UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Universidade do Algarve

Photovoltaic Power Forecast Modeling with Artificial Neural Networks

João André Martinho Bolas Soares

Dissertação

Mestrado Integrado em Engenharia Eletrónica e Telecomunicações

Trabalho efectuado sob a orientação de:

Prof. Doutor António Eduardo de Barros Ruano

Prof. Doutor Cristiano Lourenço Cabrita

2015

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Universidade do Algarve

Photovoltaic Power Forecast Modeling with Artificial Neural Networks

João André Martinho Bolas Soares

Dissertação

Mestrado Integrado em Engenharia Eletrónica e Telecomunicações

Trabalho efectuado sob a orientação de:

Prof. Doutor António Eduardo de Barros Ruano

Prof. Doutor Cristiano Lourenço Cabrita

2015

"Declaração de autoria de trabalho"

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

(João Bolas Soares)

Acknowledgments

I would like to show my greatest appreciation to Prof. Doutor António Ruano for the guidance, support and availability, without them this dissertation would not be a reality.

I would like to express to my gratitude to Prof. Doutor Cristiano Cabrita, for the availability of the data set needed to accomplish this work, for the insights and good advices on this project.

Being artificial neural networks an analogy from real neurons. And being these only capable of learning through experience. I would like to thank to my friends and my family with whom i have shared most of my experiences. Making me who i am today.

Lastly but with my greatest appreciation i would like to show my gratitude to my parents, Elsa and João Soares for the constant help and inspiring support through my academic career.

Resumo

Com uma crescente preocupação relativamente ao consumo energético global, a energia fotovoltaica surge como uma fonte energia renovável promissora. Esta dissertação é construída sob a premissa de que a capacidade de previsão de potência fotovoltaica produzida possibilita o aumento de performance da rede elétrica local através de um controlo eficiente da mesma. O trabalho desenvolvido propõe uma estrutura com a capacidade de previsão de potência produzida por um sistema fotovoltaico ligado a rede elétrica presente na Universidade do Algarve. A estrutura de previsão proposta é composta por dois modelos dinâmicos, não lineares, de previsão e um modelo estático não linear. Redes Neuronais Artificiais foram usadas como modelos. Os modelos de previsão têm como objectivo fazer previsões da temperatura do ar e irradiação solar em passos incrementais de 5 minutos para um horizonte de previsão de 4 horas. O modelo estático é construído para estimar a potência gerada pelo sistema fotovoltaico e é otimizado através de comparação entre vários tipos de redes neuronais como o perceptrão multicamadas e funções de base radial, e modelos com escalas temporais diferentes, aplicados a diferentes estações do ano, bem como um modelo anual.

Palavras-Chave: Energia Fotovoltaica, Redes Neuronais Artificiais, Previsão

Abstract

In a growing concern for the world energy consumption, photovoltaic energy sources are a reliable renewable energy alternative. This thesis is built upon the premise that the forecast of photovoltaic power production can increase performance of local electric network through an efficient network management. The work developed proposes a power production forecast structure based on a grid-connected photovoltaic system in the University of Algarve. The proposed forecast structure is composed of two non-linear dynamic forecasting models and one non-linear static model. Artificial Neural Networks were used in the development of these models which are intended to forecast solar irradiance and air temperature using Radial Basis Functions with 5 minutes time steps within a prediction horizon of 4 hours. The static model on the structure was created to estimate the power generated by the photovoltaic system and it was optimized through comparison between several network architectures (MLP and RBF) and several seasonal models, as well as a annual model.

Keywords: Photovoltaic Energy, Artificial Neural Networks, Forecast

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AC	Alternating Current
AFFNN	Artificial Feed Forward Neural Network
ANN	Artificial Neural Network
BP	Back Propagation
DC	Direct Current
ECMWF	European Center for Medium-range Weather Forecasts
FFNN	Feed-Forward Neural Network
GHG	Green House Gas
GHI	Global Horizontal Irradiance
IEA	International Energy Agency
MAPE	Mean Absolute Percentual Error
MLP	MultilayerPerceptrons
MOEA	Multi Objective Evolutionary Algorithm
MRAE	Mean Relative Absolute Error
MSE	Mean Squared Error
NOAA	National Oceanic and Atmospheric Administration
NRMSE	Normalized RMSE
NWP	Numerical Weather Prediction
PV	Photovoltaic
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SOM	Self Organized Map
STC	Standard Test Conditions
Wp	Watt peak

1 Introduction

Global energy consumption is expected to increase at least 2-fold by 2050 when comparing to the present due to population and economic growth [1], thus creating a concern for an increasing demand of more energy. In principle this demand could be met by fossil fuels, although there are parallel concerns to the exploitation of fossil fuels such has the emission of Greenhouse gases (GHG). These facts create a demand for renewable and emission free power sources, such as solar irradiance which is by far the most exploitable resource providing more energy in 1 hour than the energy consumed by humans in one year[1].

Although solar energy is a renewable and GHG emission free power source, the photovoltaic cells are completely dependent on solar radiation exposure creating some limitations to its use. Weather conditions obstructing solar radiation and day/night cycles generated by the Earth's rotation are two examples of phenomenons where the power harnessed by photovoltaic cells is disrupted or interrupted. Although the Earth's rotation is periodic, weather conditions are more complex which leads to a decreased performance of harnessed photovoltaic energy.

In order to efficiently and reliably manage solar generated power, energetically and economically, a prediction of available power becomes a necessity. The importance of these predictions allow an efficient storage of the solar generated power and an optimized control over its usage; consequently in the event of one photovoltaic structure being inefficient due to the weather conditions or day/night cycles, demanding power can be acquired from alternative power sources.

There are several perspectives in an economic sense regarding forecasting power production of renewable energies. One of the main advantages is through the forecast of solar energy production for effective storage. This allows an efficient usage of energy and through effective management of the electricity network load, any surplus power produced can be sold back to the power grid balancing the costs with hours of no power production, reducing energy related costs [2].

Another economical factor comes from the limited existence of non-renewable energy sources. According to a report of the International Energy Agency (IEA) there has been substantial market growth in 2006, with an ongoing trend in grid-connected applications[3]. Based on the supply and demand of photovoltaic technology, the reported growth from IEA can result on a more competitive market for manufactures reducing the consumer price point.

1.1 Motivation and Goals

The work developed in this thesis aims to create a forecasting model for power production using data from an array of photovoltaic panels. The forecasting model is based on artificial neural networks defined by a structure of two forecasting models for solar irradiance and air temperature serving as the input of a subsequent static model which intends to estimate a value for power production.

With the forecasting capability of artificial neural networks the work developed intends to create a reliable forecast model for the photovoltaic power production structure in University of Algarve. Through a comparison of several seasonal models over two different types of artificial neural networks, the goal is the creation of an year long forecast model.

1.2 Thesis scope

In chapter 2, a theoretical introduction to neural networks is presented along with an introduction to photovoltaic systems, air temperature and solar irradiance. Several methods to evaluate the quality of estimation and prediction of a neural network are also described. In chapter 3 the current state of the art for solar energy production forecasting is reviewed. In chapter 4, the methodology, design and implementation used in the forecasting model using neural networks is explained. In chapter 5, the results of the tests are presented and elaborated. In chapter 6, the conclusion of the work, and expected future research are discussed.

2 Theoretical Background

2.1 Artificial Neural Networks

An understanding of neural networks is essential to the identification and optimization of the parameters which will result in neural network model with refined performance.

The Neural network concept was first based and motivated on the observation of the complex biological neuron from the human brain. In biologic neurons the information that arrives in the dendrites is then integrated and passed on to the axon. The engineering perspective on the subject is an analog mathematical model composed with the weighting of the inputs followed by the sum of these inputs and bias and then processed by a transfer function. The information is then the output.

Figure 2.1 illustrates n biological neurons with several signals of intensity x and synaptic strength w feeding into a neuron with a threshold of b, and the equivalent artificial neurons system [4].

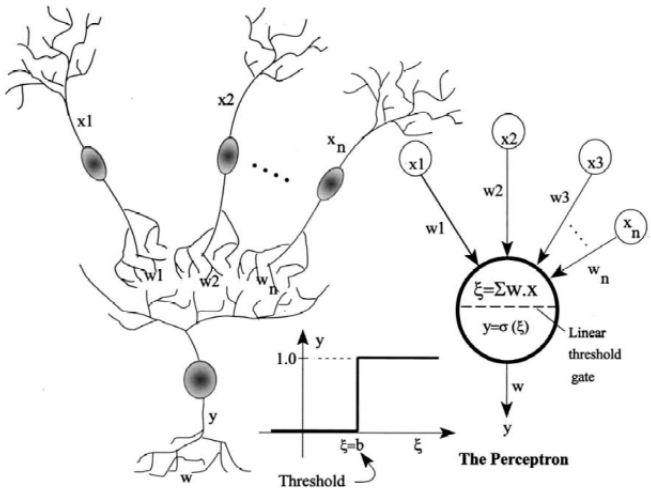

Figure 2.1: Analogy of signal interaction between n artificial neuron and n biologic neuron in a single layer configuration [4]

Although Artificial Neural Networks have a high adaptivity capability, there are several factors to be considered like the network topology and the learning algorithm. Such factors with the contextual information property of the network have a significant impact on the Neural Network performance [5].

2.2 Artificial Neural Networks properties

2.2.1 Single Input-Neuron

An Artificial Neural Network is composed of neurons, simple processing units, and weighted connections between them. In this simple model input x_i is multiplied by the corresponding weight ω_{ki} and summed with the bias b_k generating the activation potential υ_k of the neuron k represented in figure 2.2. This activation potential is then processed by the activation function $\varphi(\upsilon_k)$ generating the output accordingly to the activation function, which will be seen with further detail in section 2.2.3.

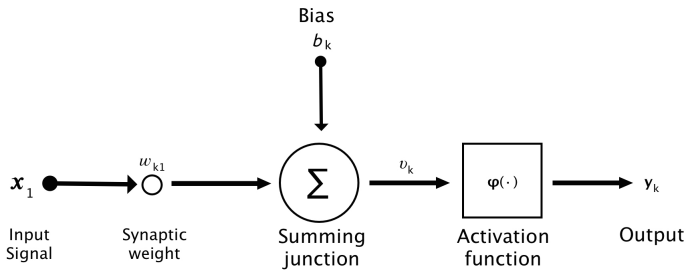

Figure 2.2: Artificial Neuron with single Input diagram

2.2.2 Neuron with an input vector

Artificial Neural Networks can have more neurons to handle multiple inputs designated input vectors, having the same concept as before. In mathematical terms it can be described by equations 2.1, 2.2 and 2.3 with respect to neuron k, for m inputs:

$$u_k = \sum_{j=1}^{m} \omega_{kj} x_j \tag{2.1}$$

$$\upsilon_k = u_k + b_k \tag{2.2}$$

$$y_k = \varphi(\upsilon_k) \tag{2.3}$$

The input vector is defined by $x_1, x_2, \ldots, x_m$ and $\omega_{k1}, \omega_{k2}, \ldots, \omega_{km}$ are the synaptic weights of the neuron k. υ_k is the activation potential resulting from the sum of the terms $\omega_{kj} x_j$ and the bias b_k. Consequently υ_k is the argument of the activation function φ.

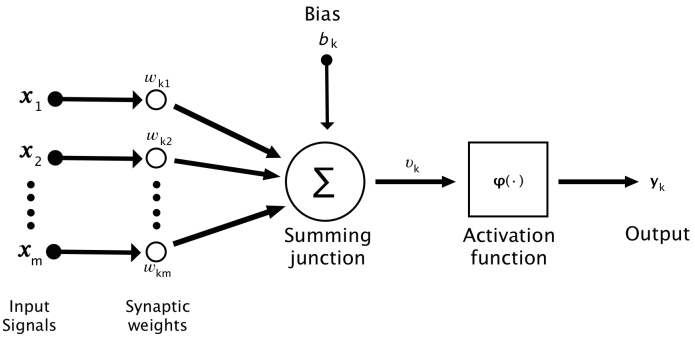

Figure 2.3: Artificial neuron with input vector diagram [5]

2.2.3 Activation function

The activation function or transfer function controls the amplitude of the output of the neuron respecting the neurons weights and its respective inputs. The most basic type of activation function are the Threshold Function, Piecewise-Linear Function and Sigmoid Function. The Threshold Function has biological inspiration and is referred to in the literature as the McCulloch-Pits model, in recognition of the pioneering work done by McCulloch and Pitts in 1943. Accordingly, the value for the threshold makes the activation function rather sensitive to small changes due to all-or-none property of the McCulloch-Pitts model [5].

