm Optimization Algorithm for Nonlinear Continuous Optimization: Application to Feed-Forward Neural Network Training," *Sci. Program.*, vol. 2018, 2018, doi: 10.1155/2018/1435810.

[98] R. Hannah Jessie Rani and T. Aruldoss Albert Victoire, *Training radial basis function networks for wind speed prediction using PSO enhanced differential search optimizer*, vol. 13, no. 5. 2018.

[99] N. Chouikhi and A. M. Alimi, "Adaptive Extreme Learning Machine for Recurrent Beta-basis Function Neural Network Training," 2018, [Online]. Available: http://arxiv.org/abs/1810.13135.

[100] T. F. Awolusi, O. L. Oke, O. O. Akinkurolere, A. O. Sojobi, and O. G. Aluko, "Performance comparison of neural network training algorithms in the modeling properties of steel fiber reinforced concrete," *Heliyon*, vol. 5, no. 1, p. e01115, 2019, doi: 10.1016/j.heliyon.2018.e01115.

[101] A. Errachdi and M. Benrejeb, "Performance Comparison of Neural Network Training Approaches in Indirect Adaptive Control," *Int. J. Control. Autom. Syst.*, vol. 16, no. 3, pp. 1448–1458, 2018, doi: 10.1007/s12555-017-0085-3.

[102] Y. J. Wong, S. K. Arumugasamy, and J. Jewaratnam, "Performance comparison of feedforward neural network training algorithms in modeling for synthesis of polycaprolactone via biopolymerization," *Clean Technol. Environ. Policy*, vol. 20, no. 9, pp. 1971–1986, 2018, doi: 10.1007/s10098-018-1577-4.

[103] K. Moshkbar-Bakhshayesh, "Performance study of bayesian regularization based multilayer feed-forward neural network for estimation of the uranium price in comparison with the different supervised learning algorithms," *Prog. Nucl. Energy*, vol. 127, no. May, p. 103439, 2020, doi: 10.1016/j.pnucene.2020.103439.

[104] S. Ramyar and F. Kianfar, "Forecasting Crude Oil Prices: A Comparison Between Artificial Neural Networks and Vector Autoregressive Models," *Comput. Econ.*, vol. 53, no. 2, pp. 743–761, 2019, doi: 10.1007/s10614-017-

9764-7.
[105] B. Garg, N. Kirar, S. Menon, and T. Sah, “A performance comparison of different back propagation neural networks methods for forecasting wheat production,” *CSI Trans. ICT*, vol. 4, no. 2–4, pp. 305–311, 2016, doi: 10.1007/s40012-016-0096-x.
[106] N. A. Zambri, A. Mohamed, and M. Z. C. Wanik, “Performance comparison of neural networks for intelligent management of distributed generators in a distribution system,” *Int. J. Electr. Power Energy Syst.*, vol. 67, pp. 179–190, 2015, doi: 10.1016/j.ijepes.2014.11.005.
[107] R. Tabaraki and M. Khodabakhshi, “Performance comparison of wavelet neural network and adaptive neuro-fuzzy inference system with small data sets,” *J. Mol. Graph. Model.*, vol. 100, p. 107698, 2020, doi: 10.1016/j.jmgm.2020.107698.
[108] R. Kumar, R. K. Aggarwal, and J. D. Sharma, “Comparison of regression and artificial neural network models for estimation of global solar radiations,” *Renew. Sustain. Energy Rev.*, vol. 52, pp. 1294–1299, 2015, doi: 10.1016/j.rser.2015.08.021.
[109] M. Ebrahimi, A. A. Safari Sinegani, M. R. Sarikhani, and S. A. Mohammadi, “Comparison of artificial neural network and multivariate regression models for prediction of Azotobacteria population in soil under different land uses,” *Comput. Electron. Agric.*, vol. 140, pp. 409–421, 2017, doi: 10.1016/j.compag.2017.06.019.
[110] M. Khairalla, X. Ning, and N. AL-Jallad, “Modelling and optimisation of effective hybridisation model for time-series data forecasting,” *J. Eng.*, vol. 2018, no. 2, pp. 117–122, 2018, doi: 10.1049/joe.2017.0337.
[111] C. V Cardoso and G. L. Cruz, “Forecasting Natural Gas Consumption using ARIMA Models and Artificial Neural Networks,” *Rev. del IEEE América Lat.*, vol. 14, no. 5, pp. 2233–2238, 2016.
[112] C. Eras, J. J. C., Santos, V. S., Gutiérrez, A. S., Plasencia, M. Á. G., Haeseldonckx, D., & Vandecasteele, “Tools to improve forecasting and control of the electricity consumption in hotels,” *J. Clean. Prod.*, pp. 803–812, 2016.
[113] I. P. Panapakidis, “Electrical Power and Energy Systems Clustering based day-

