Universidade do Algarve

Faculdade de Ciências e Tecnologia

Neural models design for solar radiation and atmospheric temperature forecast

Igor António Calé Martins

Masters in Electronics and Telecommunications Enginering

2010

Universidade do Algarve

Faculdade de Ciências e Tecnologia

Neural models design for solar radiation and atmospheric temperature forecast

Igor António Calé Martins

Supervisors

António Eduardo de Barros Ruano

Pedro Miguel Frazão Fernandes Ferreira

Masters in Electronics and Telecommunications Enginering

2010

Abstract

This work arises from the necessity of temperature and solar radiation forecast, to improve the Heating, Ventilating, and Air Conditioning (HVAC) systems efficiency. To do so, it was necessary to determine neural models capable of such forecast. The chosen characteristics were solar radiation and temperature because these two characteristics directly affect the room temperature inside a building. This forecast system will be implemented on a portable computational device, so it must be built with low computational complexity.
During this dissertation the various research phases are described with some detail. The applications were developed on *Python* programming language due to it library collection. In this task several algorithms were developed to determine the cloudiness index. The results of these algorithms were compared with the results obtained using neural models for the same purpose. In solar radiation and temperature forecast only neural models were used. The cloudiness index forecast was not implemented as this is only an intermediate step; instead measured values of cloudiness index were used for the solar radiation forecast. Regarding the solar radiation forecast two neural models were implemented and compared, one of the models has an exogenous input, the cloudiness index forecast, and the other one is simply a time series. This models were compared to determine if the inclusion of the cloudiness index forecast improves solar radiation forecast. In temperature forecast only one model will be presented, a Nonlinear AutoRegressive with eXogenous input (NARX) model, with solar radiation forecast as exogenous input.
All the neural models are radial Basis Function (RBF) and there structure was determined using a Multi-Objective Genetic Algorithm (MOGA). The models were used to determine cloudiness index, forecast solar radiation and temperature.

Key-words: cloudiness index determination, cloudiness index forecast, Multi-Objective Genetic Algorithm, neural networks radial basis function, solar radiation forecast, temperature forecast.

Resumo

Este trabalho surge da necessidade de se prever a temperatura e a radiação solar de forma a aumentar a eficiência de sistemas AVAC (Aquecimento, Ventilação e Ar Condicionado). Para tal foi necessário determinar modelos neuronais que permitam a implementação de tal previsão. As características escolhidas foram a radiação solar e a temperatura uma vez que estas afectam directamente o clima no interior dos edifícios. Este sistema de previsão deverá ser incorporado num dispositivo computacional portátil, portanto com uma capacidade computacional reduzida.

Ao longo do trabalho são descritas as várias fases de investigação, de forma a relatar detalhadamente o trabalho efectuado. A linguagem utilizada para o desenvolvimento de aplicações foi o *Python* devido à existência de uma vasta colecção de bibliotecas. Neste trabalho foram desenvolvidos algoritmos para determinação de nebulosidade simulando a previsão da mesma. Os resultados dos algoritmos implementados foram comparados com os resultados das redes neuronais implementadas para o mesmo fim. No caso da previsão de radiação e temperatura apenas foram utilizadas redes neuronais. Não será feita a previsão de nebulosidade uma vez que esta é apenas um passo intermédio, sendo usado para previsão de radiação solar os valores de nebulosidade medidos. Para a previsão de radiação solar serão implementados dois modelos neuronais. Um dispõe de uma entrada exógena, a percentagem de nebulosidade, sendo o outro apenas uma série temporal.

Os modelos neuronais são todos de função de base radial (RBF) e a sua estrutura foi determinada através do algoritmo genético multi-objectivos (MOGA). Os modelos foram utilizados na determinação de nebulosidade, previsão de radiação e temperatura.

Palavras-chave: Algoritmos Genéticos Multi-Objectivo, Determinação nebulosidade, Previsão nebulosidade, Previsão radiação, Previsão temperatura, Redes Neuronais Função Base Radial.

Acknowledgements

I would like to show my gratitude to those who made this dissertation possible:

- My coordinators;
- My family;
- My special friends and colleagues;
- The project "PTDC/ENR/73345/2006 – Intelligent use of energy in public buildings";

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

The waste of energy is an important subject nowadays. Part of electrical energy is now obtained from renewable sources. The inefficient energy consumption damages the environment and causes the so called global warming.
To decrease the global warming the human race has to decrease the pollution. Knowing that the climate control in buildings is one of the biggest energy spenders in Europe, a better efficiency in this system will have impact in the energy waste. Inside climate in buildings not only depends on the HVAC system(s), but also in the outside climate. The availability of a system able to forecast the outside temperature and solar radiation could enable the maintenance of thermal comfort, while simultaneously allowing a reduction in the energy spent to achieve it.
The work described in this report could also be applied in renewable energy too, specifically in solar thermal systems and photo voltaic panels. With the radiation forecast a better scheduling of these systems could be obtained.

1.2. Proposed Target

The target proposed is to design neural models that allow the forecast of the total solar radiation and outside temperature, with in a four hours horizon. These models are based in data, but physical knowledge can also be incorporated. The prediction of solar radiation should be compared between two models, a Nonlinear AutoRegressive (NAR) and NARX, where the exogenous input will be the cloudiness index. The cloudiness index will be determined with sky images in the same forecast horizon. Only one temperature forecast model will be built, a NARX model, with total solar radiation forecast as the exogenous input. The forecast of this last variable will use the best method selected, considering the NAR and NARX systems. The structure of all this models should be determined off-line, using an evolutionary hybrid/training mechanism approach based on Levenberg-Marquardt (LM) method.

1.3. Main Contributions

In this dissertation several Neural Networks (NN) were designed and implemented for achieving the proposed objectives. The neural networks were implemented for total solar radiation forecast, outside temperature forecast and cloudiness index determination. A linear method was implemented for cloudiness index determination and compared with the neural network made for the same objective.

1.4. Thesis structure

The first chapter presents an introduction to the thesis and the work developed. The objectives and motivation of this project are described and it is explained how the dissertation is written and structured.

The second chapter presents the literature review which is divided in sections, beginning with a briefly introduction. This chapter approaches the topics discussed and used in this report, such as, total solar radiation, outside temperature and cloudiness index.

In the third chapter a background on the important devices and techniques that will be discussed in forwards chapters is presented. The systems and applications that already exist are described as well as the computational models used in this thesis.

The fourth chapter will present the models developed for cloudiness index determination. Two approaches of each technique, a linear method and the Artificial Neural Network (ANN) method, are reported. Each section will present an introduction, a formulation and the obtained results.

The two following chapters, the fifth and the sixth, will detail the total solar radiation and outside temperature forecasting, respectively. In the first one two neural models will be presented and in the latter only one model will be designed. Both chapters will be divided in three section with a briefly introduction, a formulation and a presentation of the results.

The seventh and last chapter will conclude the dissertation with the conclusions, future work and references used in this thesis.

2. Literature Review

This study is directly related with solar radiation and temperature forecast, although there are several associated subjects with this work, such as, cloudiness indexes and neural networks.

The global solar radiation is divided in two components, direct and diffuse radiation. The direct radiation in a plane point of earth will be affected by the solar declination, zenith angle, atmospheric gases, altitude, relative humidity and cloudiness; these components affect also the temperature. In this work only the last factor (cloudiness) will be considered. The cloudiness index case could be presented in two different points of view, from the space or from the earth. This last one is the most important because the study will focus that view.

The research will start to be presented by the cloudiness index.

2.1. Cloudiness Index

The authors of [3] presents an method, different from the usual approaches. The authors define four kinds of clouds reflection: clean sky, and uncover sun that reflects on a cloud, partial covered sun and totally covered sun. The presented algorithm could be used to estimate the sky daily and hourly conditions, characterized in three classes: clean sky, partially covered sun and totally covered sky. Some articles [4, 5, 6] reported cloudiness indexes catalogued in other ways and with several weather estimation.

The method used in this article [3] to characterize the sky is based in the solar radiation values, estimating a cloudiness index from the past 5 minutes of the solar radiation. This interval, according to the authors, and supported by [7], is due to the observation that the distribution frequency in five minutes is nearly identical to instantaneous values; with a larger interval this relation is not verified.

The data used in this paper [5] was acquired in Botucatu in São Paulo, Brazil, where the solar radiation was measured with a Spectral precision pyranometer, and the direct solar radiation was measured through a paired pyrheliometer (ST-3). The measure period was about six years, between 1996 and 2001. The proposed algorithm was based on the assumption that cloudiness variations in a short period of time are due to the relative optic

air mass. As clean and cloudy sky the distributions and frequencies are different, it was necessary to divide the categories and choose a limit. According to [5] the limits are well defined and could be mathematically justified. In the algorithm a matrix is constructed that contains four columns and the first column of each line represents the instantaneous sky condition. Through this matrix is possible to analyse the data and to get the effective day length.

The fifth chapter of the article [5] presents the application of the algorithm, with Botucatu data. The error results obtained are afterwards compared with a recorder device Campbell-Stocks; the statistical errors obtained were small. The authors warn, however, that to use this algorithm in other places it is necessary to have, at least, five years of records of solar radiation. They also suggest that the algorithm should be compared with others before a final application, to compare the performance.

Still with in the cloudiness index topic, particularly in the data analysis of clouds patterns and recognition, the article [8] mentions a very interesting and functional method. The authors used two different devices to capture the sky images: the Total Sky Imager (TSI), that obtains a 160 degree view of the sky from horizon to horizon, and where the images are obtained in Joint Photographic Experts Group (JPEG) format with 352×288 pixels of size (later these images were resized to 256×256 pixels using a bilinear interpolation through MATLab® software) and the Whole Sky Imager (WSI), consisting in a digital colour camera, with a fish-eye lens (about 180 degrees of vision) pointed to the zenith. The obtained images are presented in Bit Map Picture (BMP) format with 768×576 pixels of size, yet, the authors refer that a JPEG format with 80% of compression was the ideal to process the image. The authors, based in previous articles [9] and his own experiments, defined eight conditions that could be used in classification, related with conditions due to the clouds visual aspects, instead of the classical clouds classification.

In the images characterization, the image texture is analysed based in histogram properties. This process could only be applied to a grey scale image whose pixels could be described with a single value. So the Red, Green and Blue (RGB) images had to be transformed into a grey scale. Therefore two transformed images were employed, one with the component red to blue (R/B) and an image with the values intensity, defined as a sum of the third part of each component.

Relatively to the clouds patterns, the authors used the Fourier transform of the images, because the image analysis in the frequency domain could be a help to identify the differences between classes [8]. It is important to mention that as the values are real, the spectral function is symmetric to the origin. The power spectrum functions of different sky images are, obviously, different from each others. So it is necessary to extract simple characteristics of the spectral function, to classify the cloud type.

In this article the authors chose to use two characteristics, the first being the Correlation of the Clean sky (or Correlation with Clear, CC) and the second the Spectral Intensity (SI). The first serves to quantify the difference between the spectral power function of the image in analysis and the spectral power function of a clear sky image used as reference. Specifically the CC value is the linear correlation coefficient between the logarithms of two spectral power functions. This value is ranged between 0 and 1, and the higher this value the more uniform the sky aspect is. The Sky imager also takes into account the power of spectral distribution over the wave numbers, i.e., depending on the presented clouds patterns, each particular wave number should have more or less spectral power.

The described methods consider all image pixels, equally not distinguishing between clean sky pixels and cloud pixels. So there was a need to add other characteristics, to help the sky recognition. This distinction between pixels was made through the image R/B threshold. The threshold value was computed in a MATLab® routine (*graythresh*), which calculated the value based on the image histogram [10]. Because only some image sections are analysed, the problem of analysing critical zones, such as near the horizon, does not exist.

According to the article [8], is possible to recognize sky patterns and subsequent clouds classification, using the statistical methods referred above. The authors of this article [8] exploit two additional characteristics: Fourier transform based, and pixels distinguished based. In the Fourier transform case, the main problem is due to the limited number of values in the Fast Fourier Transform (FFT) information. In terms of the technique involving a threshold to differentiate pixels, is important to refer that the results could be better if a varying threshold, taking into account the proximity of the pixel position to the sun, was employed.

Neural models, of the MultiLayer Perceptron (MLP) and the RBF type, were also used

to determine related cloudiness indexes. In [11] the authors present a neural network that determines the cloudiness indexes using the latitude, longitude, altitude, light intensity and month of the year. The authors approach the problem as a multi variable interpolation in which it is required to estimate the function relating the input to the output using a set of input-output data.

In the RBF Neural Network (RBFNN) used, the transfer function is expressed as a linear combination of multivariate Gaussian basis functions, the technique used to train the network was based on assuming fixed RBF, the training function updates the weights and bias values according to LM optimization. The centres were randomly selected from the training data set. The width of the Gaussian RBF (σ_j) was expressed in terms of the maximum distance between the chosen centre d and the number of centres M, $\sigma_j = \frac{d}{\sqrt{2M}}$. The author used a retro-propagation algorithm implemented in the neural networks toolbox of MATLab®.

The MLP network consists in one input layer, one output layer and one, two or three hidden layers. The architecture used in this model has five neurons in the first layer, one hidden layer with sigmoid activation function, and an output layer with a linear activation function. For training this network was used the same retro-propagation algorithm belonging to the neural network toolbox of MATLab®. In the end the results were selected using the Root Mean Square (RMS) error.

The data sets were collected from eight meteorological stations during ten years, and were used to train three neural networks MLP and one RBFNN. The output parameter was the cloudiness index and to get the solar radiation the author used this index and multiplied it by the maximum radiation index.

In conclusion both network types had reasonable results, but the RBFNN model was preferred due to its best results and smaller computational time.

2.2. Solar Radiation

In the field of solar radiation, the authors in [12] chose to develop a work that introduces a new variable in this topic, the relative humidity. In the paper the authors try to prove the inclusion of this variable in solar radiation estimation and forecast. The data were collected in 12 stations between January of 1998 and 2002, in a total of 1702 points,

collected in Saudi Arabia. The data consisted in temperature, relative humidity and global solar radiation values.

The model used to estimate the solar radiation, using as variables the temperature and relative humidity, was a MLP, with four inputs, two hidden layers, one with three neurons and the other with four, and two output neurons. The algorithm used to train the network was the Back Propagation (BP) algorithm with non-linear sigmoid threshold.

After introducing this neural model, the authors present three new artificial neural networks models different from the first one, all with just a hidden layer and with different inputs. For the three cases, the data used to derive the models was from 1998 to 3002, where the first 1462 points were used to train the network and the remaining 240 were used to test the model. The first of these three networks has as inputs the maximum day temperature and the day of the year, and has 24 neurons in the hidden layer. The second network used two inputs: the mean day temperature and the day of the year. The neural structure has 32 neurons in the hidden layer. The last network has 24 neurons on the hidden layer and three input neurons which are the mean day temperature, the day of the year and the daily mean of the relative humidity.

Analysing the results presented in the article [12], the relative humidity is a parameter that could improve the results of solar radiation estimation, as in this paper the third model obtained the best results.

The model used in article [13] is, once again, an artificial neural network, as as it was verified by the authors [13], these models achieve more accurate results for the solar radiation forecast and estimation than other models. The data used was collected by 12 stations localized near the *Sierra Nevada*, in the city of *Huéneja* in *Granada*, Spain. In the study, one of the 12 stations were chosen to supply the data as an input to the artificial neural networks, and the data of the remaining stations were used to generate comparison values.

The data used was solar radiation and daily clearness index, the solar radiation was obtained with a LICOR 200-SZ pyranometer between 2003 and 2005 with an interval of 2.5 minutes, the daily clearness index was obtained in the same interval with a H8 HOBO. Values of altitude, slope and azimuth of each station were also added to the data set. The artificial neural network used was an MLP with a hidden layer, for its simplicity.

The activation function was a hyperbolic tangent. In this paper a method is presented, Automatically Relevant Determination (ARD), based on the Bayesian methodology [14] to determine the artificial neural networks parameters. This method limits the complexity of the networks identifying the variables relevance, and avoids the data separation in subsets confining the maximum data number. This method is based on a probabilistic interpolation of the weights possible configuration.

The development of the MLP is composed by three steps: selecting the inputs through the ARD method, training the MLP with the selected variables, and finally the output values were compared with the measured values. The results obtained were that the day of the year and the cloudiness index are very important to solar radiation forecast and estimation in these conditions, because using these variables the results were more accurate.

Still in the solar radiation field and incorporating cloudiness indexes, in [15] another work is presented, involving also the image acquisition for solar radiation forecast.

The ANNs types used in this paper were RBF composed by three layers: the input layer is composed by the sensory inputs, i.e., the data from the sensors, the second is a hidden layer composed by several neurons chosen by a MOGA, the third and last layer has a neuron that corresponds to the output. The artificial neural networks training was divided in two parts: the first one an unsupervised procedure, the Optimal Adaptive K-Means clustering (OAKM) algorithm used to compute the initial position of the centres, the spreads being computed by a simple algorithm. The second part consists on the application of the LM algorithm for parameter refinement.

To verify the influence of the cloudiness index on the solar radiation forecast it was necessary to compare three methods, one without cloudiness index and two others with a different cloudiness index as exogenous input. One of the NARX models takes into account all the sky and the other just an ellipse area near the sun.

The device used to acquire sky images is a TSI, through a Charge Couple Device (CCD) camera. This device has a black strip to protect the CCD from the reflected sun on a concave mirror. After collecting the images, these are processed to get its cloudiness index (in percentage). The sky was classified by clean sky, clouds with low density level and high density level. The system used had an algorithm that classified the sky, however the results obtained were not satisfactory, especially when the sky has a different hue.

To determine the cloudiness index importance in solar radiation forecast, two NARX models were used with different types of cloudiness index, as it was described before. These models were later compared with a NAR model.

To build a realistic model of solar radiation three different models were used: a NAR Solar Radiation (NARSR) model; a NARX Solar Radiation (NARXSR) model, with the Percentage of Clear Sky (PCS) values as exogenous input; and a NAR Percentage of Clear Sky (NARPCS) model. The data used to train the networks has 10080 points collected in 6 days, the validation was made in four days with a total of 7200 points. The acquisition method consists in a solar radiation sensor. The lag terms were chosen by the MOGA, after a heuristic relation and having into account the solar radiation variation through the time, and the relation between the solar radiation and the cloudiness index history. In the NAR model the number of inputs was compressed to 10% and 5% in the other models. The model evaluation was implemented using the training errors, One-Step Ahead Training Error (OSATE), generalization error, One-Step Ahead Generalization Error (OSAGE) and the errors in long time forecast.

According to the results in the article [15], the cloudiness index is a relevant parameter in the solar radiation forecast, however the training methods are not optimized and gave origin to a large error in cloudiness index determination.