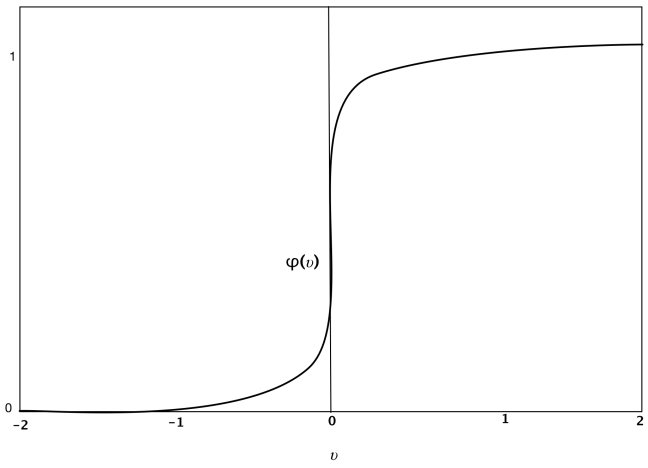

Figure 2.4: Sigmoid function [5].

The Sigmoid Function, illustrated in figure (2.4), is the most common form of activation function used in the construction of artificial neural networks. This is due to the shape of the Sigmoid function which represents a balance between linear and non-linear behavior. An example of the sigmoid function can be represented through the logistic function described in equation 2.4.

$$\varphi(v) = \frac{1}{1 + exp(-\alpha v)} \tag{2.4}$$

where α is the slope parameter of the sigmoid function and when this parameter approaches infinity the function represents the threshold function in figure 2.5.

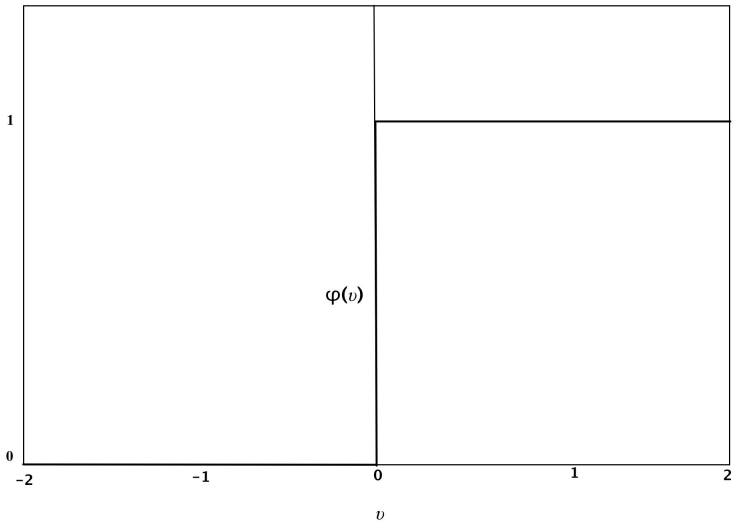

Figure 2.5: Threshold function.

This function ranges from 0 to +1, although sometimes it is desirable to range the activation function from -1 to +1 making the function symmetric with respect to the origin.

The equivalent of the sigmoids function in this particular symmetric case is a hyperbolic tangent function, defined by

$$\varphi(v) = tanh(v) \tag{2.5}$$

Allowing the activation function to range from -1 to +1, as is illustrated in figure 2.6.

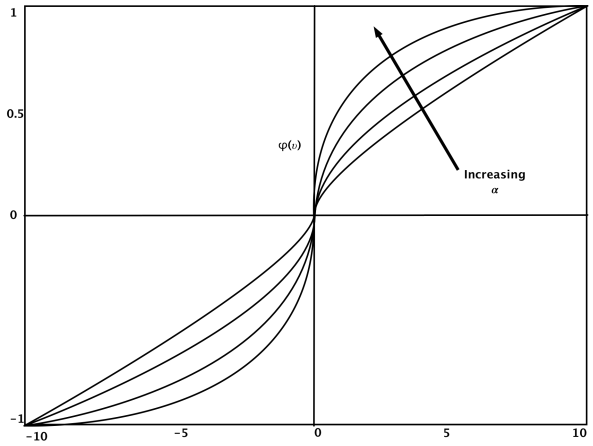

Figure 2.6: Hyperbolic tangent function[5]

2.3 Neural Networks architecture

The neuron is a nonlinear function of its inputs. The neural networks are defined by the topology of their neurons and different topologies will differentiate neural network classes in types of architecture. A topology of a neural network refers to the number of neurons per layer; figure 2.8 is an example of this representation.

2.3.1 Single-Layer Feedforward Neural Networks

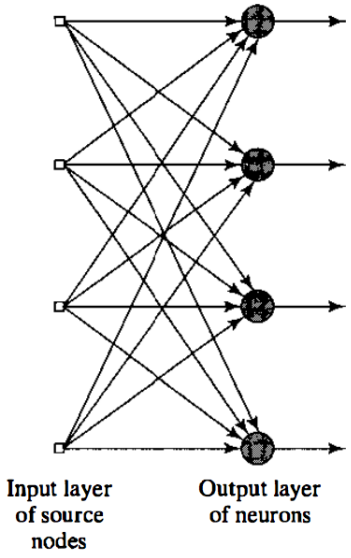

Figure 2.7: Feedforward single layer Artificial Neural Network [5].

A feedforward neural network is a composition of several neurons resulting in a non linear function of its inputs. In a feedforward network the information flows from the inputs to the

outputs only without back-loops; this type of network is also known has acyclic which translates in the lack of cycles from the output to the input. In figure 2.7 a single-layer network is represented. The denomination single-layer comes from the number of computational layers, in this case only the output layer is considered, because the input layer does not performs any kind of computation.

2.3.2 Multilayer FeedForward Neural Networks

Multilayered networks distinguishes themselves by the existence of hidden layers between the input layer and the output layer. The existence of these hidden layers affects the capacity of the network to extract higher-order statistics. This extra set of synaptic connections and higher dimensionality of neural interactions creates a global perspective on the network. To mathematically express the flow of information on a multilayered feedforward neural network, the following layer has to have the information from the preceding layer only.

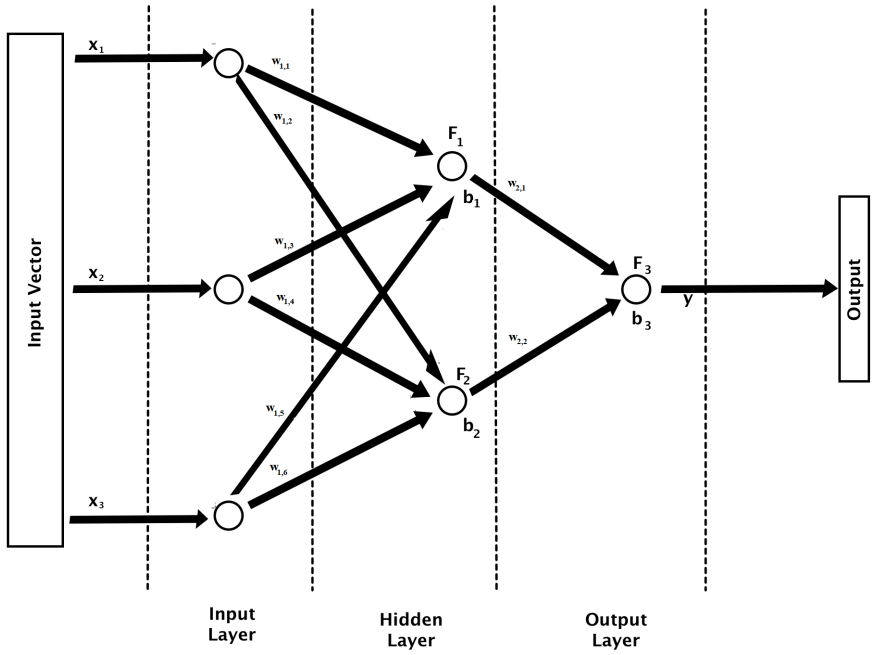

Figure 2.8: Multilayer feedforward neural network, with topology [3,2,1]

In figure 2.8 x_1, x_2 and x_3 are the input signals, y is the output signal, b_1, b_2 and b_3 are the bias, F_1,F_2 and F_3 represent the several activation functions and the $w_{c,l}$ is the l^{th} weighted connection in the c^{th} layer. The following equation describe mathematically the output when the flow of information is transited from the inputs.

$$y = F_3(w_{2,1}F_1(x_1w_{1,1} + x_2w_{1,3} + x_3w_{1,5} + b_1)+ \\ w_{2,2}F_2(x_1w_{1,2} + x_2w_{1,4} + x_3w_{1,6} + b_2) + b_3) \quad (2.6)$$

2.3.2.1 Hidden Layers

Hidden layers are composed of hidden neurons, which act as feature detectors. These features are discovered in the feature space created by the nonlinear transformation of the input data. Concerning classification purposes, patterns may become easily separable in the feature space than the input space [5].

2.3.3 Radial Basis Function

Radial Basis Functions Networks (RBFs) were firstly used for high-dimensional interpolation. They were later on introduced in the Artificial Neural Network framework by Broomhead and Lowe [6], where they were used as functional approximators and for data modeling. RBFs are also universal aproximators, and have a distinct property from the MLPs, known as the best approximation property, which states that, for a given topology there is one set of unknown coefficients that approximates better than any other set [7].

The construction of the RBF in its most simple form involves three layers with different roles represented in figure 2.9. The input layer is composed of source nodes. The second layer is the only hidden layer in the network, being composed of kernel nodes. This layer performs nonlinear transformations from the input space to the hidden space and the output layer is responsible for a linear combination of the outputs of the hidden neurons [5].

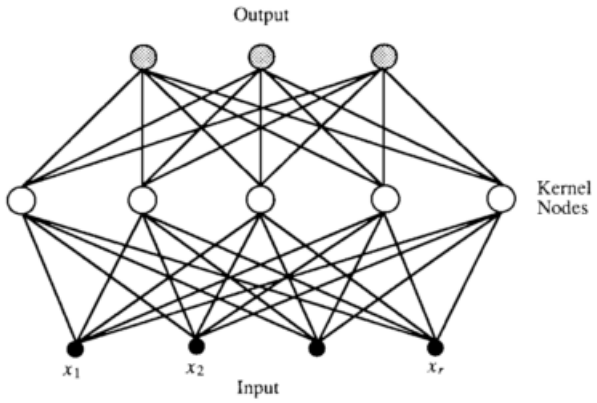

Figure 2.9: Radial Basis Function Network[8]

The hidden layer has different weights associated with each link which enables the acti-

vation function to operate on the squared distance between the center associated with the neuron, for a given input and the actual input value:

$$\mathbf{W}_{i,j} = \|\mathbf{c}_i - \mathbf{x}_j\|_2^2 \tag{2.7}$$

Where c_i is the center of the i^{th} hidden layer node, x_j represents the input value for the j^{th} iteration,$\mathbf{W}_{i,j}$ is the weight matrix, and $\| \|_2$ is the Euclidean norm [7].

The modeling capabilities of the RBF networks are determined by: the shape of the radial function, the number and placement of centers and for some specific functions, the width of the function. From the several available radial functions, the Gaussian function is the most widely used [7]. The Gaussian function is distinguishable for having a property defining it as a localized function. Considering equation 2.8 as the basis function and equation 2.9 as the Gaussian function, further defining σ as the width of the Gaussian function, the property establishes that $\varphi(r) \longrightarrow 0$ as $r \longrightarrow \infty$ [5].

$$r = \|\mathbf{c} - \mathbf{x}\|_2 \tag{2.8}$$

$$\varphi(r) = exp(-\frac{r^2}{2\sigma^2}) \tag{2.9}$$

2.4 Training Neural Networks

A very important process in the creation of an Artificial Neural Network is the training or learning. A precise definition of a learning process is difficult to formulate; a learning process in the ANN context can be viewed as the problem of updating the weights of the connections and the network architecture to efficiently make the network perform a task [9].The iterative update of weights enhances the performance of the network to recognize patterns in the given data; this process is designated as learning due to the fine tuning of the internal network weights values.

In theory, the network goal is to reach a global minimum of the error function, considering that the error function is minimized. In practice due to initial weight values several training sessions are required with different initial weight values to avoid local minima. After a consistent number of training sessions an evaluation and comparison must be done between training sessions in order to select the best session.