ahead and hour-ahead bus load forecasting models," *Int. J. Electr. Power Energy Syst.*, vol. 80, pp. 171–178, 2016, doi: 10.1016/j.ijepes.2016.01.035.

[114] B. Zeng and C. Li, "Forecasting the natural gas demand in China using a self-adapting intelligent grey model," *Energy*, vol. 112, pp. 810–825, 2016, doi: 10.1016/j.energy.2016.06.090.

[115] X. Dai and M. Bikdash, "Trend Analysis of Fragmented Time Series for mHealth Apps: Hypothesis Testing Based Adaptive Spline Filtering Method with Importance Weighting," *IEEE Access*, vol. 5, pp. 27767–27776, 2017, doi: 10.1109/ACCESS.2017.2696502.

[116] Q. Han, J. Ding, E. M. Airoldi, and V. Tarokh, "SLANTS: Sequential Adaptive Nonlinear Modeling of Time Series," *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 4994–5005, 2017, doi: 10.1109/TSP.2017.2716898.

[117] R. Adhikari and R. K. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1441–1449, 2014, doi: 10.1007/s00521-013-1386-y.

[118] Z. Lin, D. Zhang, L. Gao, and Z. Kong, "Using an adaptive self-tuning approach to forecast power loads," *Neurocomputing*, vol. 71, no. 4–6, pp. 559–563, 2008, doi: 10.1016/j.neucom.2007.09.002.

[119] C. C. Aggarwal, "Outlier Analysis," in *Data Mining: The Textbook*, Cham: Springer International Publishing, 2015, pp. 237–263.

[120] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing*. 2007.

[121] Z. Wang, Z. Liu, and C. Zheng, *Introduction to neural networks*, vol. 34. 2016.

[122] U. de Girona, "E-LAND Project," *European Union's Horizon 2020 Research*, 2020. https://elandh2020.eu/.

[123] C. for M. L. and I. Systems, "Machine Learning Repository," *UC Irvine Machine Learning Repository*, 2011. https://archive.ics.uci.edu/ml/datasets/PEMS-SF#.

[124] Kashdotten, "Ontario Energy Prices," *Independent Electricity System Operator*, 2020. https://www.kaggle.com/kashdotten/ontario-energy-prices/metadata.

[125] D. Montgomery, *Diseño y Análisis de Experimentos*, 2nd ed. Mexico D.F.: Wiley,

Limusa, 2005.

[126] XM, “INDICADORES DE PRONÓSTICOS OFICIALES DE DEMANDA,” 2019. https://www.xm.com.co/Paginas/Consumo/indicadores-de-pronosticos-oficiales-de-demanda.aspx.

[127] M. Cuturi, “Fast global alignment kernels,” *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*, pp. 929–936, 2011.

[128] K. B. Sahay, “One hour ahead price forecast of Ontario electricity market by using ANN,” *Int. Conf. Energy Econ. Environ. - 1st IEEE Uttar Pradesh Sect. Conf. UPCON-ICEEE 2015*, 2015, doi: 10.1109/EnergyEconomics.2015.7235102.