We will present an article that compare neural models with empirical ones [16], the authors considered two classic approaches to simulate solar radiation on earth surface [17]. The first approach is based in complex radiation transfer [18], and the second is based in the empiric relation between the solar radiation ratio in the earth surface and the solar radiation ratio in the atmosphere.

Among the classical methods, the solar radiation could be estimated by radiation measurement interpolation [19] or by satellite images analysis [20]. The chosen models for comparison in this article [16] were the Hargreaves and Samani (HS), the Mahmood and Hubbard (MH) and the unbiased MH. The HS model developed by Hargreaves and Samani [21], is a simple empirical model to simulate daily solar radiation received in an horizontal surface to estimate the evapotranspiration. The MH model was developed by Mahmood and Hubbard [22] and uses a different approach. In this empirical model proposed, the transmissivity is a function of the day of the year, and the daily solar radiation received by

the surface is correlated with temperature amplitude and extraterrestrial radiation. This last model is identical to the previous one the main difference being the unbiased characteristic. These models were optimized using a standard non-linear unrestricted scheme.
The neural network model used was the MLP, trained with the LM, algorithm with a Bayesian regulation. The selected inputs were the extraterrestrial radiation, the day of the year and maximum and minimum air temperature. The best performance was achieved with eight neurons on the hidden layer.
The collected data was the temperature and solar radiation obtained from 11 meteorological stations, where the data from six of them were used for training and optimization of the empirical methods, and the other five stations were used to test. Each station supplies about 225 data sets of solar radiation, in the years 2004 and 2005. Each variable has 2522 observations, 1385 of them for training and 1137 for validation.
The empirical model that obtained best results was the HS. Even after the error optimization, the errors obtained by the HS were greater than the errors obtained by the second best neural network. These results were almost unbiased.

2.3. Temperature Estimation

In the work presented in [23] the design of inside air temperature predictive neural network models, to be used for predictive control of air conditioned systems, is discussed. The authors of this paper propose to use a neural network approach over the physical models, due to its time consuming. The data set came from a system implemented in a secondary school building, in Portugal, where 90% of the year the sky is clear. The inside and outside environmental information was collected between December 2003 and January 2004. The simulation was divided into two parts. In the first, used to validate the physical model, the air temperature numerical values are compared to the experimental values. In the second part a real situation was simulated. The model structure was determined using a MOGA, while parameter estimation will be performed by the application of the Levenberg–Marquardt algorithm. The neural network models used were all RBF with 14 to 18 inputs. The input variables considered were: inside air temperature, outside solar radiation , air temperature and relative humidity. Regarding the neural networks inputs presented, analysing the results achieved , the inclusion of relative humidity was

considered.

In conclusion, it was verified that the model reproduces well the experimental values, the difference between numerical and experimental air temperature values is lower than 2°C, the maximum difference was verified in the compartments with windows turned South, during afternoon. However it was shown that by using multi-objective genetic algorithms for the off-line design of radial basis function neural networks, the neural models can achieve better results. With an adaptive methodology the neural networks could achieve good results throughout the whole year. The authors demonstrate, in simulations, that using long-range predictive models for air-conditioning systems control could lead to important energy savings.

The work presented in [24] aimed to show the suitability of neural networks to perform predictions. Nonlinear AutoRegressive with eXternal input (NNARX) neural model were employed and genetic algorithms were employed to design the networks.

The authors aimed to:

- examine prediction suitability of neural networks when physical models are unable or very difficult to applly;
- examine two important techniques: k-step ahead prediction [25] and delta method [26] for the prediction interval;
- present a comparison between the two approaches to search for an optimal neural network structure.

The data were taken from a pilot house where the indoor temperature and the relative humidity were very much affected by ventilation machines. All the neural networks has one hidden layer and were trained using the fast training algorithm Levenberg–Marquardt to perform a four-steps ahead prediction. The Matlab® based Neural Network Based System Identification Toolbox and Neural Network Toolbox were used as training tools. The estimate of the average generalization error was analysed too, two techniques were considered: leave-one-out cross-validation and bootstrap, the first one has an alternative version, the linear-unlearning-leave-one-out which is faster.

Four neural networks models had satisfactory results in the testing stage with correlation coefficients of 0.998 and 0.997 for indoor temperature and relative humidity, respectively. Indoor temperature predictions perform much better than those of indoor relative hu-

midity. The authors concluded that increasing the number of inputs does not improve prediction accuracy, but avoid that more work need to be done.

Article [27] describes inside temperature forecast using artificial neural networks, and compares the results with classical models. The authors used the AutoRegressive with eXogenous input (ARX) and AutoRegressive Moving Average with eXogenous input (ARMAX) models, to compare with the artificial neural networks. This work studies how the temperature increases in the presence of heating systems. Two buildings with different heating sources were analysed. The models used were identical in the two buildings. Three forecast were used models to compare with the neural networks: first order model forecast with one step ahead prediction, first order model forecast with two steps ahead prediction and a second order model forecast with one step ahead prediction. The parameters of the ARX model were identified by the least squares method and the parameters of the ARMAX model were identified using the iterative Gauss-Newton algorithm. The artificial neural networks structure has a hidden layer with sigmoid function, a linear output, and the training algorithm used was the Levenberg-Marquardt. The comparison criteria used were the mean square error and the absolute error.
The artificial neural networks forecast achieved accurate results and the authors concluded that solar radiation and the time of the day were important inputs to forecast the inside temperature.

3. Background

3.1. Sky Imager

There are two main systems to capture clouds images: one with satellite images of earth, that capture clouds, earth and water surface, and the most important for this dissertation, the Ground-Based All-Sky (GBAS). This last one takes images of the sky from earth, it capture clouds, sky and sun. The existing systems studied during the work were the WSI and the TSI, shown in Figure 3.1.

(a) Whole Sky Imager (b) Total Sky Imager

Figure 3.1: Ground-Based All-Sky devices (figures acquired from http://www.arm.gov and http://www.nrel.gov, respectively).

The first system mentioned, WSI, is an automated ground-based electronic imaging system used for assessing and documenting cloud fields and cloud field dynamics. It is a passive system that acquires images of the sky. It uses three spectral filters (neutral, red, and blue), to determine the presence, distribution, shape, and radiance of clouds over the entire sky using automated cloud decision algorithms and related processing.
The TSI has more than one model; in this work the model 440A (TSI-440A) will be described, as it was used to capture the sky images used in this thesis. The TSI-440 captures RGB colour images of the sky through a concave mirror. To protect the CCD-

camera the mirror has a band that covers the sun. This band is attached to the mirror that slowly turns with the help of a motor. The coordinates of the TSI location are used to adjust the band. For the band system to work, the TSI has to be aligned with the Geographical North Pole. A brief description of the operation of the capture system of this device is as follows: the sky images are captured via a fixed CCD imaging camera downward pointed to a hemispherical mirror, as described before. The mirror reflects the image of the hemisphere over the system into the lens, blocking the intense direct-normal radiation from the sun through a shadow band that work as we described earlier. After that the image is software processed.

3.2. Cloudiness Index Estimation

In the previous section the systems used to capture the GBAS were presented; in this section the application or algorithms studied and used in this dissertation will be described. The two GBAS described before have an application to determine the cloudiness index, but only the application used in TSI-440 will be presented in this section. The other algorithm, briefly described before, was created and presented in [15] and is the one used since 2004 to determine the cloudiness index.

The TSI-440 system captures images in JPEG format, which are analysed for fractional cloud cover. The system can be connected to a Transmission Control Protocol/Internet Protocol (TCP/IP) network, and acts as a sky image server to web browsers via the Internet. In many meteorological forecasting applications, the accurate determination of sky conditions is a desirable but difficult goal. According to [28], trained human observers cannot equally classify the cloudiness index, due to the subjectivity in observations. So it is impossible to have a gold standard. In TSI-440, after the capture, an image processing program on a PC workstation acquires images via TCP/IP, with a certain interval, and saves them in JPEG format for analysis, as described before. The software first step is to mask the non sky pixels of the image, like the sun-blocking shadow band, camera, its arm and landscape. The raw colour image is saved and the masked image is analysed for cloudiness index, and after that, saved.

The algorithm used to process the images since 2004 is an artificial neural network method. This method uses the same structure that the previous mentioned, the TSI-440 method.

Like the previous method, this method identifies two types of clouds, the denser and the thinner. This method uses RGB images without any transformation beyond the mask. This is a simple method created and used due to the weak results made by the TSI-440 cloudiness index determination software.
In this work a new algorithm that will be described later was developed, achieving better results than any of existing algorithms.

3.3. Neural Network Method

The RBF neural network were introduced as ANN in 1988 [29]; nowadays this type of NNs presents acceptable results with lower computational costs than the MLPs. As it is possible to observe in Figure 3.2 a RBFNN has three layers. The first layer is the input of the network from the outside environment. The second layer, also denominated as hidden layer, applies a non-linear transformation on the input set, this layer usually has a large number of neurons to achieve better results. The third, and last, layer does a linear combination of the outputs of the neurons in the previous layer. The output of this layer is the final result of the neural network; usually this layer has a single neuron.

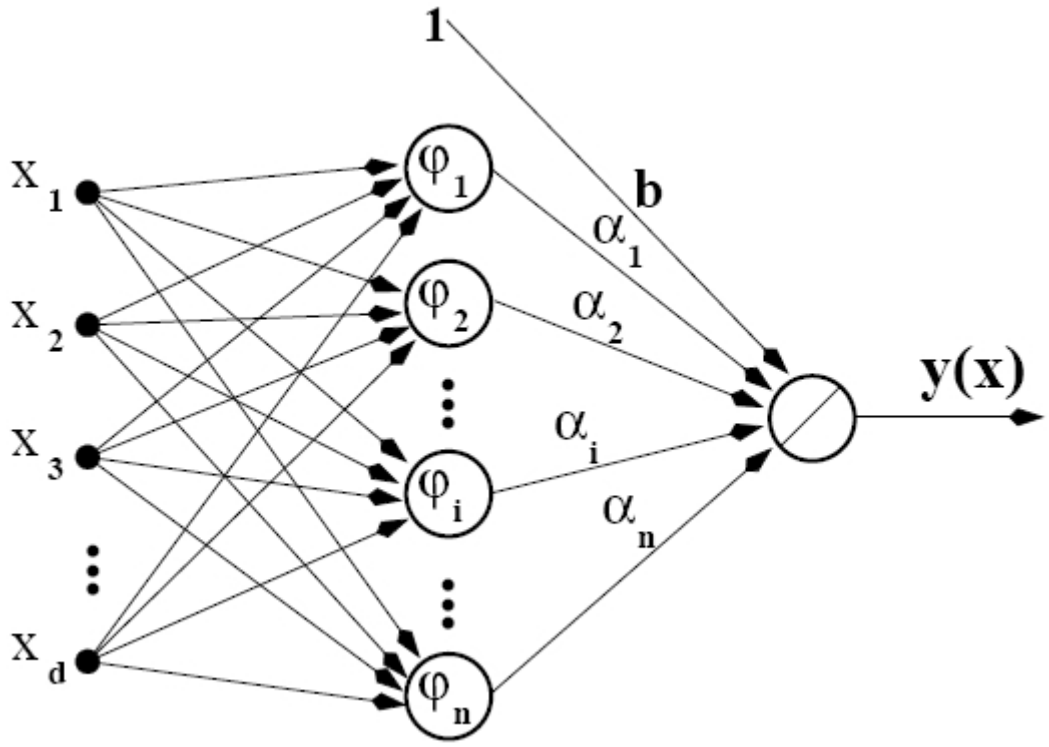

Figure 3.2: ANN generic structure scheme to a RBF model (image based on [1]).

The hidden layer is formed by a neuron set, each one represented by a radial basis function, given by:

$$\varphi_i(x) = \gamma(|c_i - x|) \tag{3.1}$$

where γ represents a transformation (usually non linear), $c_i \in \mathbb{R}^d$ is the centre of the function, where d is the number of inputs in the network, and $x \in \mathbb{R}^d$ is the inputs of the network. Note that $\mathbb{R}^d$ is the real space with dimension d, the number of inputs in the network. Some possible base functions are:

$$f(r) = r \qquad \text{radial linear function} \tag{3.2a}$$

$$f(r) = r^3 \qquad \text{radial cubic function} \tag{3.2b}$$

$$f(r) = e^{\frac{r^2}{2\sigma^2}} \qquad \text{Gaussian function} \tag{3.2c}$$

$$f(r) = r^2 \log(r) \qquad \text{thin plate spline function} \tag{3.2d}$$

$$f(r) = \sqrt{r^2 + \sigma^2} \qquad \text{multi-quadratic function} \tag{3.2e}$$

$$f(r) = \frac{1}{\sqrt{r^2 + \sigma^2}} \qquad \text{inverse multi-quadratic function} \tag{3.2f}$$

$$f(r) = \log(r^2 + \sigma^2) \qquad \text{shifted logarithm function} \tag{3.2g}$$

Among those, the most widely used is the Gaussian function, jointly with the Euclidean norm. This is a unique function, as the multivariate Gaussian function can be written as a product of univariate Gaussian functions:

$$f_i(x) = e^{\frac{\|c_i - x\|_2^2}{2\sigma_i^2}} = \prod_{j=1}^{n} e^{\frac{(c_{ij} - x_i)^2}{2\sigma_i^2}} \tag{3.3}$$

where σ_i is the standard deviation of the function.
3.3 shows the activation function values of one neuron, with centre $C_i = [0, 0]$, and $\sigma_i = 1$ It is assumed that are 2 inputs, x_1 and x_2, both in the range $]-2, 2[$.

The output of a RBFNN can be expressed as:

$$y = \sum_{i=1}^{p} w_i f_i(\|c_i - x\|_2) \tag{3.4}$$

and given the linear combination made in the last layer of the neural network, the output

will be given by:

$$y(x) = b + \sum_{i=1}^{n} \alpha_i e^{-\frac{\|c_i - x\|_2^2}{2\sigma_i^2}} \tag{3.5}$$

Where $\alpha_i^{n=1}$ is the network linear weights, n is the neurons number and b the threshold term. This term enables a constant change on the network output, i.e., this constant increases or decreases the network result.

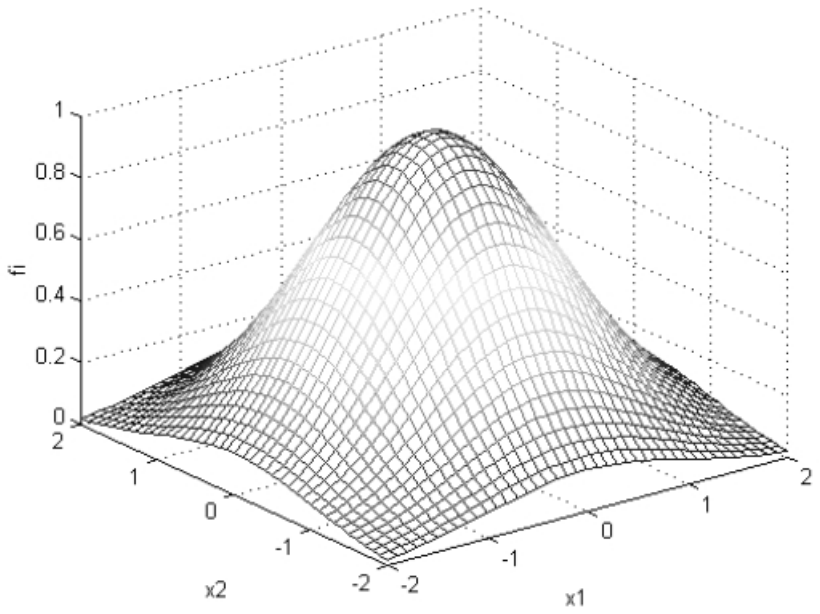

Figure 3.3: Activation function of one neuron centred in 0 limited by $]-2, 2[$ (taken from on [1]).

The Gaussian function presents some important characteristics, such as:

- The multi-variable version could be described as a product of uni-variable functions;
- Its derivative is also a Gaussian function;
- It is a localized function, the function tends to zero when the length to centre increases.

Figure 3.4 presents a simple example of a RBFNN mapping ($\mathbb{R} \to \mathbb{R}$), this network was created considering input/output discrete data as presented in the follow table:
In the figure, the black circles are the desired output and the red line the output generated by the artificial neural network. The other lines are all the radial basis functions. Considering (3.5), we can solve this problem with ANNs.
So with four centres, $n = 4$, which values are $c = [3, 7, 11, 13]$, with the linear weights, $\alpha = [5, 4, 3, 7]$, and the standard deviation characterized by $\sigma = [1.4, 1.3, 0.8, 1.3]$, the

Table 3.1: Input and Output values of the example.

Input	1	3	5	7	8	11	13	15
Output	2	5	3	4	2	5	7	4

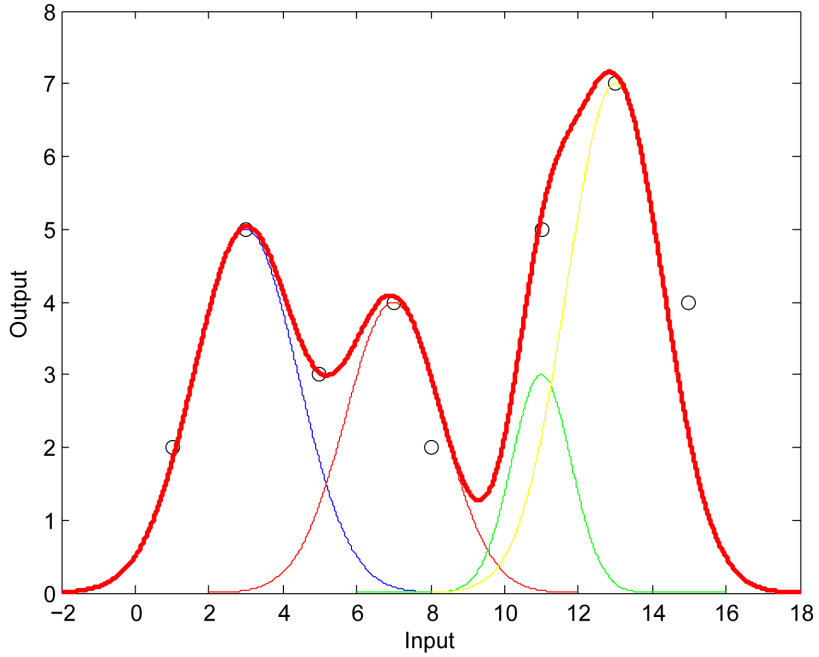

Figure 3.4: Example of a simple problem solved with ANNs.

mathematical expression that gives this mapping is described below:

$$y(x) = \sum \alpha_i e^{-\frac{|c_i - x|^2}{2\sigma_i^2}} = 5e^{-\frac{|3-x|^2}{2\times 1.4^2}} + 4e^{-\frac{|7-x|^2}{2\times 1.3^2}} + 3e^{-\frac{|11-x|^2}{2\times 0.8^2}} + 7e^{-\frac{|13-x|^2}{2\times 1.3^2}} \tag{3.6}$$

Note that the threshold parameter b, for this example, is equal to zero.