Due to the nature of Artificial Neural Networks there are multiple processes of training and design. These can differ in the existence or absence of a supervising entity that controls the learning process. Learning processes can also be classified as off-line or on-line, the difference is the timing of the weight adjustments. In an off-line or batch learning the

weights are adjusted only after a complete set of patterns has been presented to the network, on the other hand, in on-line or instantaneous learning a weight update is performed on the presentation of every pattern [7].

Supervised learning process implies that a network at hand is used as an input-output system. Having an input matrix I, we have an equivalent matrix of desired outputs or teacher signals also known as targets, T. The dimensions of both matrices must have the same number of patterns, m, although the number of inputs, n_i, and the number of outputs, n_o, vary accordingly to the design of the network. The objective of this learning process is to update the weights of the neural connections and bias such that, using I as input data, the corresponding output values, O, are as close as possible to T. Being the error, E, the difference between T and O the objective of the learning process becomes to minimize E [7]. A diagram exemplifying supervised learning is illustrated in figure (2.10).

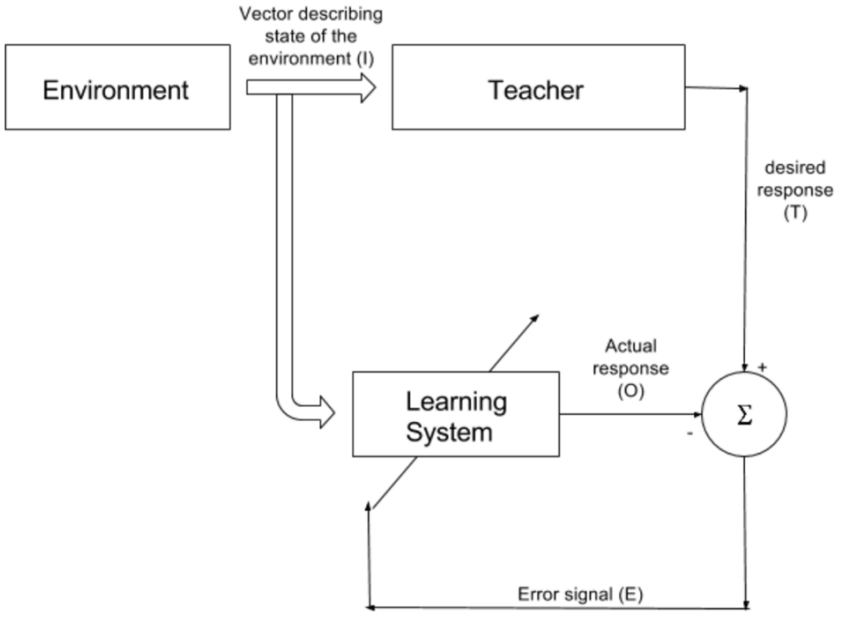

Figure 2.10: Supervised learning diagram[5]

Unsupervised learning diverges from the supervised learning in a sense where there is no supervising entity. Due to the fact that there is no supervising entity in this learning process the neural network is capable of self-organization. This learning process correlates between patterns in the data and organizes patterns into categories from these correlations [9].

Learning theory focuses on three fundamental and practical issues with learning from samples: capacity, sample complexity, and computational complexity. Capacity focuses in the quantity of patterns that can be stored, and what functions and decision boundaries a network can perform. Sample complexity quantifies the number of patterns needed to train

the network assuring a valid generalization, where few patterns may cause "over-fitting", further explained in section 2.4.1. On the other hand, computational complexity refers to the time required for a learning algorithm to estimate a solution from training patterns [9].

2.4.1 Generalization

The proper generalization of a network depends on several parameters: the selection of training data, the number of input-output patterns the network learns and the topology considered for the network.

The learning process can be viewed as a "curve fitting" problem, and generalization consequently can be compared to the effect of a good nonlinear interpolation of the input data. Overtraining and overfitting are two phenomenons that have a negative effect in the network capability to generalize. Overtraining is a phenomenon where a neural network learns too many input-output examples and may end up memorizing the training data; when this phenomenon happens the network begins to behave more like a "look-up table", losing the ability to generalize between similar input-output patterns [5]. This phenomenon will be further explained in section 2.4.1.1. The overfitting refers to exceeding the optimal ANN size which may result in a worse predictive ability of the network [10]. Also, a common concern described in the academic literature related to ANN structure design is the

"curse of dimensionality". This designation was first used by Bellman [11] when the amount of data required to generalize accurately grows exponentially with the amount of input variables or dimensions.

In order to have a network with a good generalization capability, the selection of number of hidden neurons and their organization is highly relevant. A network with too few hidden neurons would be incapable of differentiating between complex patterns leading to only a linear estimate of the actual trend. In contrast if the network has too many hidden neurons it might lead to overparametrization, which translates to the learning of unwanted characteristics such as noise, subconsequently leading to poor generalization for untrained data. There are several rules of thumb to apply in the selection of topology for the network, although in popular fashion it is common the trial and error approach, beginning with a small number of hidden neurons and building on as deemed necessary. In order to ensure good results from this technique it must be used with a cross-validation method which will be explained in section 2.4.1.1.

2.4.1.1 Overtraining

There are techniques to ensure the maximum generalization, one of the most used is the early-stopping method of training. In this early-stopping method a consistent and periodic cross-validation is performed. This cross-validation process implies the random partition of

available data into two subsets, a training set and a test set. A training session will then be interrupted periodically and tested with the test set, ensuring performance. This performance is verified with the decrease of the mean squared error in the test subset so that when the error stops decreasing and starts increasing, a minimum was found. If the error persists on increasing after a specific number of epochs, then the training should be halted due to the risk of compromising the performance of the network [5]. Figure 2.11 illustrates the performance of the mean-squared error when early-stopping is applied.

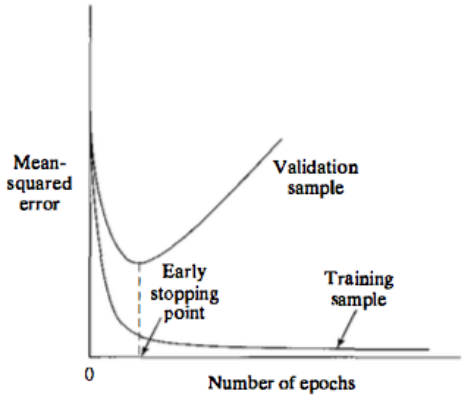

Figure 2.11: Illustration of the early-stopping method based on cross-validation[5]

2.4.2 Training Algorithm

The learning algorithm is the method responsible for calculating the update of the free parameters in a Neural Network. These are the parameters allowed to change during the training: the weights and bias of the neural network. When applied with a specific training set an error is calculated, being this error the difference between the output and the desired output. This error is then squared and back propagated through the hidden neurons to update the weights of the connections; this process is repeated until a convergence in a minimum error solution is verified. This algorithm is known as the error back-propagation algorithm (BP) and is widely used although it has some limitations, like slow convergence and traps in local minima, due to the fixed update percentage in the negative gradient, g, direction.

2.4.2.1 Steepest Descent

In this method, implemented by the BP algorithm, the adjustment of the weight vector $\mathbf{w}$ is made in the direction of the steepest descent, that is the opposite direction of the gradient g for the k^{th} iteration. The steepest descent uses first order derivatives of the error function expressed in equation 2.11, where the vector $\mathbf{e}[k]$ is defined in equation 2.10 by the

difference between the target, $\mathbf{t}[k]$, and the net output, $\mathbf{y}[k]$. The purpose of the first order derivatives is to identify the minimum in the error space and also define the gradient vector, mathematically expressed in equation 2.12.

$$\mathbf{e}[k] = \mathbf{t}[k] - \mathbf{y}[k] \tag{2.10}$$

$$E = \frac{1}{2}\mathbf{e}^t\mathbf{e} = \frac{1}{2}\sum_{i=1}^{N}\mathbf{e}^2\,[k] \tag{2.11}$$

$$\mathbf{g} = \nabla E(w) = \left[\frac{\partial E}{\partial \mathbf{w}_1}, \frac{\partial E}{\partial \mathbf{w}_2}, \ldots, \frac{\partial E}{\partial \mathbf{w}_M}\right]^T \tag{2.12}$$

This is described has the gradient of the cost function, where $\mathbf{w}$ are the weights of synaptic connections of the artificial neuron. The weight update in this method is presented in equation 2.13.

$$\mathbf{w}[k+1] = \mathbf{w}[k] - \eta\mathbf{g}[k] \tag{2.13}$$

Where η is a positive constant designated as learning rate or step size. In this algorithm the choice of a value for η is very important, incorrect learning rate values will have consequences. A small learning rate value will have over-damped behavior creating a very smooth but slow response in convergence; if it is large the transient response may be underdamped and its trajectory may follow a oscillatory path to convergence. If the learning rate surpasses a certain critical value the algorithm becomes unstable and will lose its capability to converge[5]. A simple method to increase the learning rate avoiding instability uses an additional *momentum constant,* α, including in expression 2.13 a portion of the last weight change as shown in equation 2.14. The usage of this constant allows a progressive convergence and filtering of the oscillations.

$$\mathbf{w}[k+1] = \mathbf{w}[k] - \eta\mathbf{g}[k] + \alpha(\mathbf{w}[k] - \mathbf{w}[k-1]) \tag{2.14}$$

2.4.2.2 Newton's Method

As an alternative, Newton's method uses a quadratic approximation of the cost function $E(\mathbf{w})$ around the current point $\mathbf{w}[k]$. In detail, this method uses second-order Taylor series expansion of the cost function around the point $\mathbf{w}[k]$.

$$\triangle E(\mathbf{w}[k]) = E(\mathbf{w}[k+1]) - E(\mathbf{w}[k]) \tag{2.15}$$

$$\approx \mathbf{g}^t[k]\triangle\mathbf{w}[k] + \frac{1}{2}\triangle\mathbf{w}^t[k]\mathbf{H}[k]\triangle\mathbf{w}[k] \tag{2.16}$$

Additionally to the gradient vector the Hessian matrix of the Error function, $\mathbf{H}[k]$, is also evaluated at $\mathbf{w}[k]$. This Hessian Matrix can be defined by equation 2.17, where M is the number of free parameters.

$$\mathbf{H} = \nabla^2 E(\mathbf{w}) = \begin{bmatrix} \frac{\delta^2 E}{\delta w_1^2} & \frac{\delta^2 E}{\delta w_1 \delta w_2} & \cdots & \frac{\delta^2 E}{\delta w_1 \delta w_M} \\ \frac{\delta^2 E}{\delta w_2 \delta w_1} & \frac{\delta^2 E}{\delta w_2^2} & \cdots & \frac{\delta^2 E}{\delta w_2 \delta w_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 E}{\delta w_M \delta w_1} & \frac{\delta^2 E}{\delta w_M \delta w_2} & \cdots & \frac{\delta^2 E}{\delta w_M^2} \end{bmatrix} \tag{2.17}$$

The Hessian matrix requires the cost function $E(\mathbf{w})$ to be twice continuously differentiable with respect to the elements of $\mathbf{w}$. Solving in order to $\triangle\mathbf{w}[n]$ we have:

$$\triangle\mathbf{w}[k] = -\mathbf{H}^{-1}[k]\mathbf{g}[k] \tag{2.18}$$

which translates into

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \triangle\mathbf{w}[k] \tag{2.19}$$

Newton's method has several limitations, a major one being its computational complexity. Although this method does not exhibit oscillatory behavior, for this method to work the Hessian matrix $\mathbf{H}[k]$ has to be definite positive for all k [5].

2.4.2.3 Gauss-Newton Method

The Gauss-Newton method reduces the computational complexity of Newton's method without seriously compromising its convergence behavior. To apply this method a cost function expressed by equation 2.11 is adopted. Therein $\frac{1}{2}$ is a scaling factor to simplify future analysis. The error terms in this formula are calculated on the basis of a weight vector $\mathbf{w}$ that is fixed over the entire observation interval $1 \leq i \leq N$. Linearizing the dependence of the error signal, $e(i)$, on $\mathbf{w}$ results in expression 2.20.

$$e'(i, \mathbf{w}) = e(i) + \left[\frac{\partial e(i)}{\partial \mathbf{w}}\right]^T_{\mathbf{w}=\mathbf{w}[n]} \times (\mathbf{w} - \mathbf{w}[k]), \quad i = 1, 2, \ldots, N \tag{2.20}$$

Alternatively, expression 2.20 can also be written in matrix notation, as expressed in equation 2.21.

$$\mathbf{e}'(n, \mathbf{w}) = \mathbf{e}[k] + \mathbf{J}[k](\mathbf{w} - \mathbf{w}[k]) \tag{2.21}$$

Where $\mathbf{e}[k]$ is the error vector in 2.22 and 2.23 is its N-by-M Jacobian matrix, $\mathbf{J}[k]$.

$$\mathbf{e}[k] = [e(1), e(2), \ldots, e(N)]^T \tag{2.22}$$

$$\mathbf{J}[k] = \begin{bmatrix} \frac{\delta e(1)}{\delta w_1} & \frac{\delta e(1)}{\delta w_2} & \cdots & \frac{\delta e(1)}{\delta w_M} \\ \frac{\delta e(2)}{\delta w_1} & \frac{\delta e(2)}{\delta w_2} & \cdots & \frac{\delta e(2)}{\delta w_M} \\ \vdots & \vdots & & \vdots \\ \frac{\delta e(N)}{\delta w_1} & \frac{\delta e(N)}{\delta w_2} & \cdots & \frac{\delta e(N)}{\delta w_M} \end{bmatrix}_{\mathbf{w}=\mathbf{w}[k]} \tag{2.23}$$

The Jacobian, $\mathbf{J}[k]$, is the transpose of the M-by-N gradient matrix $\nabla\mathbf{e}[k]$:

$$\nabla\mathbf{e}[k] = [\nabla e(1), \nabla e(2), \ldots, \nabla e(N)] \tag{2.24}$$

The update weight vector $\mathbf{w}[k+1]$ is defined in expression 2.25.

$$\mathbf{w}[k+1] = \arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \left\| \mathbf{e}'(n, \mathbf{w}) \right\|^2 \right\} \tag{2.25}$$

Using equation 2.21 to evaluate the squared euclidean norm of $\mathbf{e}'(n, \mathbf{w})$; then differentiate the resulting expression with respect to $\mathbf{w}$ and setting the result equal to zero, concludes in expression 2.26.

$$\mathbf{J}^T[k]\mathbf{e}[k] + \mathbf{J}^T[k]\mathbf{J}[k](\mathbf{w} - \mathbf{w}[k]) = 0 \tag{2.26}$$

Solving expression 2.26 for $\mathbf{w}$ results in the Gauss-Newton weight update expression shown in equation 2.27.

$$\mathbf{w}[k+1] = \mathbf{w}[k] - (\mathbf{J}^T[k]\mathbf{J}[k])^{-1}\mathbf{J}^T[k]\mathbf{e}[k] \tag{2.27}$$

In contrast to Newton's method , Gauss-Newton does not need knowledge about the Hessian of the cost function $E[k]$, the only requirement being the Jacobian of the error vector $\mathbf{e}[k]$. Consequently, to compute a Gauss-Newton iteration, the matrix product from $\mathbf{J}^T[k]\mathbf{J}[k]$ must be nonsingular, which means that the matrix product must always be nonnegative definite [5].

2.4.2.4 Levenberg-Marquardt method

The Levenberg-Marquardt method, due to Levenberg (1994) and Marquardt (1963), is an adjustment between the Newton's method and the gradient descent method [5]. This method uses the solution to the following system as a search direction

$$(\mathbf{J}^T[k]\mathbf{J}[k] + \upsilon[k]\mathbf{I})\mathbf{p}_{LM}[k] = -\mathbf{J}^T[k]\mathbf{e}[k] \tag{2.28}$$

The scalar $\upsilon[k]$ controls both the magnitude and direction of $\mathbf{p}[k]$. When $\upsilon[k]$ is zero $\mathbf{p}[k]$ is identical to the Gauss Newton direction and if $\upsilon[k]$ tends to infinity, $\mathbf{p}[k]$ tends to a vector of zeros, and a steepest descent direction .

Unlike previously mentioned step-lenght methods, the Levenberg-Marquardt is a "trust-region" method type, which translates in a method that makes an attempt to define the neighborhood region with a model representing an approximation of such region; such models are often represented by quadratic functions obtained by the Taylor series expansion of that region. If this approximated model proves to be good, the region is denominated as the trust-region and becomes the new point in optimization, otherwise it may be rejected and the neighborhood is constricted. The radius of this neighborhood is tuned by the parameter υ, usually denominated as the regularization factor.

In order to have this value of agreement or rejection a comparison between the quadratic assumption and the real function is performed. Assuming a predicted error vector, after step $\mathbf{p}[k]$ [7].

$$\mathbf{e}^p[k] = \mathbf{e}[k] - \mathbf{J}[k]\mathbf{p}[k] \tag{2.29}$$

The predicted reduction of Ω is:

$$\Delta\Omega^p[k] = \Omega(\mathbf{w}[k]) - \frac{(\mathbf{e}^p[k])^T(\mathbf{e}^p[k])}{2} \tag{2.30}$$

As the actual reduction is translated into:

$$\Delta\Omega[k] = \Omega(\mathbf{w}[k]) - \Omega(\mathbf{w}[k] + \mathbf{p}[k]) \tag{2.31}$$

The ratio that measures the quality of the approximation is then represented by $r[k]$, in a sense where the proximity of $r[k]$ to unity is a better approximation.

$$r[k] = \frac{\Delta\Omega[k]}{\Delta\Omega^p[k]} \tag{2.32}$$

If $r[k]$ is approximately one, the region increases, if $r[k]$ is shown to be small, the region is decreased, if $r[k]$ is smaller than a certain threshold , the step is rejected and the value of w_k is retained, decreasing the region [12]. In the scope of this work the Levenberg-Marquardt algorithm will be used to train the different ANN models.

2.4.3 Network Performance Evaluation

This section introduces several mathematical criteria used to measure the overall performance of a neural network. There is a wide variety of criteria that are expressed through mathematical equations and create an efficient and diversified manner of evaluating the global performance of the neural network.

One of this performance measurement criteria is the Mean Squared Error (MSE), shown in equation (2.33); this criteria measures the squared sum of the difference between the expected output, t, and the real output, y.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y(i) - t(i))^2 \tag{2.33}$$

Although the mean square error can be translated into an error value, depending on the evaluation, there are some criteria that can indicate certain features when comparing evaluations. Root mean square error (RMSE) indicates the sample standard deviation for given differences between expected output and real output. RMSE can be translated to equation (2.34). As a complement to RMSE, normalized RMSE (NRMSE), serves the purpose of measuring percentile residual variance of the error and can be mathematically described by equation (2.35).

$$RMSE = \sqrt{MSE(\theta)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(i) - t(i))^2} \tag{2.34}$$

$$NRMSE = \frac{RMSE}{t_{max} - t_{min}} \tag{2.35}$$

Another performance evaluator to be considered is the mean absolute percentage error (MAPE). MAPE allows a representation in percentage of the absolute error. The mathematical representation of MAPE is shown in equation (2.36).

$$MAPE = 100(\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\mathbf{y}(i) - \mathbf{t}(i)}{\mathbf{t}(i)}\right|) \tag{2.36}$$

2.5 Photovoltaic Systems

The purpose of Photovoltaic (PV) systems is to harness electromagnetic radiation emitted from the nuclear fusion occurring in the sun into electric energy. This fact makes the sun a very reliable and durable power source. To harness the energy in the photons from the sun, photovoltaic cells are used. These cells are the most basic device of a PV system which can be grouped and composed of panels or arrays. Although solar radiation can be described by a specific set of equations, the efficiency of the conversion from solar radiation

to electric current varies according to atmospheric variables such has temperature. In this section the air temperature and solar irradiance influence in photovoltaic systems efficiency will be explained, followed by photovoltaic cells technology and several power configurations of photovoltaic systems.

2.5.1 Air Temperature

Temperature is an important variable when considering any electronic circuit. In this specific case temperature conditions the efficiency of the photovoltaic structure, the air temperature in the proximity of the cells is typically used as an indicator. The crystalline silicon solar cell was one of the first developed and it is still the most widely type used. Air temperature will in fact be incorporated as a variable in the modeling process due to its variability shown in figure (2.12) and influence on the final results.

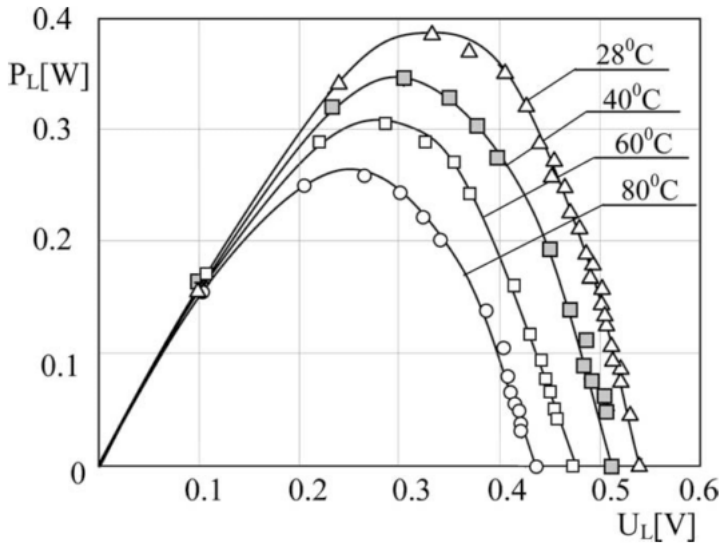

Figure 2.12: Output power versus voltage of a single crystalline silicon solar cell at various temperatures [13].

Based on experimental data crystalline silicon solar cells can decrease the output power by 0.4% per 1^{o}C [13]. This becomes highly relevant when dealing with a considerable amount of electric power and temperature amplitudes recorded in the duration of one year.

2.5.2 Solar Irradiance

Solar exposition, one of the most important factors to consider when using this type of renewable energy is dependable on parameters like Solar azimuth, elevation, zenith and distance from Earth to the Sun. Since solar exposition time windows and angles of incidence

vary greatly across the globe, efficiency also depends on geographical factors. Although practical solar irradiance values vary due to cloudiness, there are mathematical models defined by a set of equations that provide theoretical values for solar irradiance at given position on the planet surface. Depending on the latitude, longitude, altitude, azimuth and tilt of the photovoltaic panel a theoretical approximation of solar irradiance can be obtained. To achieve that, the elliptical orbit of the Earth around the Sun has to be taken in consideration. The ecliptic plane is the designation used for the plane corresponding to the Earth's elliptic orbit defined by its period of one Earth year and can be expressed by equation 2.37.

$$r = r_\theta \left[1 + 0.017 sin(\frac{360(d_n - 93)}{365})\right] \tag{2.37}$$

Where d_n is the day number counted from the beginning of the year and r_θ is equal to 1.496×10^8 km also known as one astronomical unit (1 UA). The Earth also spins around its own central axis (polar axis), maintaining a constant angle of 23.45^{o} with the ecliptic plane. Solar declination, δ, is the angle between the equatorial plane and a straight line between the Earth's center and the Sun's center; this angle is calculated from equation 2.38 and is illustrated in figure 2.13.

$$\delta = 23.45^{\text{o}} sin\left[\frac{360(d_n + 284)}{365}\right] \tag{2.38}$$

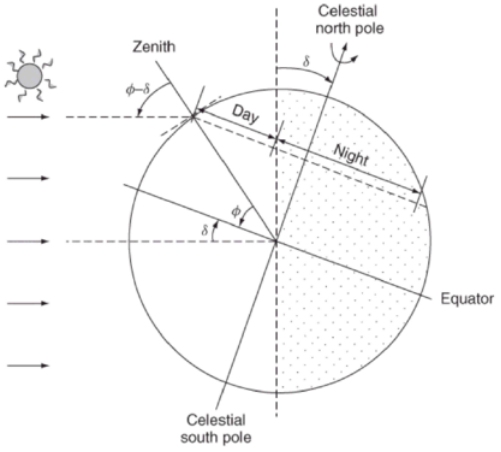

Figure 2.13: Relative Earth-Sun position at noon on winter day in the Northern Hemisphere [14]

Referring to a particular location on the Earth's surface where a certain PV system is being used, it is important to specify the Sun's position by using two angles in reference of the horizontal plane and the vertical plane. Illustrated in figure 2.14, the solar zenith, θ_{ZS} , is the angle between the vertical and the incident solar beam, the solar azimuth, ψ_S, is

defined by the angle between the incident beam and the meridians on the horizontal plane and the solar altitude or elevation, γ_S, is the complement angle of the solar zenith.