In this example the parameters were found manually to demonstrate how the RBFNNs work on their data sets. In complicated cases the neural network training is more complex, because the size of the sets increase in complex cases, like the real ones. So in real cases there is a need to have automatic methods.

As well as the MLP, the RBFNNs have also the universal approximation propriety [30]. This means that the neural networks could approximate any continuous mapping, if the neural networks have enough hidden neurons, if this example had as many centres as input patterns, the results obtained with the RBFNN could be equal to the desired output. The

artificial neural networks can achieve a great generalization of the training data, especially the RBFNNs, as this type of neural networks has a best approximation property [31]. Unlike MLPs, with the RBFNNs it is possible to change the position of the base functions without changing the other zones in the space, allowing an efficient and independent use of the neurons. The disadvantage of this property is that the training is slower and can be stuck in local minima [32].

3.4. Training Schemes

Initially when this network type as introduced, in 1988 [29], the training procedure was manual, so it was necessary to place a base function in every data point, the RBFNN behaved like a interpolator, forcing the mapping to pass for every points in the training data. This strategy translates into a raise of complexity, with the increase of the data length, usually originating badly conditioning networks. To solve this problem, this type of networks begun being used as approximators instead of interpolators, i.e., the functions base centres do not match the data points and are fewer in number.

The simplest way to build a RBFNN as an approximator is by selecting the centres randomly. As an alternative, they could also be randomly selected from the training points. The standard deviation, in this method, is usually given by [33]:

$$\sigma = \frac{d_{max}}{\sqrt{2n}} \tag{3.7}$$

where d_{max} is the Euclidean maximum distance between the centres and n is the number of centres, in the $\mathbb{R}^d$ space. This equation ensures that no base function will be too pointed or too flat, as this extreme condition should be avoided. The linear weights ($u = [a1, ..., an, b]$) could be found by:

$$\hat{u} = \Phi^{+} t \tag{3.8}$$

where Φ^{+} is the pseudo-inverse matrix of the hidden neurons, with dimension $P \times n$, being P the number of training patterns. The t vector is the desired output of the network, with $P \times 1$ dimension.

The problem with the previous training method described is that it could be necessary a large train set and several random initializations to get the desired performance. As an alternative, a two step method could be used; the second step should determine the linear parameters, which can be found by applying the linear minimum square strategy. The most popular method for grouping the data is the K-means clustering algorithm [34]. This algorithm iterates until there are no significant alterations of the centres. It is a special case of competitive learning process, designated as self-organizing map. The problem with the K-means clustering algorithm is that it only can get to a local optimum solution that depends on the centres initialization values. To solve this limitation in [35] was proposed an algorithm based on a measure that weights a variation on a group. This allows the K-mean clustering algorithm to converge to an optimum result, or close to it, without depending on initial centre values. This improved algorithm is designated by OAKM.

The standard deviations could be calculated by using Equation 3.7 or using strategies, like:

- **k-means neighbours heuristic:**

$$\sigma_i = \frac{\sum_{j=1}^{k} \|c_i - cj\|_2}{k\sqrt{2}} \tag{3.9}$$

  where k is the number of neighbours near the centre c_i in consideration;

- **empirical standard deviation:**

$$\sigma_i = \sum_{i=1}^{l} \sqrt{\frac{\|c_i - x_j\|_2^2}{n}} \tag{3.10}$$

  where n is the train pattern number associated to the i group;

- **maximum distance between patterns:**

$$\sigma_i = \frac{\max_{i,j=1\ldots p} \|x_i - x_j\|}{2\sqrt{2}} \tag{3.11}$$

  where p is the train pattern number.

3.5. Supervised Selection of the RBFNN Parameters

In this learning scheme, all the network parameters (c_i, σ_i and u) are free parameters encountered with a supervised learning process. This process could be the BP algorithm type [36].

The BP algorithm implements the steepest descent method and uses a rule for parameters actualization like it is showed below:

$$w[k+1] = w[k] - \eta g[k] \tag{3.12}$$

where w is the vector that correspond to the model parameters, n is the learning rate and g is the gradient vector. Usually, the error criterion applied is based on the square error sum:

$$\Omega(w,x) = \sum_{i=1}^{P} \frac{(t_i - y(w,x_i))^2}{2} = \frac{\|t - y(w)\|_2^2}{2} = \frac{\|e\|_2^2}{2} \tag{3.13}$$

where t is the desired output, y is the obtained output, P is the train pattern number, x_i is input vector with i index and e is the error vector. The learning rate choice is not simple, so there are the following alternatives to BP:

- **Conjugate gradient**, is an approach similar to the steepest descent, but includes an inertia term that is calculated in each iteration to produce conjugate search directions, the parameter actualization rule to this approach is given by:

  $$w[k+1] = w[k] - \eta g[k] + \beta[k]\Delta w[k-1] \tag{3.14}$$

  where $\Delta w[k-1] = w[k] - w[k-1]$, the $\beta[k]$ parameter [37] is usually calculated by:

  $$\beta[k] = \frac{g^T[k]g[k]}{g^T[k-1]g[k-1]} \tag{3.15}$$

  the other parameters are equal to the ones presented before;

- **Quasi-Newton**, this method types uses an approximation of the inverse of the Hessian matrix (H), that is the second derivative inverse matrix of (Ω). The actualization rule of this method type is:

  $$w[k+1] = w[k] - \eta[k]H[k]g[k] \tag{3.16}$$

  The approximation of $H[k]$ usually used is the Broyden, Fletcher, Goldfarb and Shanno (BFGS);

- **Levenberg-Marquardt**, this method uses the Gauss-Newton method to approximate the Hessian matrix (G). The gradient is given by:

  $$g(w) = \nabla\Omega(w) = J^T(w)e(w) \tag{3.17}$$

where ∇ is the gradient operator $\nabla\Omega(w) = \left[\frac{\partial\Omega(w)}{\partial w_1}, ..., \frac{\partial\Omega(w)}{\partial w_n}\right]^T$, being n the parameter number and J the Jacobian matrix.

This Jacobian matrix is given by:

$$J(w) = \begin{pmatrix} \frac{\partial y_1}{\partial w_1} & \cdots & \frac{\partial y_1}{\partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_p}{\partial w_1} & \cdots & \frac{\partial y_p}{\partial w_n} \end{pmatrix} \tag{3.18}$$

where y_i is the output for the i pattern, p is the number of pattern and n the number of network. The Hessian matrix is given by:

$$G(w) = \nabla^2\Omega(w) = J(w)^T J(w) + S(w) \tag{3.19}$$

where ∇^2 is the Laplacian operator and:

$$S(w) = \sum_{i=1}^{P} e(w)\nabla^2 e_i(w) \tag{3.20}$$

Assuming that $S(w)$ is small, the Hessian matrix is approximated by:

$$G \approx J(w)^T J(w) \tag{3.21}$$

The LM actualization for k iteration is given by the solution of:

$$(J(w[k])^T J(w[k]))\delta w[k] = -J(w[k])e(w[k]) \tag{3.22}$$

where I is the identity matrix and $v[k]$ a regularization parameter. Depends on $v[k]$ parameter, this algorithm can assure the convergence to a local minimum. The reduction provided criterion for this approach is:

$$\delta\Omega^p(w[k]) = \Omega(w[k]) - \frac{(e^p(w[k]))^{T(e^p(w[k]))}}{2} \tag{3.23}$$

being:

$$e^P(w[k]) = e(w[k]) - J(w[k])\delta w[k] \tag{3.24}$$

the effective reduction criterion is:

$$\delta\Omega(w[k]) = \Omega(w[k]) - \Omega(w[k] + \delta w[k]) \tag{3.25}$$

The ratio between the two $r[k] = \frac{\delta\Omega(w[k])}{\delta\Omega^p(w[k])}$, is used to actualize the regulation parameter (v) that usually uses the following rule:

$$v[k+1] = \begin{cases} \frac{v[k]}{2}, & r[k] > \frac{3}{4} \\ 4v[k], & r[k] < \frac{1}{4} \\ v[k], & \text{cc} \end{cases} \tag{3.26}$$

for negative values of $r[k]$ just the regularization parameter is actualized.

Associated with the descendent gradient algorithms there could be used stopping criteria to avoid the network poor conditioning. Bellow two stopping criteria are described:

- **Training error based**, the simplest way to stop the training is to pre-determine the iterations number to adapt the parameters. However, when the limit is reached the algorithm stops and there is not any assurance that the parameters are near the optimum ones. So, the most common criterion [37] is:

$$\Omega[k-1] - \Omega[k] < \tau_f(1 + \Omega[k]) \tag{3.27}$$

$$\|w[k-1] - w[k]\| < \sqrt{\tau}_f(1 + \|w[k]\|) \tag{3.28}$$

$$\|g[k]\| < \sqrt[3]{\tau_f}(1 + |\Omega[k]|) \tag{3.29}$$

  where τ_f is a measure of the desired corrected digits number, in the training criterion. The train ends when the three conditions are verified;

- **generalization based**, the previous criterion ensures that the network are well trained for the training set, however, it does not use information of how the network will behave when new data is submitted. This criterion assures that the network is well conditioned, i.e., the network has a great generalization capacity. As showed in Figure 3.5 against to new data the error decreases to a certain point and increases after that; it is in this point that the training should be stopped. This method is designated by early-stopping. This method needs two distinguished data sets, a training and a test set, which will be used to establish this criterion.

Some mentioned techniques could be combined to improve the network training. In this project the OAKM and LM algorithms are used with the early stopping. Notice that the

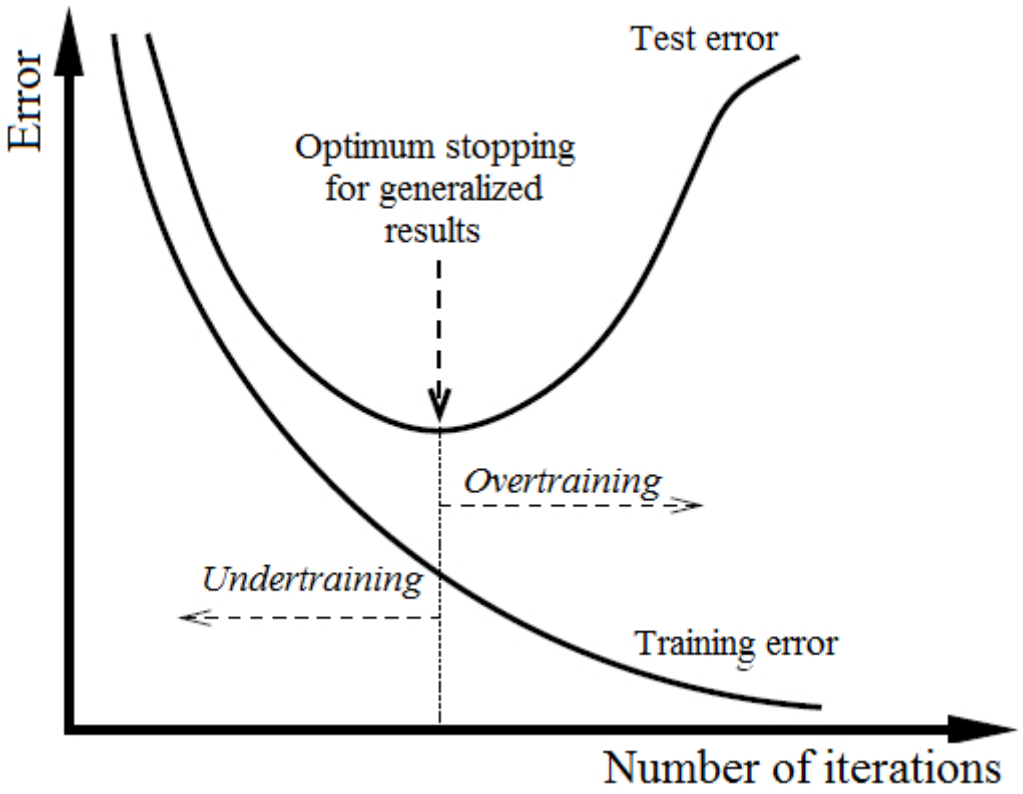

Figure 3.5: Generalized stopping criterion (image based on http://www.scielo.br/).

RBFNN have two kinds of parameters that should be found, the non-linear and the linear one. The network output should be given by:

$$y = \Phi(v)u \tag{3.30}$$

where Φ is the output matrix of the hidden neurons, with a column of ones for the threshold term. Considering the equation (3.13) error criterion:

$$\Omega(v,u) = \frac{\|t-y\|_2^2}{2} = \frac{\|t-\Phi(v)u\|_2^2}{2} \tag{3.31}$$

For any v value, the optimum value of Ω in relation to u could be found using the least square method, that can be determined using by the pseudo-inverse of Φ, $(\Phi(v^+))$:

$$\hat{u} = \Phi(v)^+ t \tag{3.32}$$

replacing (3.32) in (3.31), we get the following train criterion:

$$\psi(v) = \frac{\|t-\Phi(v)\Phi(v)^+ t\|_2^2}{2} \tag{3.33}$$

This criterion was proposed in [38], as is possible to verify, this criterion only depends on the non-linear parameters.

The application of the (3.33) criterion instead of (3.31) brings several advantages [37]:

- the problem dimensionality is reduced;
- when LM is used, each iteration is computationally cheaper;
- usually a lower number of iterations is necessary to converge to a local minimum.

3.6. Multi-Objective Genetics Algorithms

The concept of Genetic Algorithms (GA) became popular with Holland's and his partners work in the 1970s, before that many developments were made by this and other researchers. The firsts ideas that with we call today's evolutionary algorithms were implemented in the end of 1950s and beginning of 1960s decades. In 1962, many researchers had independently developed evolutionary inspired algorithms for learning purpose, but their study did not have many followers. After this studies the greater development in this area born in Technical University of Berlin with evolutionary strategy introduced by Ingo Rechenberg. The theory proposed by this German was too simple and different from the nowadays theory; briefly one individual generates another by mutation, the best of these two will evolve, again by mutation to another individual, and so on. Another development was introduced in America with the evolutionary programming concept; this concept differs from the other by using the individuals as finite-state machines. The most important development was made by John Holland, and the creation of this concept was attributed to him. In Holland's work the crossover and other operators were suggested. In his publication, [39], the GA was well defined and due to the dissertation published by Kenneth De Jong, in the same year, showing the potential of GA these concepts were accepted and developed.

This concept is based on the theory of origin of species introduced by Darwin; all the computational concepts in GA had its analogous in this Darwin biological thesis. In biology, a specie has individuals for several generations, each individual has characteristics, which changes with mutations and, more often, with crossover. In computational GA, a set of solutions, represented by population of a specie, have a number of individuals that suffer operations to improve its capabilities to solve the problem. As in biology the weak and

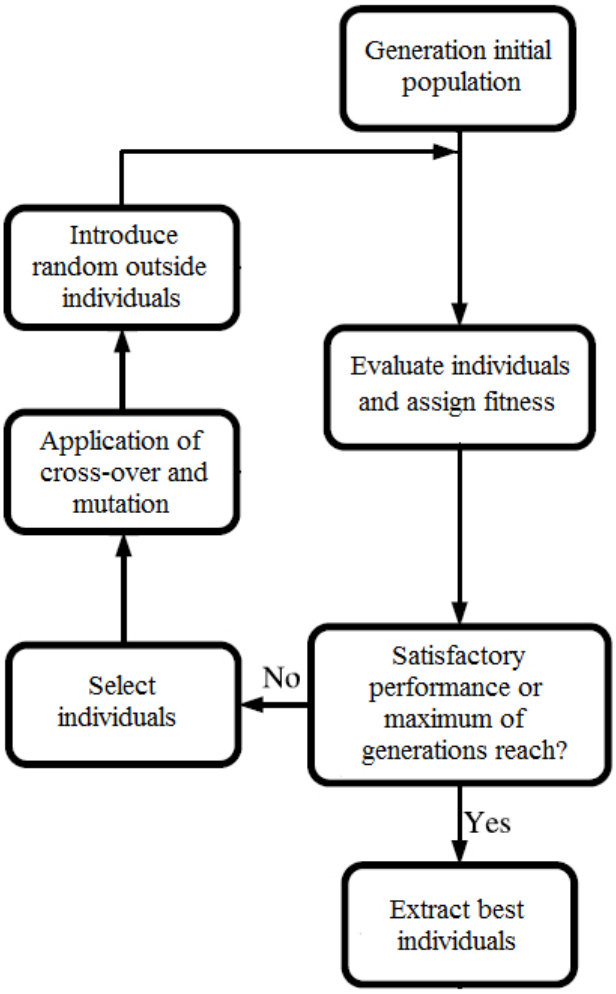

Figure 3.6: Scheme of GA steps (image based on [2]).

unfit die and do not produce a generation, in GA the worst result does not evolve. The GA starts to generate a solution set, called a population of individuals, this set is analysed and its results are compared, the best results are selected, the crossover and mutation operators are applied. Some individuals are randomly introduced in the population, represents the immigrants on a biological population; this operation serves to find new solutions. These procedures are in a constant loop until the generation reaches the maximum imposed or the performance reaches the expected value. To finish the algorithm, the best solution is extracted and used; this is graphically presented in Figure 3.6 to a better comprehension. With this method a global optimum solution can in principle be reached [40] without a full search, which is impossible for some problems. The advantages of GAs have already been used by computer vision researchers, examples of this are described in [41, 42].

In biology, each individual has a gene, corresponding to a discrete unit in GA. These genes

control the appearance of a creature, in GA each gene corresponds to a feature. In GA there are two levels of representation, the phenotype and genotype. An individual has a phenotype and a genotype; the phenotype shows the dominant chromosome allele characteristics. As an example an individual has brown hair because one of his chromosomes allele represents that characteristic, to have blond hair the two chromosomes alleles have to be equal and represent the blond hair characteristics. In GA the phenotype corresponds to the individual, and its chromosomes alleles, called genotype, corresponds to its genetic sequence [2]. An example of phenotype and genotype in a problem with a bit string is that the bits are the genotype and its result is the phenotype. Figure 3.7 shows this example.