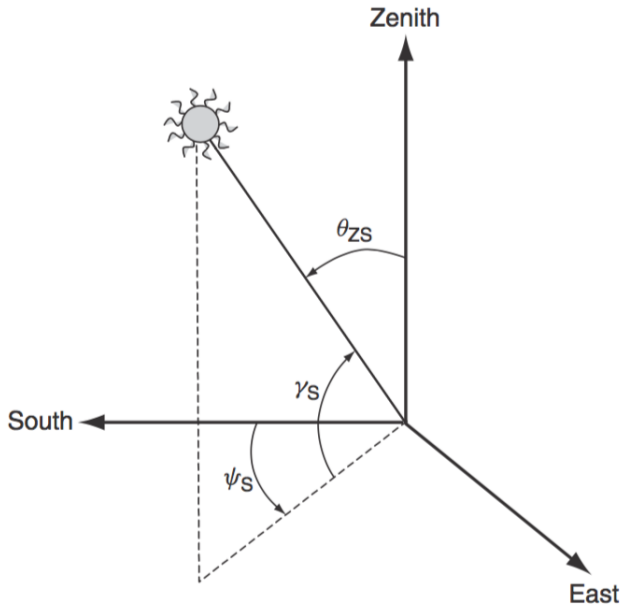

Figure 2.14: Sun's position relative to earth defined by two angles, ψ_S (azimuth) and θ_{ZS} (solar zenith). γ_S represents the altitude on the Earth's surface[14].

The angular coordinates of the sun with respect to a specific latitude ϕ are calculated from equation 2.39 and 2.40.

$$cos(\theta_{ZS}) = sin(\delta)sin(\phi) + cos(\delta)cos(\phi)cos(\omega) = sin(\gamma_S) \tag{2.39}$$

$$cos(\psi_S) = \frac{(sin(\gamma_S)sin(\phi) - sin(\delta))}{cos(\gamma_S)cos(\phi)}\,[sign(\phi)] \tag{2.40}$$

$$\omega = 15 \times (TO - AO - 12) - (LL - LH) \tag{2.41}$$

In equation 2.41, ω is the true solar time or solar hour and represents the difference between noon and the selected moment of the day in terms of a 360^{o} rotation in 24h. Values range from negative to positive being midday at each day $\omega = 0$. ω takes time values based in the difference from a specific longitude,LL, and a reference longitude, LH, in degrees with respect to the standard time, TO, and to the present time that clocks are set ahead of the local time zone. The $[sign(\phi)]$ in equation 2.40, is a reference to latitudes, taking value of "1" for the northern latitudes and "-1" for southern latitudes.

Assuming a receiver photovoltaic panel with a tilt β, and azimuth α, as shown in figure

2.15.

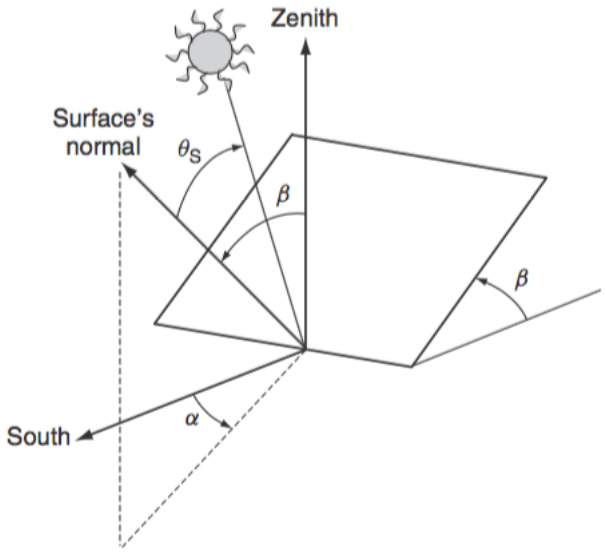

Figure 2.15: Receiver PV panel (tilt β, azimuth α) and sun beam incidence angle [14].

The resulting angle from the solar incidence and the normal of the panel can be calculated from equation 2.42.

$$\begin{aligned} cos(\theta_S) = sin(\delta)sin(\phi)cos(\beta) - [sign(\phi)]\, sin(\delta)cos(\phi)sin(\beta)cos(\alpha) + \\ cos(\delta)cos(\phi)cos(\beta)cos(\omega) + [sign(\phi)]\, cos(\delta)sin(\phi)sin(\beta)cos(\alpha)cos(\omega) + \\ cos(\delta)sin(\alpha)sin(\omega)sin(\beta) \end{aligned} \quad (2.42)$$

When the sun's irradiance reaches the Earth atmosphere it has a constant value of $1367 W/m^2$; this value is considered as such due to the lack of matter in the intermediate space between Earth's atmosphere and the Sun that would reflect or scatter the solar radiation. Such constant is designated by B_0 for further mathematical purposes. On the Earth's atmosphere, the solar irradiance can be deviated or absorbed due to reflection from the clouds, absorption by some atmosphere components such as ozone, oxygen, carbon dioxide, and water vapor, solar irradiance can also be scattered by water droplets or dust transforming part of the original direct irradiance into diffuse radiation [14].

To consider this effect on the radiation beam the concept of Air Mass exists. The purpose is to approximate the amount of air mass the direct radiation has to pass until it reaches a designated position on the surface of the Earth. This air mass calculation can be done with equation 2.43 [15].

$$AM = \frac{1}{cos(\theta_{ZS}) + 0.50572(96.07995 - \theta_{ZS})^{-1.6364}} \tag{2.43}$$

The approximation of the intensity of solar irradiance can be described by the following equations.

$$I_D = 1.353 \left[(1 - ah) 0.7^{AM^{0.678}} + ah \right] \tag{2.44}$$

where $a = 0.14$, h refers to the height of receiving panel and AM refers to the air mass [16].

2.5.3 Photovoltaic Technology

A photovoltaic cell is essentially a semiconductor diode whose p-n junction is exposed to light; due to its four valence electrons silicon is often used has the base material for these semiconductors because of the tetrahedral stable arrangement. To create a p-n junction one of the sides of the silicon semiconductor must have impurities, a process called "doping".

This process can be negative-carrier or positive-carrier; this relies in the number of valence electrons of the dopant. A dopant such as phosphorus has five valence electrons, if replaced by one of the silicon atoms, due to the extra valence electron in phosphorus, one electron would be become free if enough energy were to be transferred to it, and the phosphorus atom would be positively charged. The positive-carrier doping process relies in the absence of electrons, the opposite principle of the negative-carrier process, where boron atoms are used as dopant and contains 3 valence electrons; this originates "holes", making the silicon atoms positively charged when an electron would be freed.

An imbalance in the p-n junction is created which originates a fixed potential electric field barrier that stops the crossing of holes or electrons. When light strikes the junction, if it has enough energy, it will create a free electron and a free hole. The reaction of the free pair with the junction electric field will be to repulse the electron to the n-type side and the hole to the p-type side.

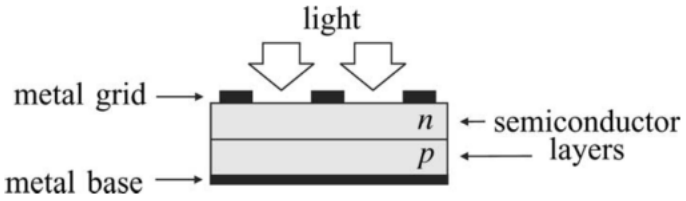

Figure 2.16: Physical structure of Photovoltaic Cell [17].

Due to illumination and the p-n junction electric field, both sides of the junction will

gather an excess of electrons (n-type side) and holes (p-type side) creating an imbalance in the junction, and if the junction is connected to an external electric circuit it will behave like a power supply in order to decrease the imbalance in the p-n junction.

The amount of light incident on the cell creates a near-proportional amount of current [18].

Photovoltaic cell panels or arrays are used as solar energy harness solutions; these can be deployed with different configurations, grid-connected system or self-consumption system, both of these configurations will be explained in section 2.5.4 and 2.5.5 respectively. These systems are composed by photovoltaic panels or arrays, and inverters; in some cases the use of batteries and a power controller for them is also required.

Based on the fact that photovoltaic elements produce DC electricity, to use that electricity in standard use appliances (AC) an inverter is needed. Several schemes for the application of these inverters are used nowadays, with the more traditional being a central inverter to which all the photovoltaic elements connect to and a more recent approach where micro-converters are installed in every photovoltaic element. There are several configurations where the use of an energy storing system is more efficient; as the amount of power collected by the photovoltaic element is heavily dependent on the amount of solar irradiance, the power source becomes variable for most electrical equipments to cope without an alternative power supply [19].

2.5.4 Grid-Connected Photovoltaic Systems

Grid-connected photovoltaic Systems are photovoltaic, illustrated in figure 2.17, power producing systems connected to the electricity grid allowing the system to feed any surplus of produced electrical energy in exchange for a tariff of the electricity companies. These systems allow a continuous use, creating a balance between hours of production where the surplus electricity is sold to the electricity grid and during periods of time where the solar irradiance is absent or insufficient resulting in the need of purchasing electricity from the electricity grid.

Grid-connected systems have the requirement of having inverters synchronized with the electricity grid. Additionaly, systems designated as grid interactive systems use inverters capable of using a power backup for self sustainability in the case of disconnection from the electricity grid [19].

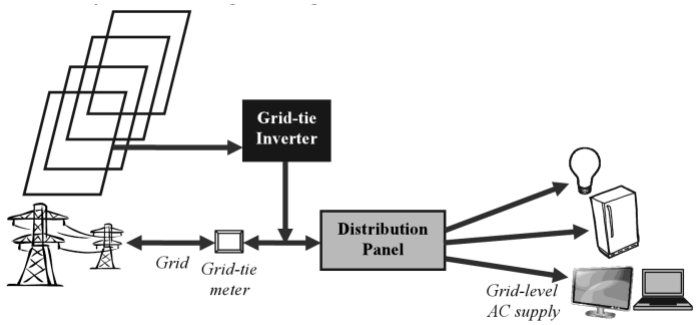

Figure 2.17: Example of a schematic of a grid-connected PV system [19].

2.5.5 Standalone Photovoltaic Systems

Off-grid or standalone photovoltaic systems are reliable solutions for systems independent from the electricity grid. Originally designed to provide a power source to systems where there were no other power source alternatives available. This type of system needs to scale accordingly with the peak power demand and it's highly dependable on the power backup system, as shown in figure 2.18.

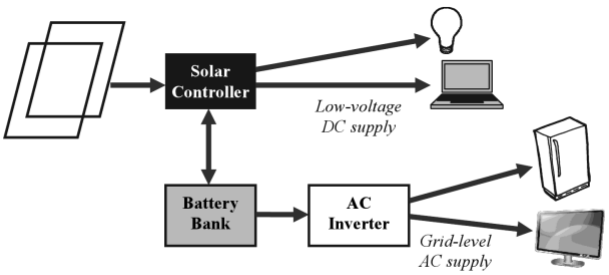

Figure 2.18: Example of a schematic of a standalone PV system [19].

2.6 Forecasting with Artificial Neural Networks

Forecasting and time series analysis are an active research field where the accuracy of time series forecasting is fundamental to decision processes, making forecasts with ANNs be extensively studied. In similarity to static models the forecasting models pattern recognition results from a series of inputs. Although the goal is prediction, the input series is designated as time series and is presented as a sequence of continuous, periodic and historical values with the purpose of developing the capability of forecasting future values [20].

Forecasting models generated from ANNs can vary in the duration of forecast. Forecasts can be short-term, medium-term and long-term, although the three specific forecast terms show very distinguished levels of detail in the forecasts, these are associated with the change in periodicity.

There are several forecasting methods. Traditional methods use statistical models for short-term forecasting, such as linear regression methods, exponential smoothing, Box-Jenkins approaches and Kalman filters. These techniques can be translated into linear models and as such only provide reasonable accuracy. On the other hand ANNs have been widely applied in the short-term load forecasting with their good capability for extracting complex and nonlinear patterns through a training process using historical data [21].

One type of ANN, with several applications such as function approximation, prediction and recognition is the Radial Basis Function Neural Network (RBFNN), previously mentioned in section (2.3.3). The usage of the RBF universal approximation property and its fast convergence speed related to the simplicity of the one hidden layer structure makes it a very efficient input-output predictor.

In general input-output relation of forecasting models is expressed through a time series of samples, being the output of the network the result of a series of time lagged inputs and output allowing one step predictions. These characteristics are mathematically described by equation (2.45):

$$\hat{y}(t+1) = f(y(t), y(t-1), \ldots, y(t-n_y)) \tag{2.45}$$

Where n_y is the maximal lag in the network input [21].

As in most ANNs, a common challenge is the optimization of the ANN structure in terms of the number of neurons, weights parameters and additionally for forecasting models the number of lags and their specific time delays. The parameter and structure identification approach of these systems is often done iteratively in an ad-hoc fashion focusing on the model parameters. Due to the huge number of combinatory possibilities of selection of inputs, model structure and model order the network designer is presented with the curse of dimensionality. The design criteria can also include multiple conflicting objectives, giving the model identification problem a multiobjective combinatorial optimization character. Evolutionary multiobjective optimisation algorithms are notably applicable to this problem evolving optimised model structures bound to pre-specified design criteria in acceptable computing time [22].

2.6.1 Multiobjective Evolutionary Algorithms

Multiobjective Evolutionary Algorithms (MOEAs) are one class of Evolutionary Algorithms that are defined by a set of procedures and operators inspired by the process of natural selection from the survival of the fittest in natural evolution with the purpose of finding the Pareto set solution of a given multiobjective problem. The solution candidates are labeled as individuals, being a set of these a population. Consequently, one iteration

of the MOEA has the target of improving the Pareto set for the population in the initial generation resulting into a better following generation. It is expected that after sufficient number of generations the population evolved achieving a satisfactory approximation to the Pareto front.

The Pareto set originates from the nature of multiobjective optimization. As soon as there are several contradicting objectives to be optimized simultaneously, there is no longer a single optimal solution but a whole set of possible solutions of equivalent quality. A solution could be best, worst and also indifferent to other solutions with respect to the objective values.

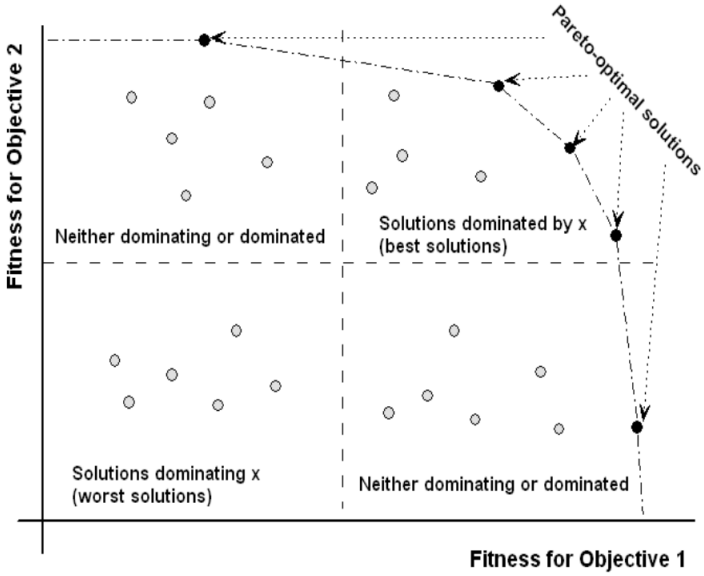

Figure 2.19: Concept of Pareto optimality [23].

As shown in figure 2.19, the Pareto-optimal solutions are characterized by not being dominated by any other solution in the search space, being localized on the Pareto frontier. Subsequently an entire set of such optimal trade-offs solutions is called a Pareto-optimal set or Pareto set [23].

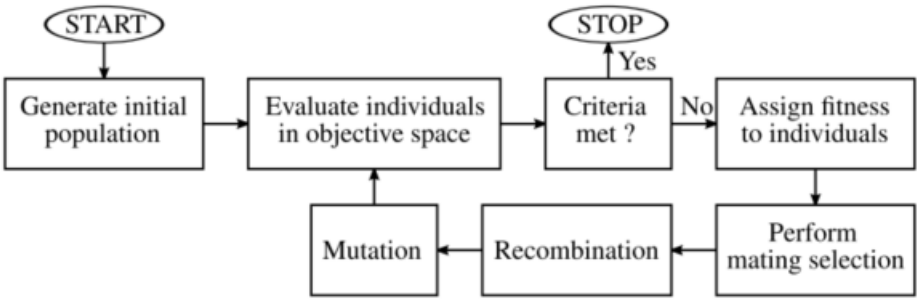

Figure 2.20: Generic flow of operation of most MOEAs [22].

In figure 2.20 it is shown the flow of operations of most MOEAs. During each iteration of the MOEA the population is evaluated for the specified objectives and a verification is made to confirm if the design criteria was met. If this is the case the MOEA algorithm stops and the designer obtains the individuals with the approximation to the Pareto front of the present generation. Alternatively the algorithm proceeds assigning a fitness value to each individual and mating them based on their fitness value. Subsequently each mated pair will generate two offspring individuals resulting from the recombination operator, thus creating the next generation. Lastly the mutation operator is applied to the individuals in the new generation before repeating the whole process [22].

3 Electric Power production forecast with Artificial Neural Networks

This chapter introduces the state of the art in the usage of Artificial Neural Networks to create non-linear models for forecasting and estimation of photovoltaic energy related subjects, with the purpose of efficiently managing electric power. Some state of the art applications for these non-linear models are also presented.

In [3] the authors present an alternative characterization method for the photovoltaic modules V-I curve. The authors intend to compare the characterization performance of a static model using Artificial neural networks based on data acquired with measured data and with the traditional Osterwald's method. The ANN architecture used was MLP with module temperature and solar irradiance as inputs. The ANN was trained with the Levenberg-Marquardt algorithm and early-stopping was employed. The authors conclude the artificial neural network model acquired better results than the traditional method, due to the fact that it has taken into account second order effects [3].

In [24] the authors propose a local online short-term forecasting model, for daily predictions of photovoltaic generated power using three distinct RBFNN as forecasting models classified according to the type of day, "sunny", "cloudy" and "rainy". According to the authors the classification process used a self-organized map (SOM) with inputs like solar irradiance, total cloud amount and low cloud amount. This classification allows the use of one of three RBF models (for each type of day) where the suggested forecast approach relies on using Numerical Weather Predictions (NWP) data, using the day of the month (t), the daily power output of the PV system ($P_s(t)$), the mean relative humidity $H(t+1)$, the mean daily wind speed $W(t+1)$, the mean daily solar irradiance $I(t+1)$ and the daily air temperature $T(t+1)$ producing as output an hourly forecast of generated power from the PV system for the following day. The authors present MAPE results between 8.29% and 10.80% for "sunny" days, 6.36% and 15.08% for "cloudy" days and 24.16% and 54.44% for the "rainy" days [24].

The authors in [25] propose a new methodology for local forecasting of daily global horizontal irradiance (GHI) based on ANN and spatial modeling. The model relates between US National Oceanic and Atmospheric Administration (NOAA) forecasts and real values of the GHI at the target location. The Multi Layer Perceptron neural network used was progressively trained with a database using back-propagation algorithm. Based on surrounding forecast data for the target region provided by NOAA the ANN model could estimate the GHI for such area. The authors conclude that the model developed in the French Alps and Cadarache located in south of France estimated daily GHI with satisfactory accuracy showing a relatively good agreement between measured and predicted values for the duration of the

test data [25].

In [26] the authors present a forecast based on classification of different days for a large-scale grid-connected photovoltaic power plant. Using data from Numerical Weather Prediction (NWP) and European Center for Medium-range Weather Forecasts (ECMWF) the data is classified into a type of day, each classification representing a different generated power forecasting AFFNN model, based on the expected generated power with a prediction step of 1 hour. It uses as inputs future values of in-plane solar irradiance at time $G_c(t+1)$, cell temperature at $T_c(t+1)$ and the present value of produced power $P(t)$ whereas the output is represented by the future value of produced power $P(t+1)$. They conclude that using a 1-year dataset split into three different subsets proved to be efficient . It is also noted the excellent ability of AFFNN-models to forecast the power produced by the PV plant, although it is recommended to update the bias and the weights of the ANN models developed periodically [26].

4 Methodology Applied

This chapter has the objective of explaining the procedure taken to create the forecasting system. It is based on a structure of a static model having as its inputs the prediction of a forecasting model for solar irradiance and a forecasting model for air temperature. Several static models were created, starting first with a seasonal approach and afterwards a year-long approach. The objective is a comparison between an year long model and the seasonal models, due to seasons across Earth being distinguishable when compared among them in terms of temperature and solar exposition. The models were created through the use of two types of Artificial Neural Networks, feed-forward neural networks (FFNN) and radial basis function neural network (RBFNN). The approaches to both types of ANN will be explained and compared.

4.1 Structure of the forecasting approach

In order to create this forecasting system a network structure was put in place. The structure considered was composed of two forecasting models for air temperature and solar irradiance which will be inputs for the static estimation model of solar generated power.

The forecasting models built for air temperature and solar irradiance are Radial Basis Function Networks. These models create the base structure for the forecasts of produced power. Based on the delayed values for air temperature and solar irradiance, these models will present a forecast with a time step of 5 minutes, for a total of 48 time steps i.e. a maximum forecast of 4 hours. Forecasts are sequenced, i.e., to create a forecast 2 time steps into the future, the first forecast time step is needed. These models are further described in section 4.6, followed by results and performance evaluations presented in section 5.4.

The methodology to create the static model is further described in the following sections, for the several stages of the model, from data preprocessing in section 4.3, data selection in section 4.4, training in section 4.5 to evaluation criteria in section 4.7.

Figure 4.1, represents the flowchart for the developed forecasting system where f represents the forecast time-step considered and d_1 to d_{na}or d_{ns} represent the interval of time delays used for the forecasting model being d_{na} and d_{ns} the maximum input delay for air temperature and solar irradiance respectively.