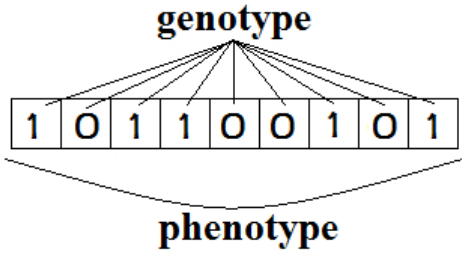

Figure 3.7: Labelled bit string.

In genetic algorithms the problem is defined by an objective function. To compare the individuals it is necessary to evaluate each individual and associate a cost function to its aptitudes. This cost depends on the problem and the individual aptitude is related to the selection process. The aptitude cost mapping should be monotonic in $\mathbb{R}^+$ and there are two ways to attribute the aptitude of each individual. The first takes into account all the population, each individual is ordered and attributed by its cost function value, the other converts the cost function in aptitude through a linear or a non-linear function.

The aptitude classification could lead to a stagnation problem when a 'super' individual appears in the population and generates individuals too similar to him. This individual differs of the others because it has a great aptitude value and dominates all the population generating a possible local minimum. To avoid this problem is necessary to use classifications strategies. Another tactic used to avoid the stagnation is the application of operators such as crossover and mutation. The mutation operator changes some characteristics of one parent as illustrated in Figure 3.8.

The crossover operator mixes the characteristics of two or more individuals to generate another one, Figure 3.9 demonstrates this operation.

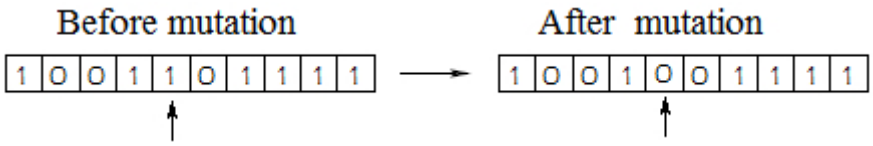

Figure 3.8: Application of mutation in a binary example string (image based on [2]).

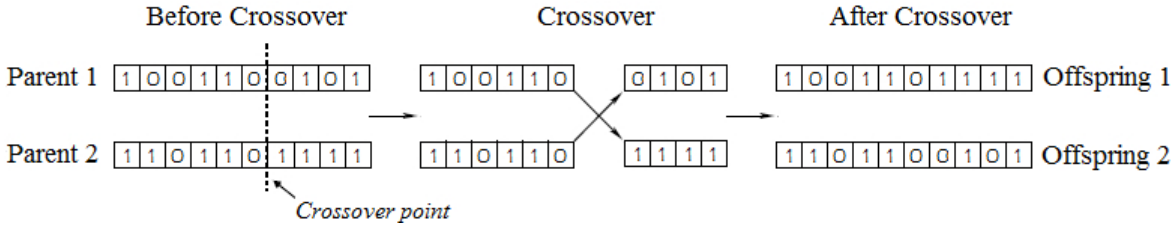

Figure 3.9: Crossover application in a binary example (image based on [2]).

After the individual's selection, genetics operators are used to modify each individual, to create a new generation, keeping the best properties of the most able individuals and, above all, avoiding search stagnation. These stochastic nature operators and their parameters, such as, the choice of the crossover and mutation rates and the alterations location or type, should be well defined, because the good performance of the algorithm could depend on these operators. A population of individuals grows due to these genetic operators. In some strategies the descendants, a new generation of individuals, have the opportunity to compete with the generation above.

In GA the selection process has a stochastic nature, i.e., the probability of an individual being selected to originate others depends on the aptitude value that it presents. To select the individual taking into account its aptitude, it is necessary to use one of the three known schemes of standard selection: Roulette Wheel Selection (RWS) proposed by Goldberg's [43], Stochastic Universal Sampling (SUS) [44] and Tournament Selection (TS) [45]. The first scheme mentioned, RWS, works like a spinning wheel divided in N parts where the size of each part corresponds to the aptitude value; when the wheel stops a random individual is selected. The SUS instead of single selection could be used to select multiple individuals; SUS uses N equally spaced pointers, where N is the number of selections required. The population is shuffled randomly and a single random number higher than zero and lower than the sum of aptitude, F, divided by N, is generated. This is the first individual selected, the next selections are the subsequent F/N individuals.

The last scheme, TS, uses the comparison between two individuals and discards the one with lower aptitude. The TS scheme could discard a good individual if this competes with a better one and include a bad one if it competes with a worst, leading to selection errors. The RWS was not create for multiple selections and if it used for that could lead to selection errors too. For these reasons, and due to it zero bias, the SUS is the most widely used selection scheme in GAs [46].

3.7. Multi-Objective Optimization using Genetic Algorithms

The Pareto-ranking is one of several methods for adapting GAs to optimize the problem over the number of dimensions [47]. For the majority of the problems more than one objective needs to be reached simultaneously, but usually improving one objective deteriorates the others, the so called conflicting objectives. In these cases, the problem presents a set of optimum solution instead of just one; this set is designated by Optimal-Pareto or non-dominated individuals [48].This individuals are the ones that have the best solution given all objectives. Figure 3.10 illustrates this concept, and shows that the main objective of a multi-objective optimizer is to approximate the error results a Pareto frontier.

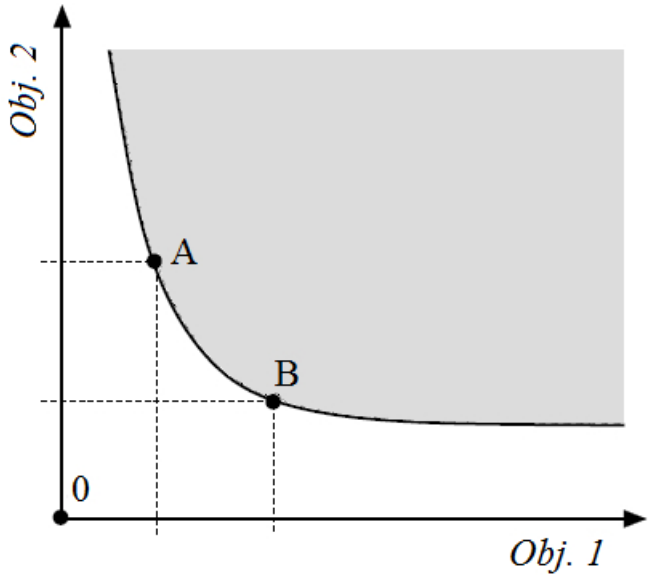

Figure 3.10: Pareto frontier example (image based on [2]).

In Figure 3.10 we can see a continuous line corresponding to Pareto frontier, the outside of

this line, lighter part, represents the possible non-dominated solutions; the line represents the found non-dominated solutions and all the solutions found or that will be found inside the line are all the dominated solutions. This representation is only possible if the number of objectives is lower or equal than three, and with three objectives a 3-dimensional plot is necessary. So for these cases the individual objectives are usually represented as shown in Figure 3.11. In this case an integer number represents an objective and a line is an individual with the respective values for each objective.

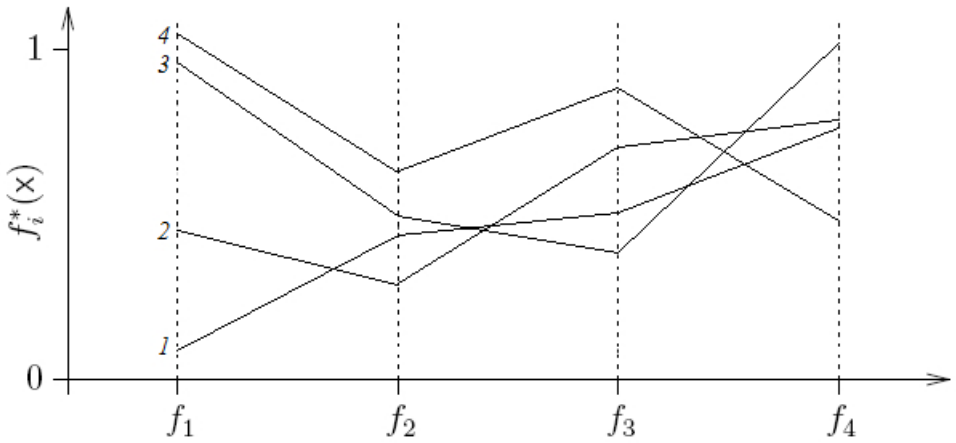

Figure 3.11: Method to observe the MOGA results when the objectives number is higher than three (image based on [2]).

In this representation the dominated individuals are the higher ones that do not cross at least one of the others. In Figure 3.11 if only the first three objectives are considered the third individual is dominated by the fourth (if it is a maximization problem).

As in multi-objective optimization a method which evaluates a set of solutions on a parallel search is appropriate [2], the GA provides results for each individual in the space defined by the objectives. The standard GA proposed by Goldberg in 1989 [43] could resolve multi-objective problems by extrapolation. In these cases the cost is more difficult to attribute due to the several cost description in multi-objective problems; these descriptors should be combined to generate a unique cost attribution. To generate this cost we could use several methods, for example a simple way is the balanced sum - this technique associates a weight to each objective defining a relative importance between its. This technique presents some flaws:

- a wrong chosen weights could be generated by a wrong search;
- in most of the real applications, only after the search it is possible to find the weights values, that requires several executions of the GA, to find the appropriate weights;
- little variations on the weights could take to big variations on the objectives vectors, and the opposite is valid too.

This approach does not evaluate non-convex regions of the search space as we referenced in [37, 2]. The scheme implemented in MOGA by Fonseca and Fleming [49] is used on this work to optimize the RBFNNs structure. In this strategy the Pareto dominance could be used to attribute individual costs and attribute automatically aptitude values through classification. This classification enables the evaluation of convex and non-convex regions and each individual is classified according to its number on the domain. Figure 3.12 illustrates the individual's classification. In Figure 3.12 an example of individual classification where the numbers above the circumferences are the number of better solutions found for any objective, the zero corresponds to the non-dominated individuals.

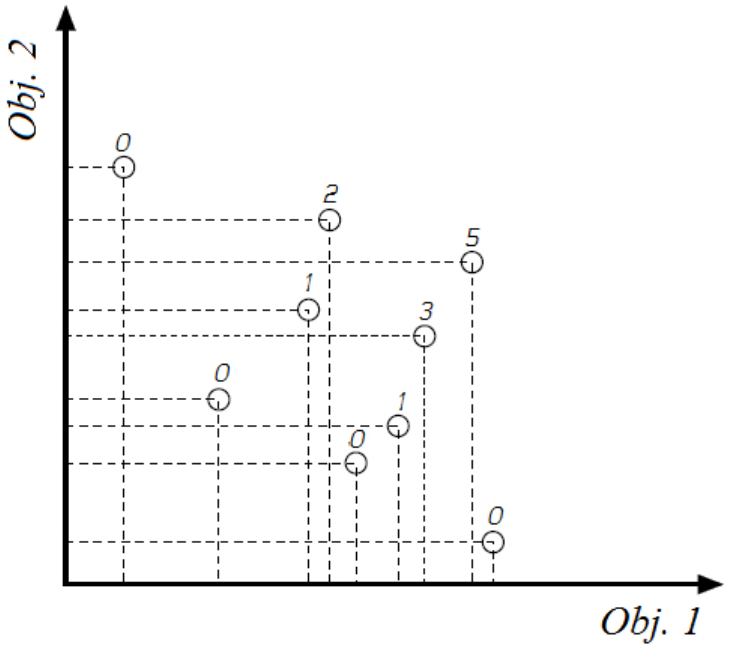

Figure 3.12: Classification example of MOGA individuals (image based on [2]).

3.8. Neural Networks Models Determination

The search for a neural structure is not an easy or exact task. To get the best neural

network structure it is necessary to make a full search for all the possible structures. This is impossible due to the number of structures combinations. So MOGA is used for this purpose. Using the genetic approach the search could achieve a result close to optimal. The MOGA mechanism was explained in the last section, in this one we will explain what the input parameters are and how these parameters are found.

The parameters that are necessary to configure in each MOGA run to get forecast neural structures are:

- Maximum number of times to run genetic algorithm;
- Maximum number of generations;
- Maximum number of inputs to NNs;
- Maximum number of NN neurons;
- Minimum number of NN neurons;
- Prediction horizon for NNs;
- Size of regression window;
- Maximum lag for NN inputs;
- Number of lag terms.

Usually some of these are not really used. For example, the maximum number of generations is not really applied because the search will be manually stopped when it appears to achieve the optimal result. In the studied cases the maximum number of times to run the genetic algorithm was only one. This option could be used to make the results more credible, but this delays the search.

To choose the value for maximum number of NNs inputs and hidden layer neurons it is necessary evaluate the problem. The complexity will be greater if we increase this option, also the waiting time will be longer too. If the computational resources are limited the most indicated value is the one that computational resources could support. If the computational resources are not a problem, to achieve a consistent value, there is a need to find a limit of neuron numbers which will deteriorate the results. To assure that the number of neurons was not too low the Pareto frontier should not have ANNs structures with neurons number equal to the limit imposed, or at least not so many. If the maximum number of NN neurons is set too high, the waiting time will be longer and wasted.

The minimum number of NNs neurons in the hidden layer is connected to the compu-

tational resources too, as networks with small number of neurons are faster but not so accurate. The GA was improved to find complex neural networks, so, for some small networks the GA cannot present a proper result.

The prediction horizon is the number of steps ahead to forecast. This value is easily calculated, by assuming a forecast time, a sampling interval, and dividing the former by the latter, in the same units.

The regression window size and maximum lag for NN inputs are the same. These two options exists to improve this algorithm which is constantly in development. The value to use depends on the problem; in the studied case 24 hours of regression were used because the temperature and radiation have daily patterns.

Finally the number of lag terms is the length of the regression vector. To calculate this value we need to sum the lengths of all regression variables. In the studied case it was used a lag of 24 hours of auto regression with the next shape: the first 8 hours with a 5 minutes interval, the next 8 hours with 10 minutes interval, and the last 8 hours with 15 minutes interval, in a total of 176 lags. To these values is necessary to sum the length of the exogenous variables, 48 for global solar radiation and 72 for temperature forecasts. To get these values, usually a simple correlation is needed; to demonstrate how it was made the temperature example will be presented.

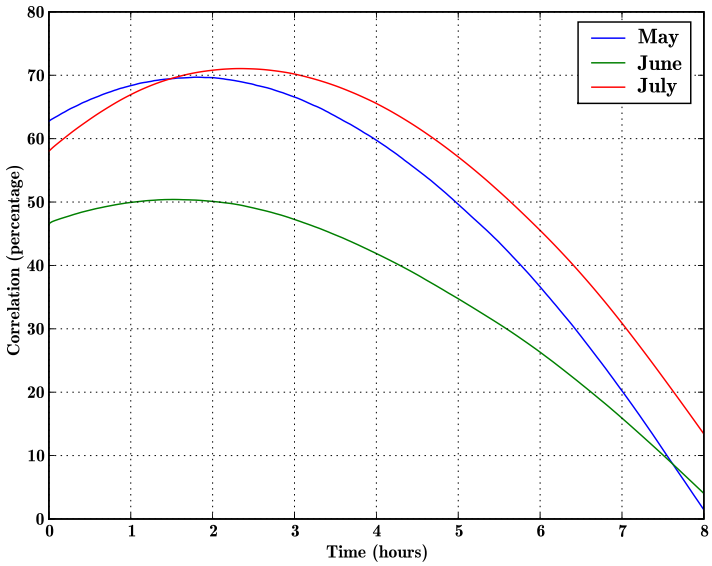

Figure 3.13: Correlation of three months between radiation and temperature.

First, it is necessary to have uninterrupted data for the variables in use (in this example

global solar radiation and temperature). After that, a step-by-step correlation is applied and the results need to be analysed to decide the longest lag term possible. In the example Figure 3.13 shows the correlation results. To make a step-by-step correlation it is necessary to fix one of the variables and within a loop to increase the lags of the other using a number of steps larger than the expected. In this example lags of one minute during 8 hours were applied. Analysing these results and taking into account that the July/August months are the hottest in Portugal, and the ones which have more lags in temperature/radiation correlation, we can decide that using a lag of 6 hours covers the extreme cases. In the plot it is possible to verify that, in July of 2009, the instant and 5-hours-ahead correlation between atmospheric temperature and global solar radiation is similar, so to include unusual hot months and places, the 6 hours horizon was chosen.

4. Cloudiness Index Determination

The problem of cloudiness index determination from the analysis of images is a complex one because the measurement of the percentage of clouds in the sky image introduces various problems. The biggest one is the absence of a reference index for each image, as supervised methods will require. Other problems arise from the images themselves and from the image acquisition system. Because clouds have different densities, thinner clouds may easily be considered sky or cloud. Another problem is that with a GBAS imager the bright area around the sun may cause a blind spot in the image, demonstrate in the bottom figure 4.2, which misleads the classification. Also, the image acquisition system needs to be clean at all times in order to capture high quality images, otherwise the segmentation methods will produce incorrect results due to artifacts in the images. At last, if the device has a shadow band, it needs to be always in the correct place or the sun will spoil the image.

To solve the problem, first a set of images was analysed, for which a mask including all the cloud pixels was arbitrarily specified according to the author personal judgement. This mask was used as a reference to compute the classification error for the different segmentation methods considered. The second step consisted in analysing distinct colour channels, or transformations of them, because different ones have important properties. In the third step, some statistical measures were analysed in order to classify some of the extreme cases, like totally cloudy and totally clear sky images. The final step was to evaluate the segmentation methods, after all transformations and prior classifications, to find the best threshold that separates clouds and clear sky in the images.

This chapter starts by presenting the data sets employed and the colour models studied. After that the thresholding methods used are presented, followed by the neural network method that is being proposed. The first method presented in section 4.4, adopted due to its simplicity, consists in using a fixed threshold for all the images. It is followed by the Ridler, Calvard and Trussel (RCT) [50, 51] and Otsu's [52] methods. At the end of the chapter, section 4.5 presents the results achieved by the methods mentioned, including two subsections devoted to the results achieved when using or not the prior statistical

classification.

4.1. Data Sets

The all-sky images used in the experimental work were acquired using a TSI-440A manufactured by *Yankee Environmental Systems, Inc.*, located on top of one building ($37^o02'N$, $07^o57'W$) in the University of Algarve, Faro, Portugal. The images are stored in RGB colour mode (8 bit/channel) using the JPEG file format and having the dimensions 704×576.

The images were selected from a series ranging from October 2004 to March 2005. In order to have a representative data set, the day was divided in five parts considering the clock time, and within each part approximately the same number of images were selected by another five intervals of the sky clouds percentage, the result of this selection is depicted in figure 4.1.

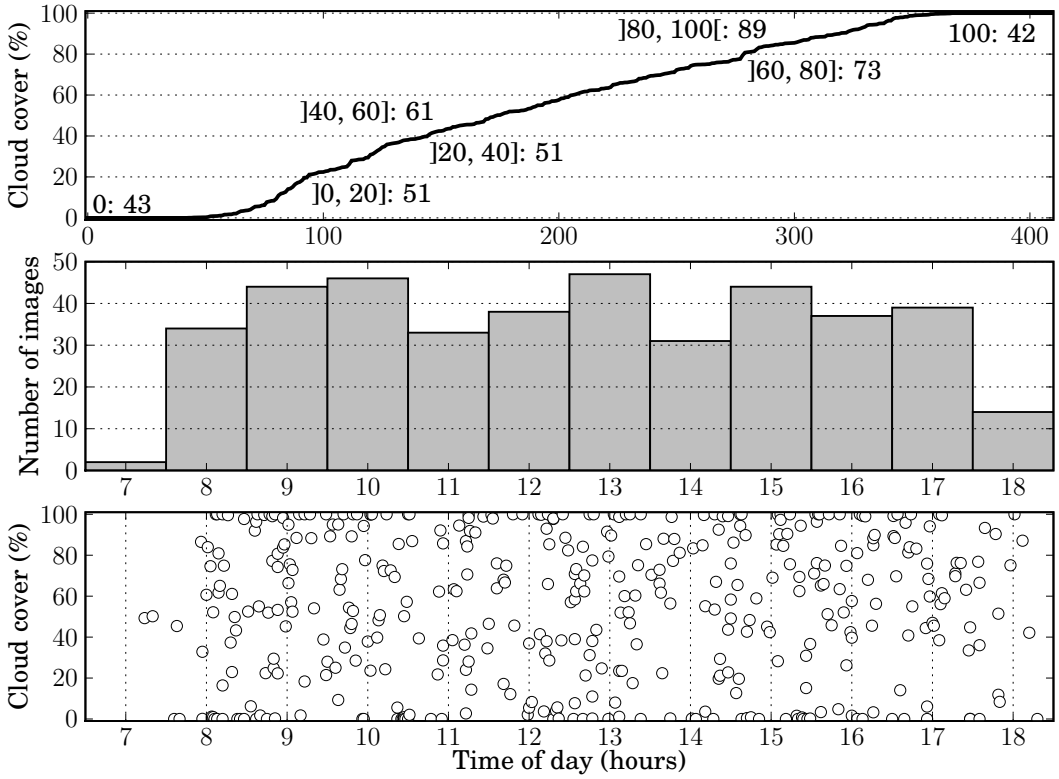

Figure 4.1: Details about the 410 images used. Top: number of images by ascending cloud cover intervals. Middle and bottom: respectively, the number of images and cloud cover distribution by the time of the day.

In order to make the selection, the images were first processed in order to specify the clear and cloudy parts of the sky. The method used to accomplish this consisted in manually marking all the clear sky pixels in grey, the pixels that represent clouds in white, and the remaining in black. Using these masks, a simple function can count the total of white pixels and divide by the sum of white and grey pixels, thus achieving the cloud fraction of the image.

At last, a total of 410 images were divided into three parts, the first one (290 images) to be used by supervised methods and to determine the statistical classification rules, the second one (60 images) to verify if the segmentation methods are generic enough and, possibly, to make some improvements, the last one (60 images) to perform final validations. The last data set was never used to make improvements to the methods.

Figure 4.2 presents two examples of the images that were analysed along with the masks specified. The pair of images at the top is a difficult one because it has thin clouds that are not well defined.

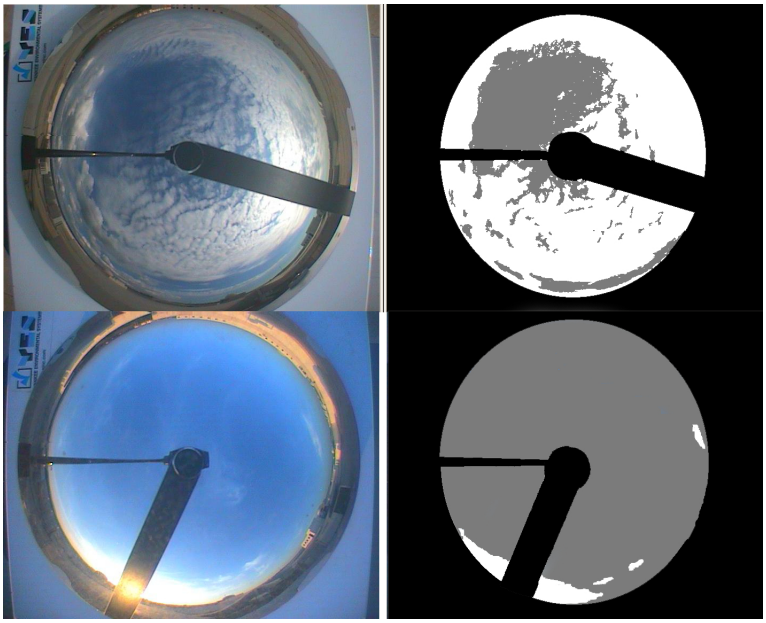

Figure 4.2: Image of sky, in left, and its cloud/sky decision.

For images that have a white spot around the sun, such as the one shown at the bottom of the figure 4.2, sets of images that were captured before and after that image were visually inspected to discover if there are clouds in that blind spot.

4.2. Colour Models

The study started by analysing the conventional RGB colour model, as it is the most common and the one used in the images. In this case, it was noticed that the red channel has more important information when compared to the remaining channels, blue and green. This is justified by the higher saturation of blue, and also green, in clear sky pixels. In all three channels, clouds have a larger saturation, so the red channel has the highest contrast between cloud and clear sky pixels.

Another known colour model is the Cyan, Magenta, Yellow and Key (CMYK), which uses subtractive instead of additive colours. In this model the last component is the saturation of black, the other three correspond to the primary colours. No benefits were found in the analysis of this colour model when compared to the RGB, so it was not used any further.

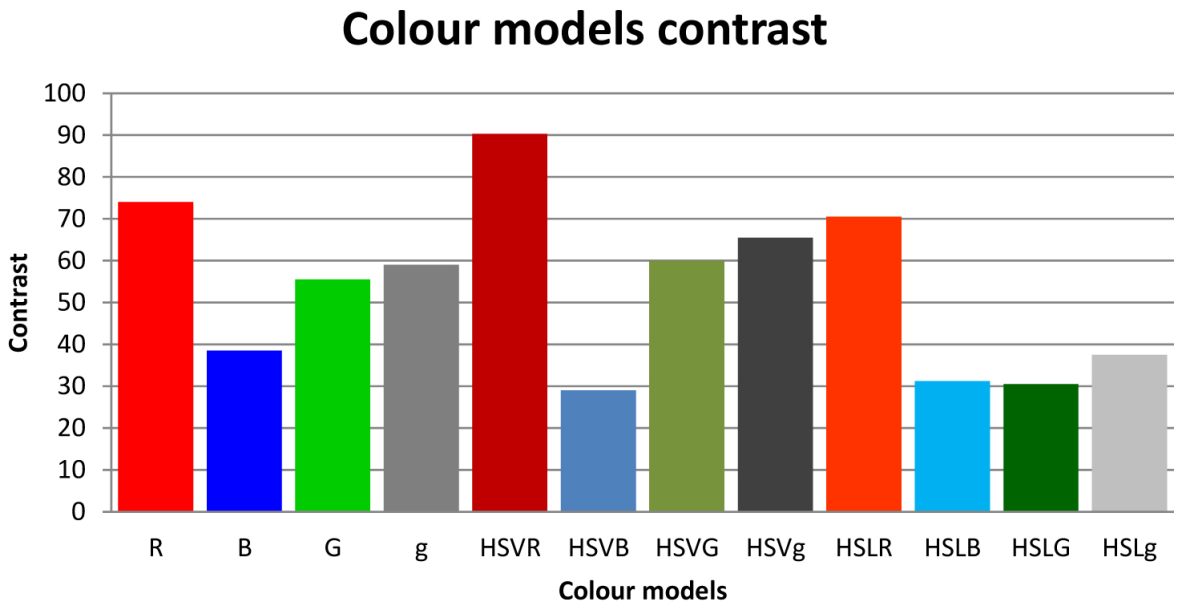

Figure 4.3: sky/clouds contrast for several studied pixel intensity scales.

Other important colour models are the Hue, Saturation, Value (HSV) and the Hue, Saturation, Lightness (HSL). These work directly with a model of the light present in the images, so a deeper analysis was conducted. The first one, HSV, uses the Value channel to store the "light" information, so, theoretically, if this channel is equalized we get a more efficient image in order to compare pixels far and closer to the sun and an increased contrast between cloud and clear sky pixels. In practice, because the Value channel does not have a real light, but a mathematical model of it, near the sun the channel saturates and a white spot appears. This white spot is present in all colour models constituting a

lack of information due to the strong sun's light and radiation, this is demonstrated in the bottom figure 4.2.

The HSL is another colour model that tries to separate light from the other colour components, but with a different mathematical model. This model was considered better for some images, although globally, provided lower contrast when compared to the HSV. So, after the analysis done, it was decided to include channels from the HSL and HSV models in the remaining of the work.

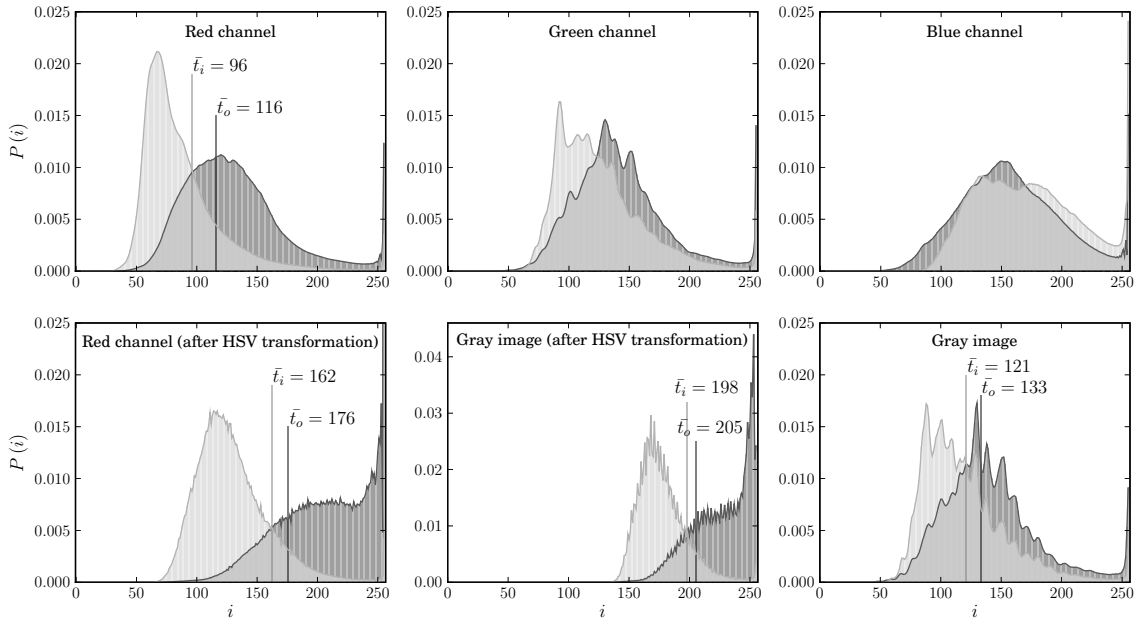

Figure 4.4: Histogram of clean sky (class S) and clouds (class C) classes for six different pixel intensity scales.

In an attempt to use the HSL, HSV and RBG model benefits, derived from the improved contrast in the red channel of the RGB model, the images were transformed into HSV and HSL, and the Value and Lightness channels were set to 1.0 and 0.5, the maximum and median respectively, in order to equalize the light without loss of information in the saturation and hue channels. After these changes, the images were transformed back into RGB colour model and the red channel was used due to the increased contrast achieved. Considering that the segmentation methods work on one pixel intensity scale, also gray-scale images were tested by generating them from the available RGB images (original, after HSV transformation, and after HSL transformation).

In order to assess the level of contrast between cloudy and clear sky pixels in the various pixel intensity scales, the average of the absolute difference between the sample means of

pixels in the cloud (C) and clear sky (S) classes was analysed.
As it may be seen in figure 4.3, the HSVR (Red channel after HSV transformation) has the higher contrast between C and S. This shows that the HSVR pixel intensity scale is probably the most important and should be used by the segmentation methods. This conclusion is also corroborated by the plots presented in figure 4.4 where the probability mass functions of the C and S classes are shown for some of the pixel intensity scales.

4.3. Statistical Classification

An attempt was made to perform a statistical classification in order to correctly classify images that have only pixels from one of the classes, C or S, as some segmentation algorithms tested assume that the images have two classes and will always segment the input into two classes. Clearly, if an image has only one class, these algorithms will fail and deteriorate the results.
To perform the statistical classification the first set of images, mentioned in section 4.1, was statistically studied. Knowing that the white colour in the RGB colour model corresponds to the value 255, and that clouds after the HSV transformation correspond to white or near white values, it becomes possible to classify images having a high average on the considered pixel intensity scale. As will be shown, this method correctly classifies the majority of images that only have the C class.
The classification of clear sky images using a similar procedure is more difficult. Theoretically, images with a low average on HSVR could be classified as clear sky, but as for many of these images a white spot is present near the sun position, their sample mean resembles that of images having both clear and cloudy sky. This means that clear sky images can not be correctly classified only by looking at the average pixel intensity. To solve this problem, the standard deviation and the skewness, which are the second and the third statistical moment, respectively, were studied. In theory, the standard deviation should have a small value as clear sky images have the pixel intensities well concentrated around an average intensity, but again due to the sun white spot this behaviour is distorted and a correct classification can not be achieved. The same is true for the skewness, that, for a set of ideal images, could get the cloudiness index with a small error, because it shows if the histogram has a higher density before or after the sample mean.

By considering more than one statistical moment, if a higher mean and a positive skew were considered, some clear sky images are correctly classified, but not all. If the skewness and the standard deviation were used, more images of this type could be correctly classified. Finally, using the three statistical moments it becomes possible to have an improved separability between clear sky images and those that have a small amount of clouds.

The results obtained by the statistical classification are presented in figure 4.5. Although the three statistical moments were employed, as explained above, in the plot only the sample mean of HSVR and the cloud percentage are shown.

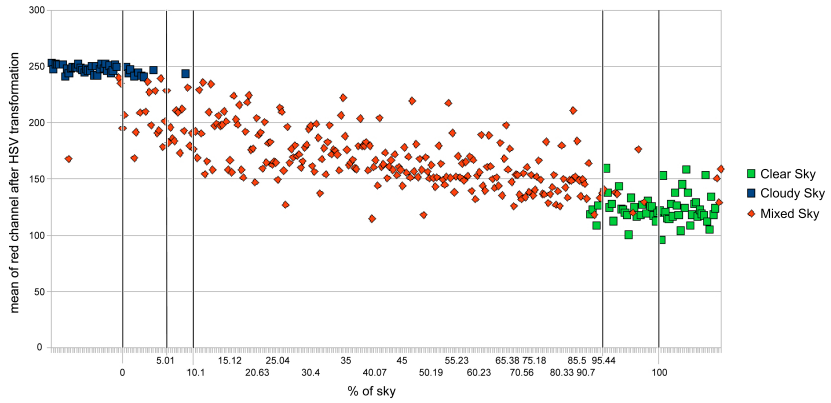

Figure 4.5: Statistical classification results.

As it may be seen, considering totally cloudy sky, only one image is misclassified with a cloud fraction error around 10%. For the clear sky case, around 60% of images were classified with an error lower than 1%, 95% with an error lower than 5%, and all the images with an error lower than 10%.

4.4. Segmentation Methods

4.4.1 Fix Threshold

Considering that an approach with a neural network classifier is computationally expensive, other simpler and not so computationally expensive methods were studied.
The histogram based analysis from which figure 4.4 was produced, helped to identify a single threshold that provides the best cloud fraction estimation error for each of the pixel intensity scales. Obviously, the best fixed threshold should be found around the Main

Intersection Point (MIP) of both classes. One of the vertical lines in each plot presents the average optimum threshold (minimising the cloud fraction error estimation) threshold, $\overline{t_o}$, found over all images. This point is hardly the best threshold, in this case, but it is shown for comparison purposes. Figure 4.6 presents the average absolute cloud cover estimation error as a function of a range of thresholds around the MIP, along with $\overline{t_o}$, for the most favourable pixel intensity scales.

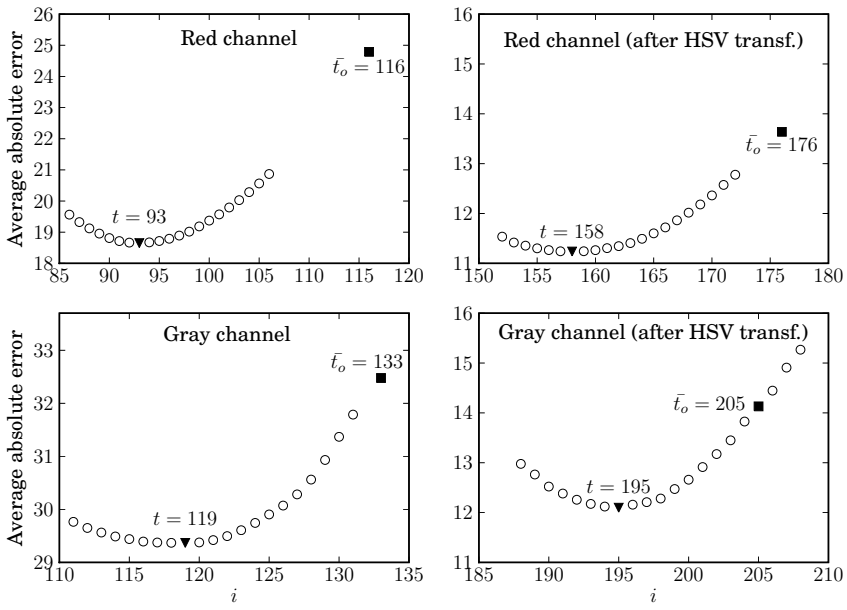

Figure 4.6: Fixed thresholds performance.

The dark triangles depict the best error performance and show the corresponding threshold, the dark squares denote the error performance obtained by $\overline{t_o}$, whose threshold value is also shown. The threshold for which the best error was achieved is also shown in figure 4.4 by a vertical line and label $\overline{t_i}$. As expected from the analysis of figure 4.4, the best results were found on the HSVR channel, with $t = 158$, for which the average absolute cloud cover estimation error was 11.24. The minimum error obtained was 0.0 and the maximum 82.64. This performance serves as a baseline for comparison, as any other more complex method should achieve better error figures to be considered useful in this problem.