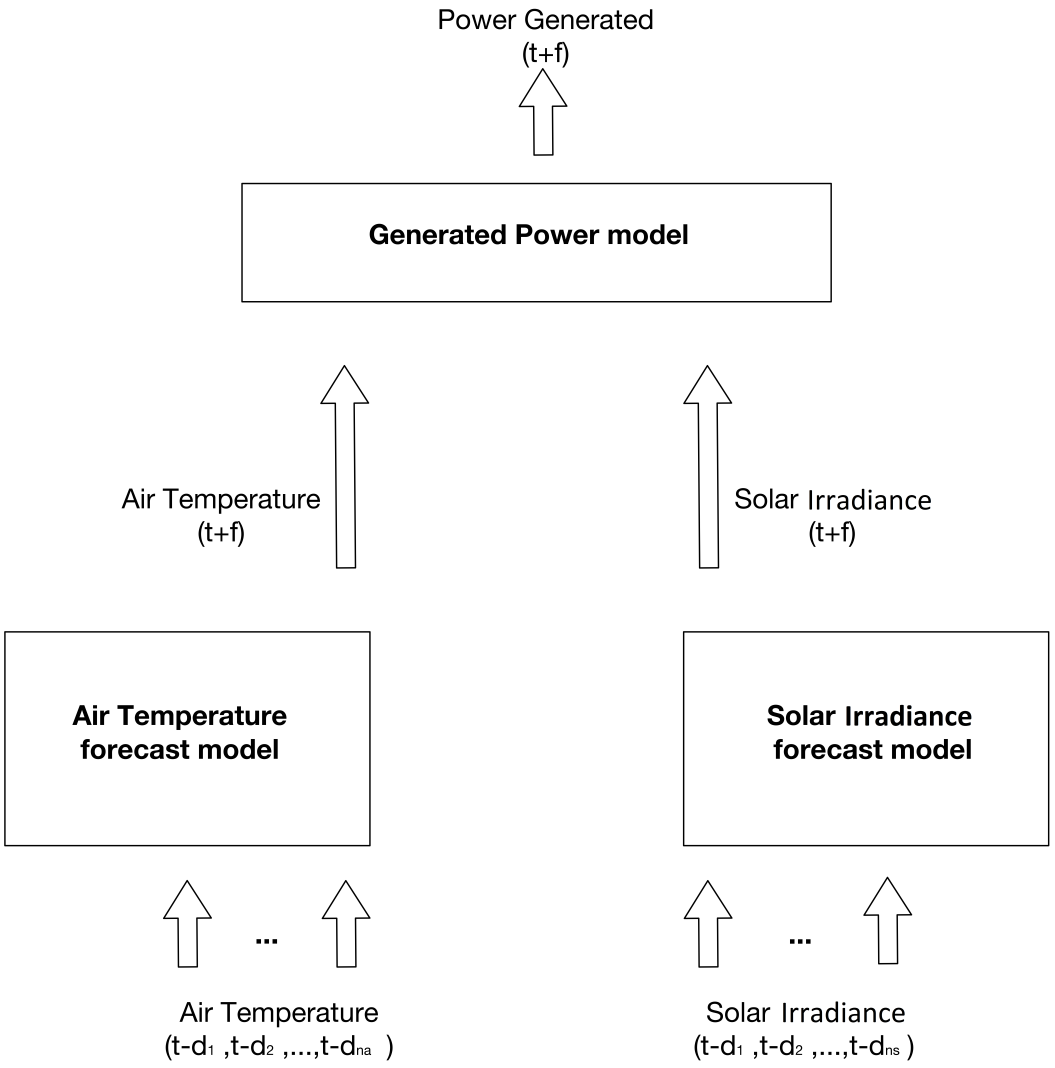

Figure 4.1: Flowchart of forecasting approach

4.2 Data Collection

To properly train the Artificial Neural Network a data set was used. This data set was collected from 20^{th} of June of 2014 until 29^{th} of June of 2015 from a pre-installed on-grid system in the University of Algarve, Faro, Portugal, with GPS coordinates $37^{o}01'40.0''N\ 7^{o}55'20.3''W$, with an Azimuth of 144^{o} and a tilt of 30^{o}, shown in figure 4.2. Data was recorded for a total of 343 days. The missing days corresponde to maintenance events of the photovoltaic system that prevented data acquisition. The actual number of days per season is shown in table 4.3.

Figure 4.2: Photograph of PV array

This pre-installed on-grid photovoltaic system was constructed and assembled by Rolear©. The system is composed by fourteen photovoltaic modules, in table 4.1 the standard test conditions for one module is presented, a PIKO Kostal inverter, an energy monitor and a programmable logical controller.

STC	SW 250
Max Power Wp	250
Open circuit voltage V	37,6
Max power voltage V	30,5
Open circuit current A	8,81
Max power current A	8,27
Solar irradiance W/m^2	1000
Temperature ^{o}C	25

Table 4.1: Standard Test Conditions (STC) characteristics of the Photovoltaic module

Standard test conditions (STC), are standard environmental conditions where the parametrization of the photovoltaic panels happens. The manufacturer of the photovoltaic panels, Solar World© establishes 250Wp (Watt peak) for a solar irradiance of 1000 W/m^2 at 25^{o}C, which means a total of 3500 Wp for the 14 panels in the photovoltaic system. The values acquired by the data exceed the maximum value expected for Watts peak production; this can be justified as air temperature and solar irradiance recorded values are above standard test conditions.

The data collected is composed of 31 parameters displayed in table 4.2 followed by the range of these parameters.

Data parameters	Minimum value	Maximum value
Year	2014	2015
Month	1	12
Day	1	31
Elapsed time since 6.00AM (seconds)	863	53254
Temperature of the solar module (^{o}C)	0.6	62
Air temperature (^{o}C)	1	46
Solar irradiance (W/m^2)	11	1482
AC Voltage in phase1 (V)	0	3168
AC Current in phase1 (A)	0	230
Power supplied in phase1 (W)	0	57350
AC Voltage in phase2 (V)	0	2960
AC Current in phase2 (A)	0	308
Power supplied in phase2 (W)	0	57350
AC Voltage in phase3 (V)	352	2832
AC Current in phase3 (A)	0	230
Power supplied in phase3 (W)	7	57350
DC voltage in the solar panel string (V)	0.3	448
DC current in the solar panel string (A)	0.3	9
DC power in the solar panel string (W)	0.2	3849
current in phase1 (A)	0.3	12
current in phase2 (A)	0.3	15
current in phase3 (A)	0.2	16
simple voltage in phase1 (V)	224.1	247
simple voltage in phase2 (V)	224	249
simple voltage in phase3 (V)	224.6	249
Line voltage between phase 1 and 2 (V)	388.4	429
Line voltage between phase 2 and 3 (V)	389.3	432
Line voltage between phase 3 and 1 (V)	388	429
three-fase real power (W)	-3.6	5
three-fase complex power (W)	-2.6	3
three-fase apparent power (W)	0	5

Table 4.2: Parameters from data collected

Data collection was aperiodic, with data from power generation peak hours being sampled more frequently than the initial and final hours of the day. The recorded time window for each day is also variable; the system records a time stamp in the form of elapsed seconds from 6 AM in the data, although the earliest record of data registered is from 863 seconds from 6AM (6:14 AM), as is shown in table 4.2. These parameters were collected until 9 PM of the same day or when the power generated from the photovoltaic panels reached a low

threshold in that same day. The recording time varies with the time of solar exposure. The average of collected hours, according with seasons is shown in table 4.3.

	Average		Number of days
	seconds from 6 AM	stopping time of data	
Summer	52409	20h 30min	92
Autumn	44062	18h 15min	91
Winter	42064	17h 40min	87
Spring	49606	19h 45 min	73
Yearly	47306	19h 8min	343

Table 4.3: Average time of collection of data per day separated according to seasons

4.3 Data Preprocessing

In order to achieve a proper learning, a preprocessing of all the data is required. The purpose of this preprocessing is to facilitate the modeling performance of the Artificial Neural Network. The procedures used will be described in the following sub-chapters. It is uncommon to have data that does not need some adjustment to improve the neural network efficiency. In this specific case, several adjustments were needed in order to properly use the data to train the network. These adjustments serve the purpose of correcting some incoherent data acquired, like the removal of outliers, correction of the sampling period and also the resizing of the amplitude of the data to allow the artificial neural network to learn more relevant patterns and minimize noise.

4.3.1 Outlier Removal

Due to errors in the data acquisition an outlier removal is needed. An outlier is considered as a data sample with values superior to the physical values allowed by the photovoltaic system with a certain error margin. The standard limiting values for power production for this specific photovoltaic panels were acquired in Standard Test Conditions (STC) represented in table 4.1, so a slack margin of 10% for the outlier is considered. Several conditions were put in place to remove the outliers. These conditions are the maximum range of DC voltage from the photovoltaic panels, maximum range of electric current, comparison between of recorded power acquired from the photovoltaic panels and calculated from the DC voltage and current readings, and disregard for all recorded data samples with negative values for current, DC voltage, air temperature sensor and solar irradiance.

4.3.2 Power Filter

As an optimization measure, a filter was applied to the generated power data. The application of this filter was due to incoherent and noisy data present in the samples registered in the beginning and the end of the daily data records. The filter changes the data original range of 0.2 to 3849 Watts to the modified range of 150 to 3849 Watts restricting the range of recorded power production data samples in 4%.

4.3.3 Data Normalization

To allow a proper learning of relevant patterns, important relationships become clearer when data is compressed to a given interval. This compression of data is designated as normalization and defines the lower and upper values of the data.

Being x a single sample of data from a data set where x_{min} and x_{max} represent the minimum and maximum of that given data set respectively, the compressed value is given by x_{norm}.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

In this specific case, data normalization was performed in the inputs of the neural networks, due to the two very distinct range of the inputs, solar irradiance and air temperature. The distinction of these two inputs from a neural network point of view comes from the amplitude of its values as shown in table 4.2.

4.3.4 Sampling Period

As it was mentioned in section 4.2, the original data acquired is aperiodic. While the lack of periodicity may not represent a problem in the development of the static model to estimate the power generated, a periodic sampling is necessary in order to validate the combination of forecasting models with the static model.

The used approach was a periodic sampling in the given data starting at 6AM, with a period of 300 seconds (5 minutes), where the absent samples where created by a linear interpolation of the immediate preceding value and the immediate following value; the exceeding samples that compromised the periodicity of data were removed. This approach was used for all 31 parameters of data, recreating a periodic approximated data set. The resulting daily data in the data set was composed by periodic samples separated by 300 seconds ranging from the time of the first sample to the time of the last sample.

4.4 Data Selection

Selection of data is a crucial aspect of neural networks when the goal is to accomplish a good generalization. To achieve this, one of the several tasks is the use of a great variety of input samples. In this specific case very distinguishable days are a reliable source of input samples for the learning process of the neural network. This classification of distinguishable or characteristic days will be explained further in section 4.4.1.

In this section the process and the criteria chosen to select different sample days from different seasons, in order to properly select data samples to be used in the training process will be presented. In order to maximize the input space a selection of characteristic days was employed. This type of selection is based on a classification of type of days, which translates into an analysis of mean value and variance of generated power by the photovoltaic panels. This analysis and selection of characteristic days is going to be further explained in section 4.4.1.

The sesonal and yearly model approachs will also be explained with further detail, with reference to the importance of maximizing the input space to enforce generalization.

4.4.1 Characteristic days

In order to create the classification of characteristic days, a division of the data set was created. Characteristic days are divided in three different types: sunny days, partial cloudy days and overcast cloudy days[26]. The main parameter taken in consideration for the classification was the generated power. This choice was based on the high correlation between generated power and solar irradiance. This correlation can be viewed in figures 4.3, 4.4 and 4.5.

Sunny days are characteristic for having a parabolic curve representing the amount of power generated along the day, represented in figure 4.3. The days represented in figure 4.3, are the 26^{th}, 27^{th} and 28^{th} of June, 2014.

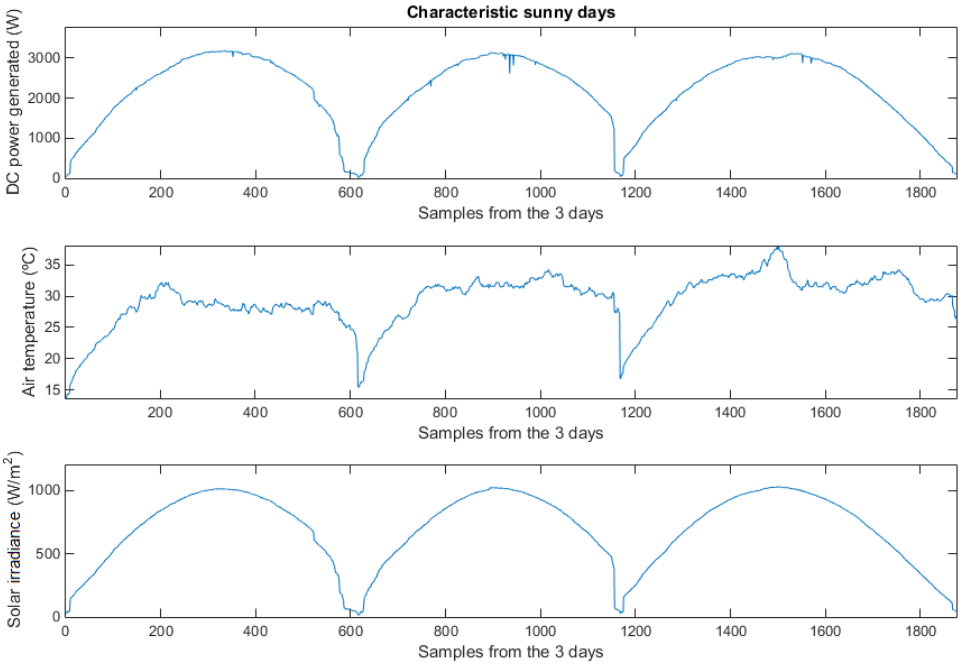

Figure 4.3: Characteristic sunny days

Partially clouded days are visually characterized from a parabolic shape similar to sunny days in generated power data, with minor interferences. The representation of exemplary partially clouded days can be seen in figure 4.4. The days represented in figure 4.4 are the 18^{th}, 23^{th} and 24^{th} of October 2014.