4.4.2 RCT Method

This method was proposed by Trussel in 1979 [50], using an iterative threshold algorithm proposed by Ridler and Calvard [51] in 1978. In this dissertation this method will be

denoted as RCT. As initially proposed the method starts by selecting a random threshold from the pixel intensity range within the image. In this work the threshold initial value was set to the average image pixel intensity, as this is more usual nowadays. This threshold generates the two initial classes assigning to one the pixels below or equal to the threshold, and to the other the pixels above it. In subsequent iterations a new threshold is estimated as the average of the sample mean of the two classes. Finally, when the estimated threshold does not change from one iteration to the next, the algorithm stops. The threshold, $\hat{t}'$, is estimated at each iteration by means of the following equation:

$$\hat{t}' = \frac{1}{2}(\mu_S + \mu_C) \tag{4.1}$$

where μ_S and μ_C are the sample means of the classes generated by the threshold estimated in the previous iteration.

4.4.3 Otsu's Method

This method is commonly used in computer vision and image processing applications in order to automatically separate the pixels of one image into two classes. As for the RCT method, this algorithm assumes the existence of two classes in the input. It was proposed in 1979 by Otsu [52], for further information [53] may be consulted. The idea implemented by the algorithm is quite simple: search exhaustively for the intensity value that divides the image in two classes having the lower intra-class variance between them. It was also shown by Otsu that this is equivalent to maximise the inter-class variance, as expressed in the equation,

$$\sigma_B^2(t) = \omega_{\mathbf{S}}(t)\,\omega_{\mathbf{C}}(t)\left(\mu_{\mathbf{C}}(t) - \mu_{\mathbf{S}}(t)\right)^2 , \tag{4.2}$$

where $w_S(t)$ and $w_C(t)$ are the probabilities of the classes S and C being separated by the intensity t, and $\mu_S(t)$ and $\mu_C(t)$ are the two classes sample means.
The implementation is very simple as one needs only to cycle over the pixel intensities within the image and search for the value of t that maximises the equation above.

4.4.4 Neural Networks

4.4.4.1 Segmentation Approach

An attempt is made to identify a radial basis function neural network image thresholder to replace (if beneficial) the various methods already presented. As illustrated in figure 4.7 the output of the neural network is the threshold to be used in an image, the inputs are a certain set of features extracted from that image.

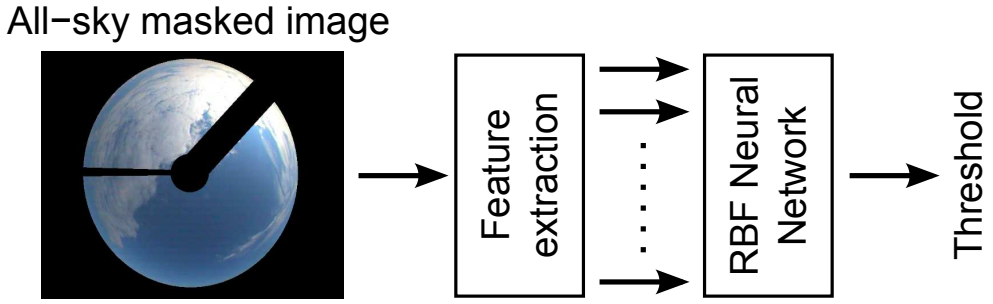

Figure 4.7: Neural network image thresholding approach.

The features were obtained from the intensity channels of the original RGB image, the HSV and HSL transformed images and their corresponding RGB versions, and finally from the gray-level images obtained from the RGB images. The various formats and transformations were described in section 4.2. A total of 17 distinct pixel intensity channels were considered from which the sample mean, standard deviation, and skewness were extracted. Additionally, from the histograms of channels with higher contrast between the two classes (reds and greys), the most frequent, first non-zero, and last non-zero intensity levels were also extracted, resulting in a total of 69 features.

Considering that the number of features is relatively large and that the number of neurons for the ANN also needs to be selected, the ANN input-output structure was selected using the MOGA as it was presented in section 3.3, or [54, 55]. Briefly, the MOGA starts with a randomly generated population of 100 individuals (ANN image thresholders) which are evaluated and ranked in order to prepare the next MOGA generation. Hopefully, after a sufficient number of generations a set of ANNs meeting pre-specified objectives are evolved by the MOGA.

Three sets of images, as explained in section 4.1, were used in order to train the networks, to test their generalisation ability, and for validation after the MOGA structure optimisation. Two objectives were specified for the MOGA: the RMS of the ANN output error over

the training and generalization data sets. As the ANN parameter initialisation is made randomly, to decrease the probability of one good ANN to be poorly evaluated because of a bad initialization, or vice-versa, 25 training trials were executed for each individual, and the average of both objectives was used for evaluation purposes. This number was chosen after a few experiments to assure that the number of trials was not too high, which would result in greater computational efforts, or to small, which would not favour a fair comparison among the individuals in one MOGA generation. The MOGA execution is stopped when for a reasonable number of generations no improved solutions are found according to the objectives specified. After completion of the MOGA, for the set of resulting image thresholders, a larger number of training trials was executed in order to select one ANN for further comparison with the previously described methods. In this case, the final selection considered also the results over the validation data set in addition to the results over the training and generalisation data sets.
The ANN training method used was the LM algorithm using a modified criterion exploiting the non-linear/linear topology of the RBFNN [56, 37]. The stopping criterion used was early stopping, in order to obtain good generalization ability.

4.5. Results

4.5.1 Automatic Thresholding Methods

4.5.1.1 After statistical classification

In this section the threshold estimation error and the associated error in the determination of the percentage of clouds in the sky, will be presented considering the statistical classification and the RCT method applied to the images that were not classified using the statistical method (mixed sky in figure 4.5).
The average absolute error in the estimation of the cloud cover is shown in Figure 4.8, considering the three data sets defined in section 4.1.
The algorithm proposed has excellent results in the first groups of images, good results in the second, and promising results in the third set. In figure 4.9 the associated threshold estimation error is presented. This error behaviour, which is not the primary objective,

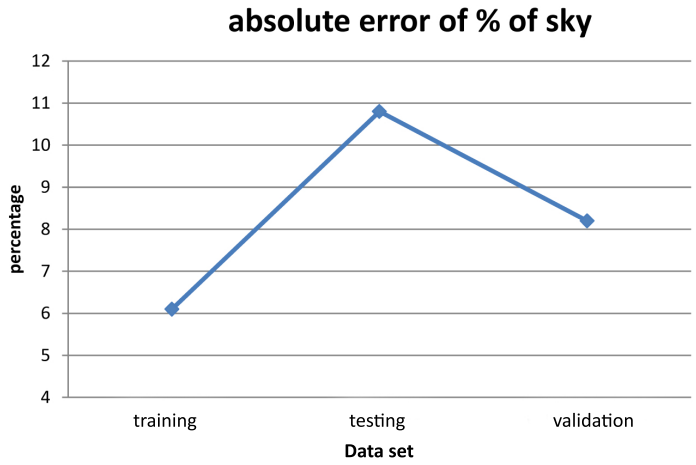

Figure 4.8: Results in percentage of clean sky error.

shows that an ideal threshold may not be necessary as for images having a high contrast between the classes there is a range of thresholds that will result in similar cloud cover estimation.

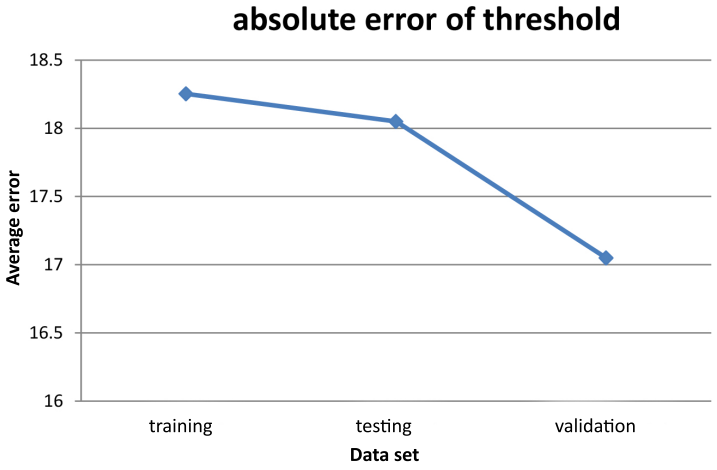

Figure 4.9: Results of threshold error.

As it may be seen from both figures, when comparing the results in different data sets, there is not a clear relation between threshold estimation and cloud cover estimation. This fact might be due, among other reasons, to the variable number of images that were statistically classified in the data sets.

4.5.1.2 Without statistical classification

As it was mentioned before, the segmentation methods already presented will always split the image pixels into two distinct classes, therefore a solution is required in order to improve the results when only one class exists. The results presented in this section and in 4.5.2.1 were published in an article [57] presented at the *International Federation of Automatic Control, Conference on Control Methodologies and Technology for Energy Efficiency* in 2010.

Table 4.1: Segmentation algorithms results in four channels.

Channel	RCT method			Otsu's method		
	min.	av.	max.	min.	av.	max.
R	0.24	31.27	90.07	0.39	31.54	89.45
HSVR	0.10	18.92	95.67	0.02	19.57	95.20
G	0.13	32.78	89.76	0.06	32.80	88.59
HSVg	0.23	18.89	95.70	0.00	20.84	95.05

Channel	FT method		
hsvR	0.00	11.24	82.64

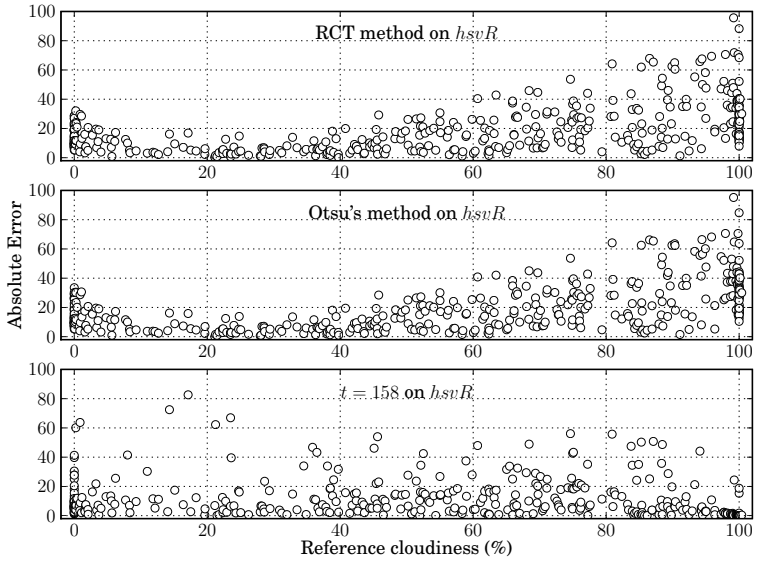

Figure 4.10: Cloud cover estimation error.

Table 4.1 shows the minimum, average, and maximum values of the absolute cloud cover estimation error for the methods employed. For the fixed threshold method, only the results obtained on the HSVR colour channel are shown as these were the best for this method. For the remaining methods, RCT and Otsu, the results achieved in the best four colour channels are presented. From the results it becomes clear that the best results are achieved over the HSVR channel, followed by the HSVg channel, which is related to HSVR. Nevertheless the two adaptive thresholding methods have similar results that are worse when compared to the fixed threshold (FT), which is the best method overall. As it was discussed earlier these segmentation methods assume that the images have two different classes, which for a significant number of images is a wrong assumption, therefore the methods are deceptively used. The outcome is that when the image has only one class the algorithm always introduces error.

Figure 4.10 highlights this characteristic by showing the absolute error values obtained in individual images for those images reference cloud fraction. As it may be seen, the results achieved by the FT method are more uniform than those obtained by the adaptive methods that exhibit increased error as the reference cloudiness approaches the limits, with emphasis to images with more clouds. The next section presents a possible solution and the results achieved.

4.5.1.2.1 Improving the Accuracy

To solve the problem of a higher error on cloudy images a number of pixels, generated randomly with the C and S class distribution were added, to the algorithms find two classes within the input pixels. These pixels were analysed and thresholded by the non-fixed methods and removed when the threshold was chosen. After that the threshold were used to get the results shown above in Table 4.2. This approach was only attempted in HSVR colour model because this colour model shows the best results. It is not difficult to generate the random pixels since the probability functions were already made and shown in the lower left plot of figure 4.4.

As we can see in figure 4.11 the results obtained with cloudy sky were better. In the Otsu's method even the maximum error decrease. These results are more reliable to use in an automatic thresholding method, but the absolute error still is to high.

Table 4.2: Segmentation algorithms results for HSVR after addition of pixels.

% added	RCT method			Otsu's method		
C/S	min.	av.	max.	min.	av.	max.
5/5	0.00	12.50	94.49	0.00	13.08	98.52
10/10	0.00	12.13	98.74	0.00	12.42	98.74
20/20	0.00	11.72	91.91	0.00	12.18	98.52
40/40	0.00	11.73	92.92	0.00	12.08	98.21
5/15	0.00	11.34	98.21	0.00	11.07	63.59

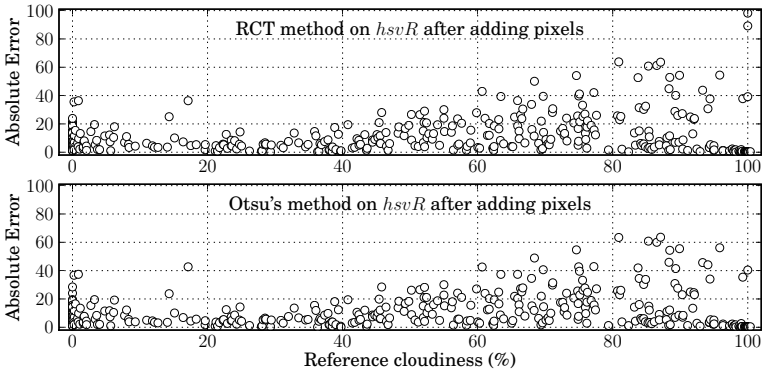

Figure 4.11: Error in cloud cover estimation after addition of pixels.

4.5.2 Neural Networks

4.5.2.1 After Statistical classification

This section presents results from the first experiments to identify a neural network, by means of a MOGA, to estimate the threshold dividing the images into the two classes. In these first attempts the data sets considered for training, generalisation, and validation purposes where those previously presented, but excluding the images that could be statistically classified using the method presented in section 4.5.1.2. The results are presented graphically in figures 4.12, 4.13, and 4.14.

Figure 4.12 presents the average of the absolute cloud cover estimation error for the preferred set of individuals resulting from the MOGA execution where the images that were successfully classified statistically were not included in the data sets. As explained before, the networks are randomly initialised, therefore for each model structure evolved,

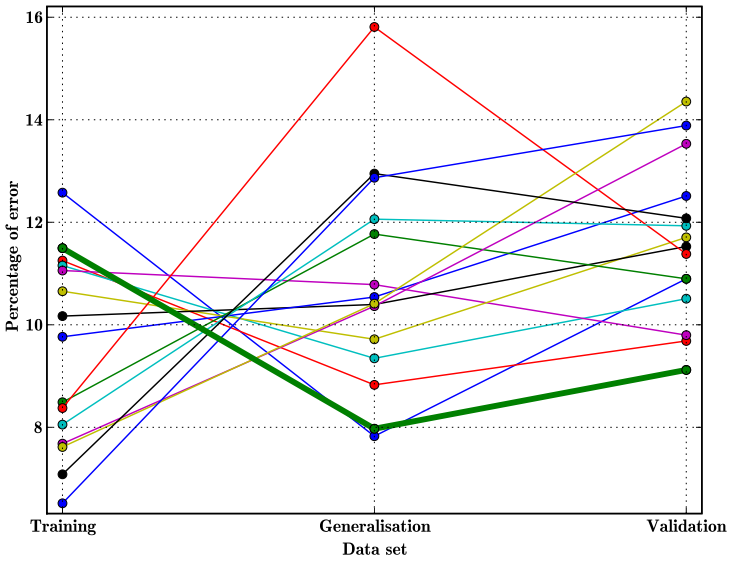

Figure 4.12: Results from the MOGA execution, when excluding images classified using the statistical approach.

50 additional training trials were executed in order to increase the chance of getting a more favourable result. A selection of the best results achieved from these trials are presented in figure 4.13. Finally, figure 4.14 presents the error performance achieved by the preferred set of networks generated on a second MOGA execution where only the images statistically classified as cloudy were removed from the data sets employed.

As it is possible to verify in the various plots shown, the choice of one single NN cannot be considered correct or definitive, it can only be regarded as a suggestion. For some of the instances it becomes easy to decide for their exclusion because they have train, test and validation errors higher than other NNs. Once these are excluded, the choice would be based in the homogeneity and significance of the various results, and on the complexity of the NN. The final choice of a NN from the first MOGA execution, highlighted by means of a thicker line, was made for a NN having 5 neurons and 15 inputs, by considering the results achieved on the testing and validation datasets.

As we can observe in figure 4.14 the results obtained when excluding only the cloudy sky images, are less homogeneous. As in the previous plots, the selected NN is denoted by a thicker line. Globally the results from the first experiment were preferable, especially when considering the validation data set results and the NN complexity, which in the second experiment was in general increased.

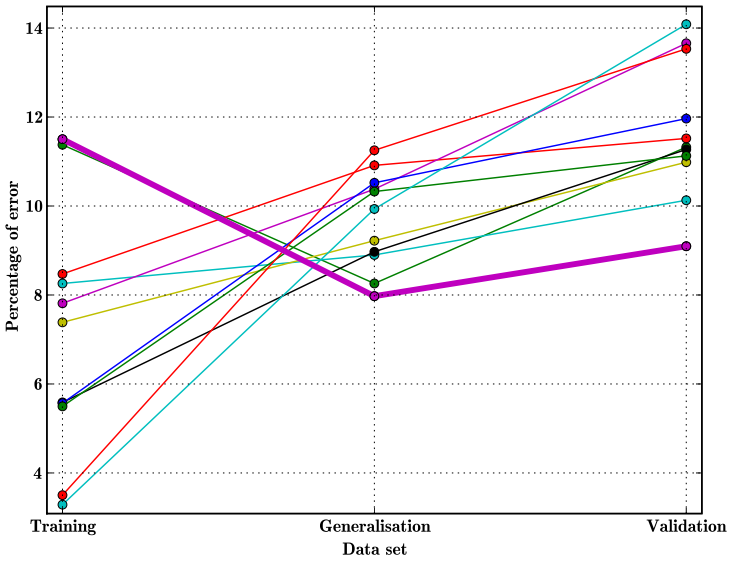

Figure 4.13: Selection of results after executing additional training trials for the preferred set of NN structures.

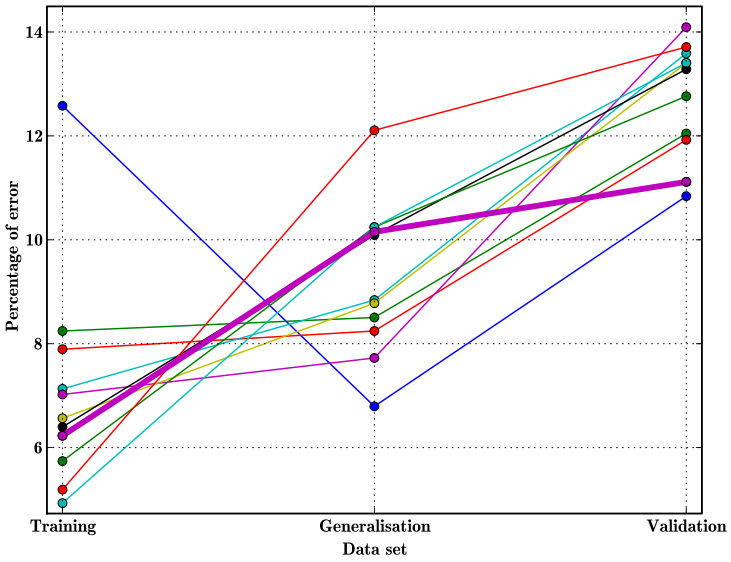

Figure 4.14: Threshold error results in train, test and validation when the statistical classification is used.

Considering the best results over the training, generalisation, and validation data sets, obtained from these two MOGA experiments, the choice relies on a NN having 5 neurons and the following 15 inputs:

- minute of the day;
- histogram peak;
- first non-zero histogram intensity;
- standard deviation of HSVR;
- mean of HSVR;
- skewness of HSVR;
- standard deviation of HSLR (Red channel after HSL transformation);
- mean of HSLR;
- skewness of HSLR;
- skewness of the red channel;
- standard deviation of the blue channel;
- mean of the blue channel;
- mean of the Value channel on HSV;
- mean of the Lightness channel on HSL;
- threshold given by the Otsu's method.

This neural network was chosen by the good results achieved, already better than those presented in the previous chapter, having also in attention the numbers of neurons and inputs.

4.5.2.2 Without Statistical classification

In this section the results are from one MOGA execution where all the images were included in the data sets, thus no statistical classification was employed, and also where the complete set of features presented in section 4.1 was considered. It should be noted that this complete set of features was not employed in the experiment described in previous section, because by the time that experiment was executed the set of features was not completely specified. After 50 generations the MOGA execution was stopped and eleven ANNs were selected because they were included in the Pareto frontier as shown in the top-left plot of figure 4.15 where a detail of the objective values obtained is depicted. The selected NNs employed between 29 and 36, the latter number being the maximum possible. Regarding the number of neurons in the hidden layer, 4 of the 11 ANNs had from 12 to 14 neurons, the remaining 7 had 22 or 23 neurons. The maximum allowed was

24.

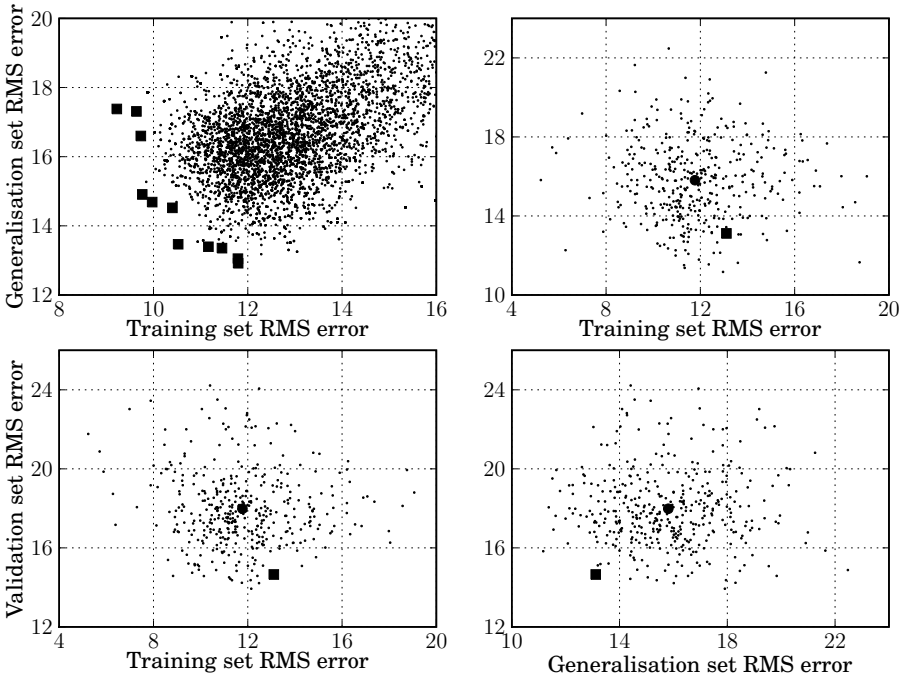

Figure 4.15: MOGA results.

The top-right plot shown in figure 4.15 shows the results after 50 additional training trials for each of the evolved ANN structures. The bottom plots consider the RMS output error of the training and generalization data sets against the RMS output error on the validation data set. In these, the black circle represent the average of the results and the black square represents the ANN chosen after a detailed analysis of the results.

The most favourable balance in training, testing and validation data sets, taking into account the objectives, presents a RMS of the output threshold estimation error of 13.10, 13.12, and 14.65, respectively. The network employed 30 out of the 69 available input features and had 22 neurons in the hidden layer.

Regarding the associated cloud cover estimation error, the results are shown in table 4.3, where the training and testing data set are presented together, as these were involved in the MOGA network optimization, along with the results obtained over the validation data set and over the entire set of 410 images. As in section 4.5.1, the statistical measures used were the minimum, the average and the maximum of the absolute cloud cover estimation error. These results are the best achieved, presenting improvements on the average error value of about 40% to 50% when compared with the previous methods.

Table 4.3: Statistical results obtained.

Data set	min.	av.	max.
Training and testing	0.00	5.31	58.46
Validation	0.00	4.74	43.71
Altogether	0.00	5.22	58.46

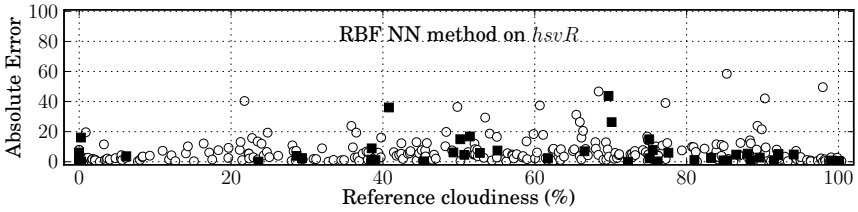

Figure 4.16: Selected RBFNN image thresholder error performance.

Regarding the first approach with ANNs, the results are clearly more homogeneous. The maximum values obtained, about ten times higher than the average, are also the best achieved among all methods. This can be confirmed in figure 4.16 which shows in black squares the results over the validation data set, and in circles the results over the training and generalisation data sets. When comparing these results with all the remaining, the improvements are remarkable on cloudy sky images.

5. Solar Radiation

5.1. Introduction

This section describes the experiments for the identification of RBFNN solar radiation prediction models. Two approaches were followed: the first consisted in identifying a NAR model, the second considered the identification of a NARX model where the cloud cover, estimated as described in the previous chapter, was the exogenous input. The goal is to conclude if there is an advantage in using the cloudiness estimation in order to obtain the solar radiation prediction.

5.2. Formulation

As for the cloudiness index determination, the MOGA was employed in order to identify the ANNs input-output structure. The solar radiation data was acquired using a one minute sampling rate and then re-sampled to a five minute rate by taking its average value at consecutive five minute intervals. In order to keep the MOGA search space at an acceptable size, the available inputs (lagged solar radiation values) were taken from the past 24 hours in the following manner: all the lagged terms from the most recent 8 hours, those taken at 10 minute intervals in the following 8 hours, and finally, the input terms taken at 15 minute intervals for the last 8 hours. This selection was made taking into account that the radiation value near the forecasting instant is more important when compared to other values as they become closer to the 24 hours before. Regarding the NARX model, the exogenous input terms considered were extracted from the 4 hours previous to the prediction instant. The number of neurons for the models was restricted to the range from 2 to 16. The training methodology, the stopping criterion and network type were the same as used in the previous chapter. Regarding the NARX model and the exogenous input, and considering a recursive multi-step prediction, when future values become necessary the measurements were employed. This approach is usually called the Minimum Error Prediction (MEP). The alternative would be to first identify a cloudiness index predictive model and afterwards use its predictions as required by the solar radiation models. This will be addressed in the future as it is out of the scope of this thesis,

which is confined to finding if, in theory, the cloud cover estimation improves the solar radiation prediction or not. The MOGA population size was set to 100 individuals, as in previous experiments, with the introduction of 10 new individuals randomly generated.

In what concerns the objectives, the primary objectives were the RMS of the output error over the testing data set and a long term prediction performance measure taken within the prediction horizon, specifically a four hour horizon. The RMS of the output error over the training data set was considered a restriction. To a better understanding of the long term prediction performance measure, figure 5.1 shows a typical error profile where each value is the RMS of the error taken at each instant within the prediction horizon, for all the instants of a simulation data set.

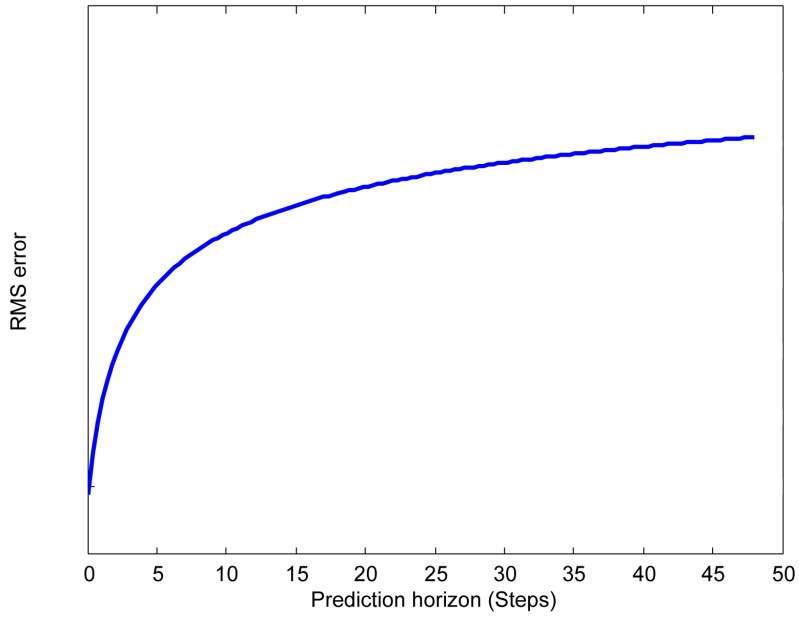

Figure 5.1: Typical evolution of the RMS of error over the prediction horizon.

The long term prediction performance measure that is to be minimised, denoted as ρ, is the sum of the RMS of error values in this profile.

5.3. Results

The MOGA execution for the identification of the NAR model was stopped after about 100 generations, generating a Pareto frontier composed by 11 ANNs, highlight by means of red dots in figure 5.2. The input layer of these 11 ANNs had a number of inputs

between 13 and 25, which means the maximum possible, 32, was not reached. Regarding the number of neurons in the hidden layer, 5 of the 11 had 6 to 9 neurons, the remaining 6 employed more than 10 or 11 neurons.

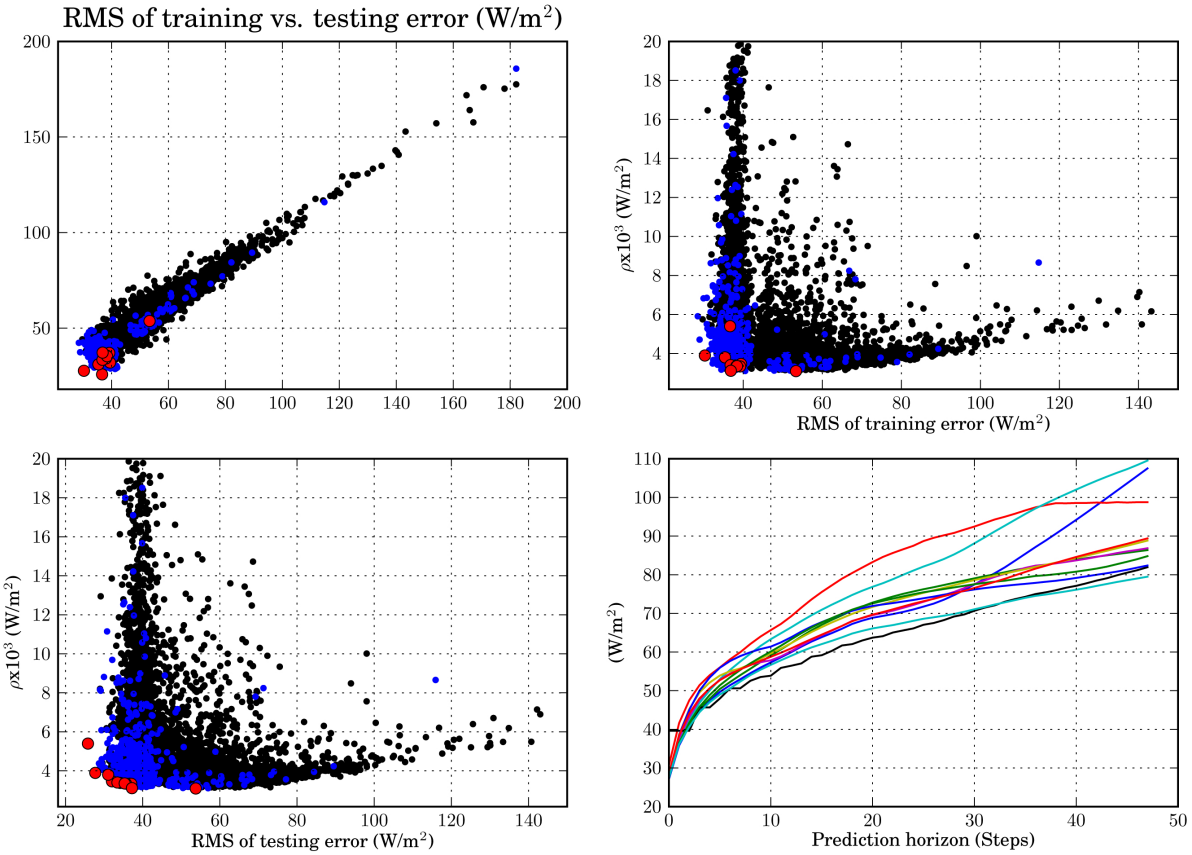

Figure 5.2: Training, testing and prediction results of NAR MOGA generated models.

The top-right and bottom-left plots of figure 5.2 present details of the relations between the training and testing errors and the prediction performance measure, ρ, for the selected ANNs. The similarity is well explained by the top-left plot presenting the training versus the testing error. The bottom-right plot presents the long-term prediction performance of the group of models evolved by the MOGA.

The MOGA execution for the identification of the NARX model was also finished at about 100 generations, again evolving a preferred set of 11 ANNs. Figure 5.3 shows the results obtained in an analogous manner to that of figure 5.2. The input layer of the 11 ANNs had a number of inputs between 15 and 28, also never reaching the maximum allowed of 32. The number of neurons in the hidden layer varied between 6 and 7, with only one network having 6 neurons

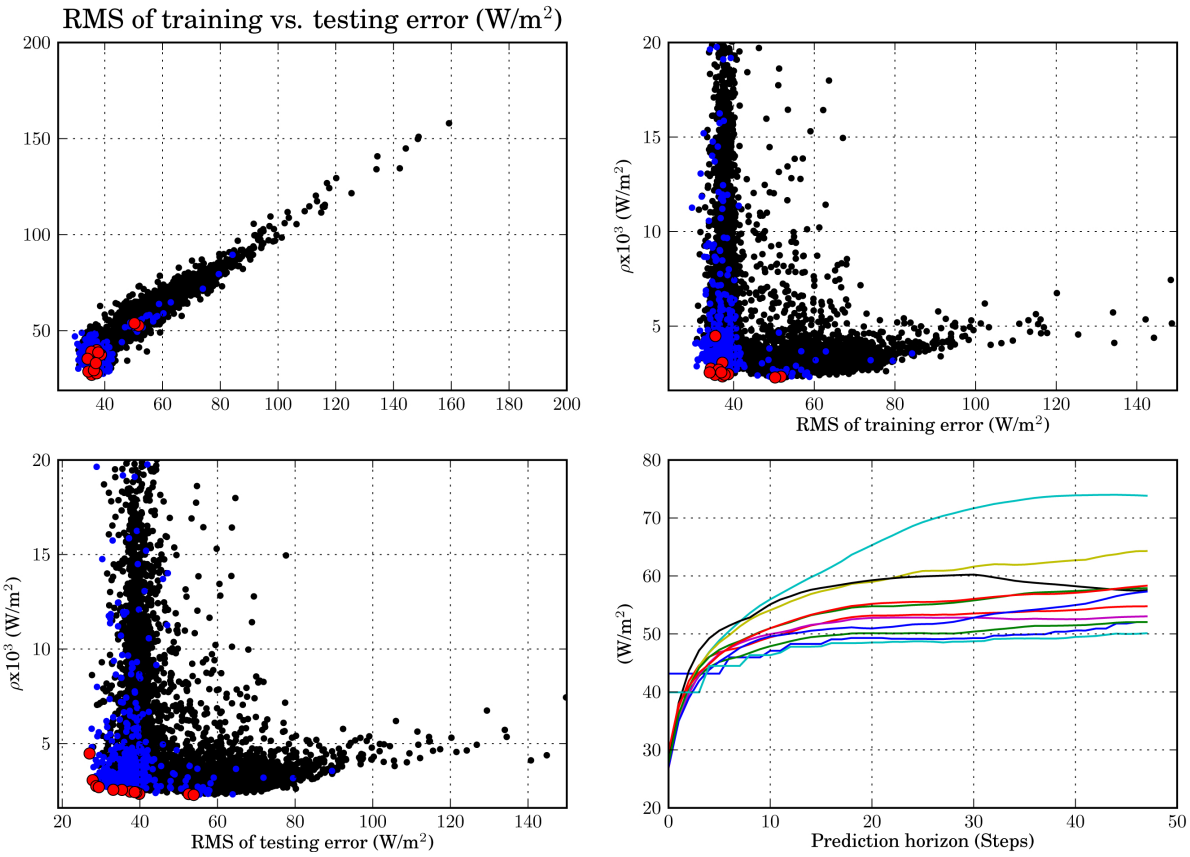

Figure 5.3: Training, test and prediction of NARX MOGA results.

The long term forecast was considered in order to compare the NAR and NARX approaches. The best result for the NAR system presented before was achieved by a neural network with 22 input terms and 6 neurons in the hidden layer. This structure achieved an RMS error of about $80 W/m^2$ at the four hour ahead prediction. The error profile shows that, although at a small rate, it is still increasing at the end of the prediction horizon. For the NARX system, the best RMS error at the four hour horizon was a bit in excess of $50\, W/m^2$, in this case having an asymptote well defined at this value. This represents an error decrease of approximately 34%. This means that the final choice should rely on the NARX approach. Even so, the conclusion can not be considered final because the measured cloud cover estimates were always employed when computing the solar radiation prediction. If cloud cover forecasts were used when required, the results would certainly be worse than the ones obtained, although probably better than the ones obtained from the NAR approach.

6. Temperature Forecast

6.1. Introduction

The method used to identify an ANN to forecast the air temperature in a four hour horizon, was similar in almost all aspects to that employed in the identification of the NARX solar radiation predictive model. In this case, the exogenous input was the solar radiation forecast and the MEP technique was also used, which means that final results considering the complete cascade of models still need to be computed in future work where the cloudiness index and solar radiation forecasts will be taken into account. A NAR temperature prediction approach was not followed because, as shown in the literature review in chapter 2 , it was proved that the outside temperature forecast with a NARX model with solar radiation as exogenous input achieves improved performance.

6.2. Formulation

The sampling rate used in data acquisition was the same as for the solar radiation model and the temperature input terms selection followed the same principles and reasoning. For the exogenous input, solar radiation in this case, the input terms were extracted from the 6 hours previous to the forecasting instant. The number of neurons for the models was restricted to the range from 2 to 20. The training methodology, the stopping criterion and network type were the same as used in the previous section. The MOGA population size was set to 100 individuals, as in previous experiments, with the introduction of 10 new individuals randomly generated.

6.3. Results

The MOGA execution was stopped after about 200 generations, in this case the Pareto frontier was composed by 7 ANNs. Figure 6.1 depicts the results obtained by the preferred set of models in an analogous manner to figures 5.1 and 5.2, also highlighting the models evolved by means of red dots. The input layer of the 7 ANNs had a number of inputs within the range from 15 to 25, not reaching the maximum of 32. Regarding the number

of neurons in the hidden layer, 6 models had 4, the other one had 5.

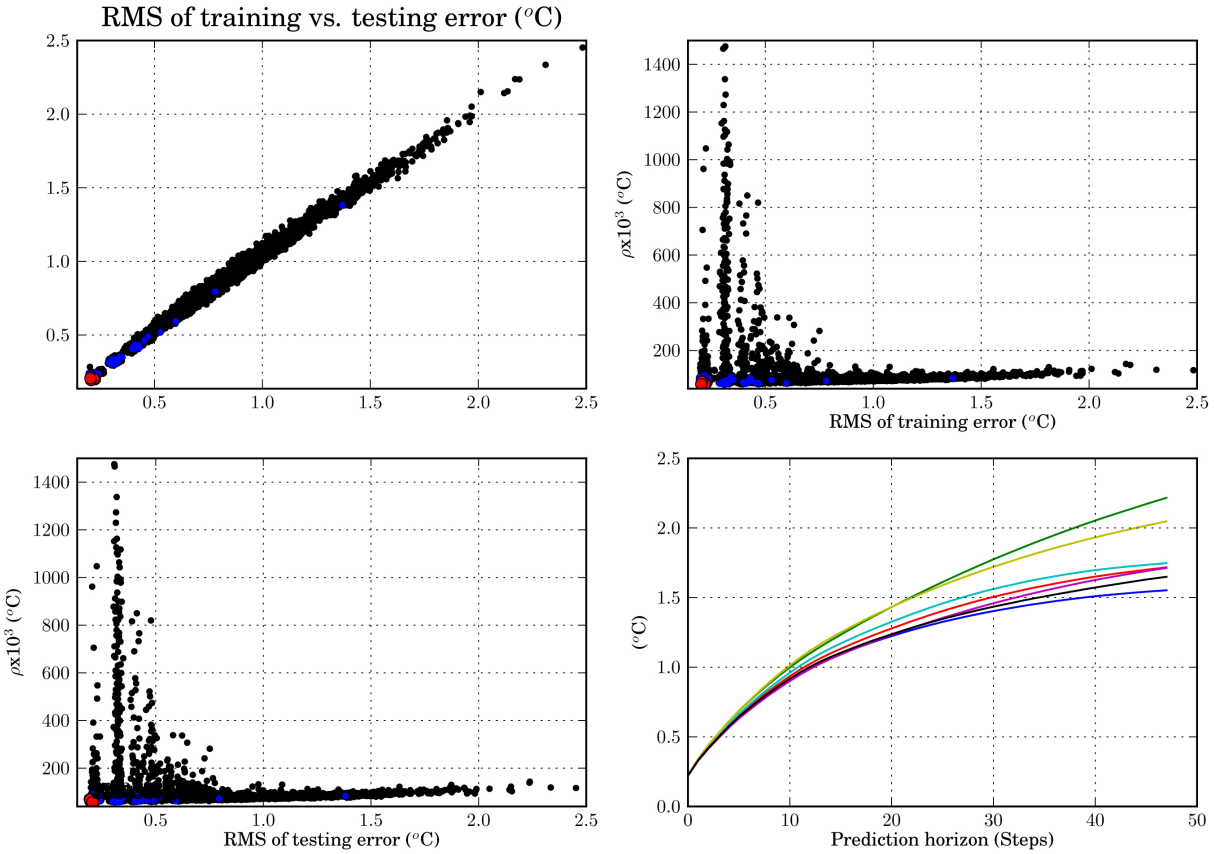

Figure 6.1: Training, testing and prediction results of NARX MOGA generated temperature models.

The best long term temperature forecast produced an error of approximately 1.5^oC at the four hour horizon, having an asymptotic behaviour right above this value. It was achieved by a NN having 19 input terms and 4 neurons in the hidden layer. As a final remark, it should be noted that this result will deteriorate due to error propagation, once the full cascade of models is considered to obtain the temperature forecast.

7. Conclusion

Considering the cloud cover estimation results achieved it may be concluded that the ANN solution is better than all other approaches, although at the cost of higher computational demand and image pre-processing effort. The results achieved were satisfactory, although it is believed that they could be improved by using other MOGA parameters and input features, by having an increased set of images, and by determining the reference cloud cover in a different way.

Regarding the prediction of solar radiation, although the conclusion is not final from the practical point of view, the results presented in this thesis show that cloudiness can be exploited in order to improve the global solar radiation prediction and consequently the temperature prediction. The comparison between the NAR and NARX approaches clearly point to this conclusion.

If the mentioned improvements in the cloud cover estimation provide results allowing a cloudiness forecast with an error performance close to the estimation error presented, the results for solar radiation prediction when using cloudiness forecasts instead of the actual estimated values, would probably be not so different than the ones achieved and still preferable to a NAR approach.

When compared to previous results analysed, the ones presented in this work provided improvement in the three aspects considered: cloud cover estimation, solar radiation prediction, and air temperature prediction.

7.1. Future Work

Regarding the cloud cover estimation by means of an ANN, other features could be added to the existing set and the identification experiments repeated in order to check if improvements would be achieved. Additional features may be formulated from different sources. For example the relation between the blue and red channels in RGB colour coded images should be investigated as these seem to provide useful information [28, 58]. Another possibility is to use frequency transformations, like the Fourier transform or the cosine transform, to formulate frequency related features.

Regarding the use of the MOGA to select NN thresholders, the objective space should

be specified in a different way. In most images ε is not symmetric around the optimum threshold, thus minimising the threshold estimation error may not guarantee the best results. A better approach would consist in building a matrix where for each image (lines) the value of ε is computed for each pixel intensity (columns), so that it becomes possible to map the NN threshold estimation to a cloud cover estimation error. The latter should be minimised in the MOGA search for NN structures.

For applications where the computational burden incurred in the identification of ANN thresholding models is to high, existing adaptive thresholding methods are probably the correct option. In this case, if the performance of the RCT and Otsu's methods is not satisfactory, other methods should be further investigated, for example the Niblack method [59].

Another improvement that could be made is on the method by which the reference cloudiness is specified. In principle, more accurate results would be possible if a group of people is asked to classify the data set of images by visually selecting a threshold for each image, and by using the average threshold as a reference. In order to accomplish that a simple application must to be developed and introduced to the group of people. This ongoing work by the time of writing this thesis.

Regarding the solar radiation and air temperature predictive models, results should be computed when considering the full cascade of models in order to make definitive conclusions about the usefulness of the cloudiness estimation on the performance of the predictive models in a realistic, practical situation. In order to improve the solar radiation prediction itself, an approach that could be followed consists in separating the clouds in two classes: dense and thin. It may be argued that this constitutes useful additional information.

The last direction suggested for future work points in the direction of the hardware and software implementation of a complete system. Nowadays there are systems that can capture all-sky images, systems to forecast temperature and radiation but there is no system that can do these tasks simultaneously on a single device. Such system will find plenty of applications on the control systems field.

References

[1] A. E. Ruano, "Artificial neural networks," 2003.

[2] C. M. Fonseca, *Multiobjective genetic algorithms with application to control engineering problems*. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 1995.

[3] H. F. Assunção, J. F. Escobedov, and A. P. Oliveira, "A new algorithm to estimate sky condition based on 5 minutes averaged values of clearness index and relative optical air mass," *Theoretical and Applied Climatology 90, Springer Verlag*, 2007.

[4] J. Calbó, J. A. González, and D. Pages, "A method for skycondition classification from ground based solar radiation measurements," *Journal of applied meteorology 40, American Meteorological Society*, 2001.

[5] D. H. W. Li and J. C. Lam, "An analysis of climatic parameters and sky condition classification," *Build Environment 36, Elsevier*, 2001.

[6] A. Orsini, C. Tomasi, F. Calzolari, M. Nardino, A. Cacciari, and T. Georgiadis, "Cloud cover classification through simultaneous ground based measurements of solar and infrared radiation," *Atmospheric Research 61, Elsevier*, 2002.

[7] M. Jurado, J. M. Caridad, and V. Ruiz, "Statistical distribution of the clearness index with radiation data integrated over five minute intervals," *Solar Energy 55, Elsevier*, 1995.

[8] J. Calbó and J. Sabburg, "Feature extraction from whole-sky groundbased images for cloud-type recognition," *Journal of Atmospheric and Oceanic Technology 25, American Meteorological Society*, 2008.

[9] K. A. Buch, C. H. Sun, and L. R. Thorne, "Cloud classification using whole sky imager data," *Conference 9, symposium on meteorological observations and instrumentation. Charlotte, NC (USA)*, 1995.

[10] R. C. González, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Prentice Hall, NJ (USA), 2004.

[11] A. S. S. Dorvlo, J. A. Jervase, and A. Al-Lawati, "Solar radiation estimation using artificial neural networks," *Applied Energy 71, Elsevier*, 2002.

[12] S. Rehman and M. Mohandes, "Artificial neural network estimation of global solar radiation using air temperature and relative humidity," *Energy Policy 36, Elsevier*, 2007.

[13] J. L. Bosch, G. López, and F. J. Batlles, "Daily solar irradiation estimation over a mountainous area using artificial neural networks," *Renewable Energy 33, Elsevier*, 2007.

[14] G. López, F. J. Batlles, and J. T. Pescador, "Selection of input parameters to model direct solar irradiance by using artificial neural networks," *Energy 30, Elsevier*, 2005.

[15] E. M. Crispim, P. M. Ferreira, and A. E. Ruano, "Prediction of the solar radiation evolution using computational intelligence technics and cloudiness indices," *International Journal Of Innovation Computing 4, Information And Control*, 2008.

[16] J. G. Fortin, F. Anctil, L. . Parent, and M. A. Bolinder, "Comparison of empirical daily surface incoming solar radiation models," *Agriculture And Forest Metereology 148, Elsevier*, 2008.

[17] F. Tymvios, C. Jacovide, S. Michaelides, and C. Scouteli, "Comparative study of angstroms and artificial neural networks methodologies in estimating global solar radiation," *Solar Energy 78, Elsevier*, 2005.

[18] C. Gueymard, "Prediction and performance assessment of mean hourly global radiation," *Solar Energy 68, Elsevier*, 2000.

[19] Y. Xia, M. Winterhalter, and P. Fabian, "Interpolation of daily global solar radiation with thin plate smoothing splines," *Theoretical and Applied Climatology 66, Springer Wien*, 2000.

[20] M. Lèfevre, L. Diabaté, and L. Wald, "Using reduced data sets isccp-b2 from the meteosat satellites to assess surface solar irradiance," *Solar Energy 81, Elsevier*, 2007.

[21] G. H. Hargreaves and Z. A. Samani, "Estimating potential evapotranspiration," *Journal of the Irrigation and Drainage Division 108, American Society of Civil Engineers*, 1982.

[22] R. Mahmood and K. G. Hubbard, "Effect of time of temperature observation and estimation of daily solar radiation for the northern great plains," *Agronomy Journal 94, American Society of Agronomy*, 2002.

[23] A. E. Ruano, E. M. Crispim, E. Z. E. Conceição, and M. Lúcio, "Prediction of building's temperature using neural networks models," *Energy and Buildings 38, Elsevier*, 2005.

[24] T. Lu and M. Viljanen, "Prediction of indoor temperature and relative humidity using neural network models: model comparison," *Neural Computing and Applications, 18(4): 345-357*, 2009.

[25] M. Nørgaard, O. Rvan, N. K. Poulsen, and L. K. Hansen, "Neural networks for modelling and control of dynamic systems," *Springer, London*, 2000.

[26] R. Tibshirani, "A comparison of some error estimates for neural network models," *Neural Comput, 8:152–163*, 1996.

[27] B. Thomas and M. Soleimani-Mohseni, "Artificial neural network models for indoor temperature prediction: Investigation in two buildings," *Neural Computing and Applications 16, Springer London*, 2006.

[28] V. R. Morris, "Total sky imager (tsi) handbook," tech. rep., U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, 2005.

[29] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems 2*, 1988.

[30] J. Park and I. W. Sandberg, "Universal approximation using radial-basisfunction networks," *Neural Computation 3*, 1991.

[31] F. Girosi and T. Poggio, "Networks and the best approximation property," *Biological Cybernetics 63, Springer Berlin / Heidelberg*, 1989.

[32] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and adaptive systems: Fundamentals through simulations*. Jonh Wiley and Sons, NY (USA), 2000.

[33] S. Haykin, "Neural networks: A comprehensive foundation," *Prentice Hall, NJ (USA)*, 1999.

[34] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation 1, MIT Press*, 1989.

[35] C. Chinrungrueng and C. H. Séquin, "Optimal adaptive k-means algorithm with dynamic adjustment of learning rate," *Neural Networks 6, IEEE Transactions*, 1995.

[36] D. Rumelhart, J. McClelland, and P. R. Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, MA (USA), 1986.

[37] A. E. Ruano, *Intelligent control systems using computational intelligence techniques*. IEE Press, London (UK), 2005.

[38] A. Ruano, D. Jones, and P. Fleming, "A new formulation of the learning problem for a neural network controller," *Proceedings IEEE Conference on Decision and Control, Brighton, U.K., pp. 865-866*, 1991.

[39] J. H. Holland, *Adaptation in Natural and Artificial Systems*. MIT Press, MA (USA), 1992.

[40] H. P. Schwefel, *Numerical optimization of computer models*. John Wiley and Sons, Inc., NY (USA), 1981.

[41] G. Roth and M. D. Levine, "Geometric primitive extraction using a genetic algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 1994.

[42] A. Hill and C. J. Taylor, "Model-based image interpretation using genetic algorithms," *Image and Vision Computing 10, Elsevier*, 1992.

[43] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," *Addison-Wesley Professional, MA (USA)*, 1989.

[44] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," *Proceedings of the Second International Conference on Genetic Algorithms and their Application (Hillsdale, New Jersey: L. Erlbaum Associates)*, 1987.

[45] P. J. B. Hancock, "An empirical comparison of selection methods in evolutionary algorithms," *Evolutionary Computation 865, AISB Workshop*, 1994.

[46] A. N. S. Zalzala and P. J. Fleming, *Genetic algorithms in engineering systems*. Cambridge University Press, NY (USA), 1997.

[47] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimisation," *Evolutionary Computing 3, AISB Workshop*, 1995.

[48] A. Ben-Tal, "Characterization of pareto and lexicographic optimal solutions," *Multiple Criteria Decision Making Theory and Application, Fandel and Gal, Eds. Springer-Verlag*, 1980.

[49] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimisation: Formulation, discussion and generalisation," *Proc. of 5th International Conference on Genetic*, 1993.

[50] H. Trussel, "Comments on picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man and Cybernetics 9*, 1979.

[51] T. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man and Cybernetics 8*, 1978.

[52] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics 9*, 1979.

[53] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging 13, SPIE*, 2004.

[54] P. M. Ferreira, E. Faria, and A. E. Ruano, "Neural network models in greenhouse air temperature prediction," *Neurocomputing 43, Elsevier*, 2002.

[55] P. M. Ferreira and A. E. Ruano, "Exploiting the separability of linear and non-linear parameters in radial basis function neural networks," *IEEE Symposium 2000: Adaptive Systems for Signal Processing, Communications, and Control, Canada*, 2000.

[56] P. M. Ferreira and C. F. A. E. Ruano, "Genetic assisted selection of rbf model structures for greenhouse inside air temperature prediction," *IEEE Conference on Control Applications, Turkey*, 2003.

[57] P. M. Ferreira, I. A. C. Martins, and A. E. Ruano, "Cloud and clear sky pixel classification in ground-based all-sky hemispherical digital images." International Federation of Automatic Control, Conference on Control Methodologies and Technology for Energy Efficiency, 2010, 2010.

[58] C. N. Long, D. W. Slater, and T. Tooman, "Total sky imager model 880 status and testing results," tech. rep., Pacific Northwest National Lab., Richland, WA (USA), 2001.

[59] W. Niblack, "An introduction to image processing," *Prentice Hall, NJ (USA)*, 1986.