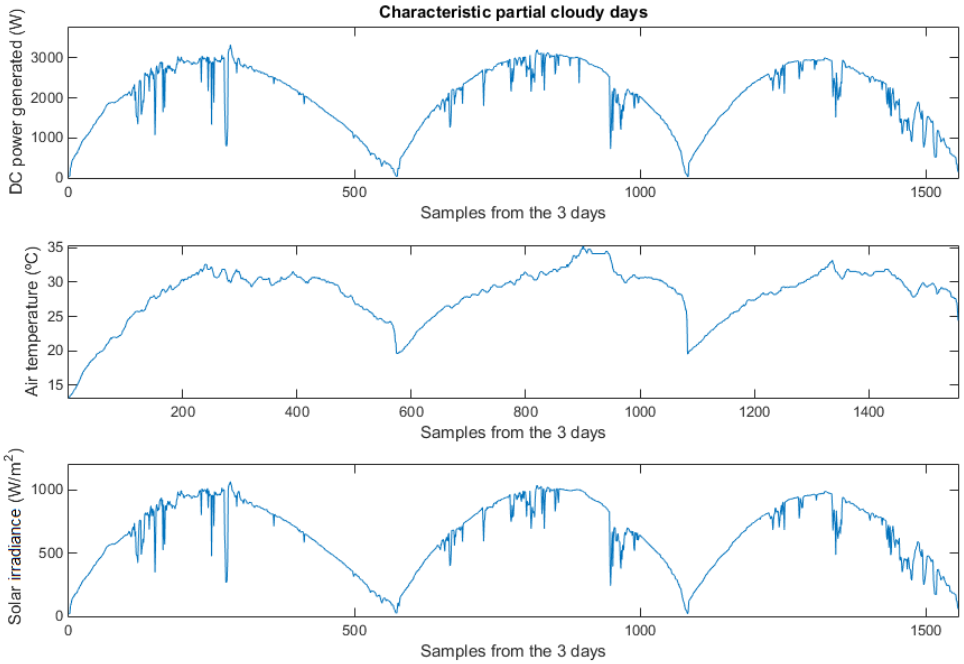

Figure 4.4: Characteristic partially clouded days

Overcast cloudy days represent days that are visually characterized from their lack of parabolic shape in the generated power and by their inconsistency along the day. These are represented in figure 4.5, dating from 23^{th}, 27^{th} and 29^{th} of December 2014.

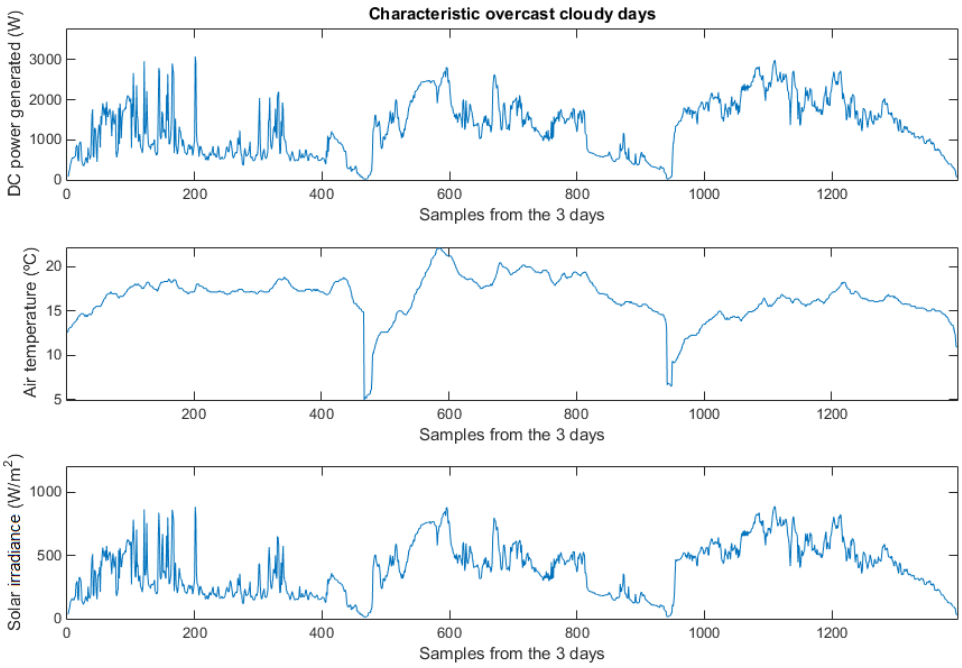

Figure 4.5: Characteristic overcast cloudy days

To mathematically differentiate these characteristic days an approach based on the average generated power and variance was created. This approach was created having in mind the data set of 343 days, allowing a proper base separation of characteristic days, creating a list of characteristic days that stand out in measured parameters. The most significant characteristic days from all three types (sunny, partially clouded and overcast clouded) were identified, and then chosen to maximize the input space.

The process of parsing starts with the measurement of the average generated power of the data set. As shown in table 4.4, characteristic days classified as sunny days stand out with a larger average. Due to different seasons having a different maximum generated power value, a technique based on fixed thresholds could not be used. To maximize this distinction between sunny days and any other days the data set is separated in half, where sampled days with generated power above average are considered potential sunny days, and from these only the ones with higher average will be indeed considered sunny days and consequently used for the learning procedure of the neural network.

The part that is discarded from being sunny days, are the potential clouded days. To create a distinction from partially clouded days and overcast clouded days an analysis was made in the variance of generated power. Knowing that obstructed solar panels produce less

energy than unobstructed panels a distinction is made in average power generated, although, to properly differentiate partially clouded days from overcast clouded days there are major differences in variance as shown in table 4.4. This change in variance is related to time period of solar obstruction created by clouds, where fewer clouds will be represented by high values of variance, and consistent cloudy days or overcast cloudy days will be represented by lower values of variance.

Table 4.4 represents the approach to distinguish characteristic days, using the days used in the previews example in figures 4.3, 4.4 and 4.5.

Characteristic days	Sunny	Partially clouded	Overcast clouded
Mean of generated power (W)	2191.9	2031.8	1307.3
Generated power variance	763390	657480	482710

Table 4.4: Mean and Variance of generated power in the three different characteristic days

This process, due to the use of the average generated power to separate the data set in two different subsets, was only used for the seasonal approach, which will be further explained in section 4.4.2.

For consistency a different method was used to select characteristic days when analyzing the models performance in section 5.4. This method relies on using the mathematical model described in section 2.5.2 for comparison with the acquired data. In figure 4.6 the comparison is shown for three different days.

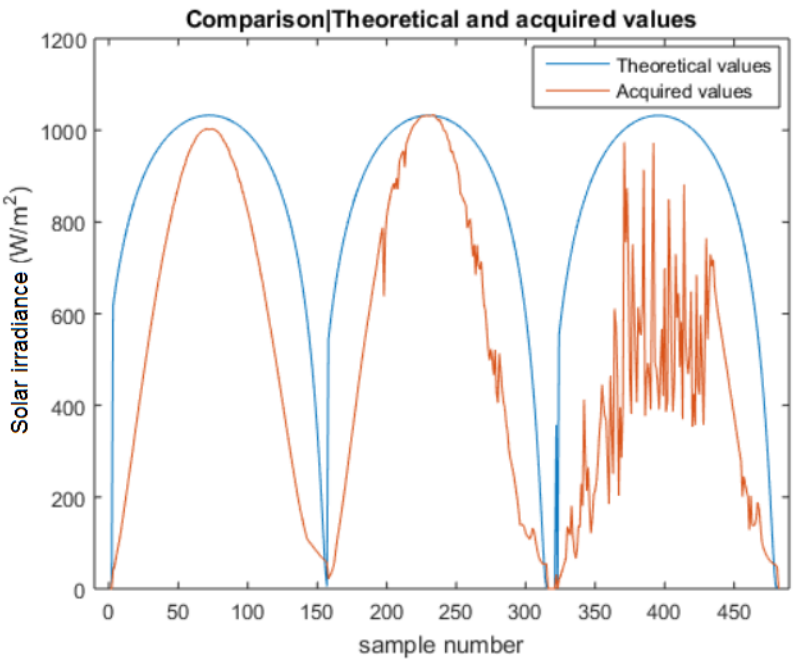

Figure 4.6: Comparison between theoretical and measured solar irradiance values.

This comparison was done with daily data, allowing a categorization of all available days based on the mean relative absolute error (MRAE) shown in equation 4.2, being N_d the number of daily samples, t_{SR} the theoretical values of solar irradiance, r_{SR} the acquired solar irradiance values from real data.

$$MRAE = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{|t_{SR}(i) - d_{SR}(i)|}{t_{SR}(i)} \tag{4.2}$$

This creates a more accurate classification of available characteristic days allowing an overview of the quantity of different characteristic days. In figure 4.7 the distribution of days according to MRAE is shown. The largest concentration of days is bounded between an MRAE of 0.4 and 0.6. From figure 4.7, days with a MRAE inferior to 0.4 were classified as sunny days, days with an MRAE between 0.4 and 0.6 were classified as parcially clouded days and days with a MRAE superior to 0.6 were considered overcast clouded days.

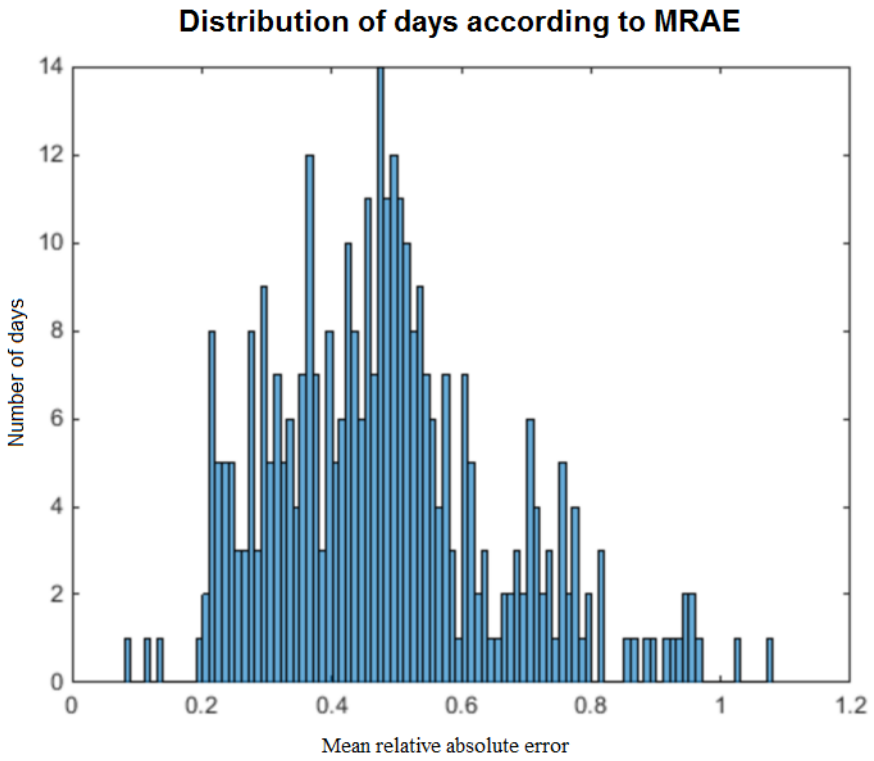

Figure 4.7: Characteristic day distribution according to MRAE.

4.4.2 Seasonal Approach

The creation of these four seasonal models represent the days contained in those seasons, differentiated by calendar date and not by the characteristics of those seasons. The calendar limits of the data, maximum, minimum and mean of the network inputs of the seasonal models are represented in table 4.5.

Seasons		Summer	Autumn	Winter	Spring
Start date		21/06/2014	23/09/2014	23/12/2014	21/03/2015
End date		23/09/2014	22/12/2014	20/03/2015	29/06/2015
Air Temperature (ºC)	Max	40.7	34.8	28.5	36.5
	Mean	28.6	21.4	17.4	25.2
	Min	14.7	9.4	1.15	7.25
Solar Irradiance (W/m^2)	Max	1322	1037	1142	1353
	Mean	515	314.4	434	535.5
	Min	18	14	15	15.4

Table 4.5: Seasonal Characteristics

Although not directly represented in data there are significant characteristics associated with seasons that will influence the power generated by the photovoltaic panels. Among such

characteristics, solar obstruction is a characteristic that can be correlated with momentary low values in the data representing solar irradiance and generated power. The creation of seasonal models is intended to compare their performance against an overall yearly model. Given that specific seasons have different sets of data, and certain characteristics based on weather events interpolated by the model being more frequent in certain seasons, a comparison is needed to classify the yearly model as a reliable model.

The selection of the training data for seasonal models was made using the characteristic days approach mentioned in section 4.4.1. Adding to this selection, due to the interpolation capability of the neural network and to maximize the input space, the days with peak values of generated power in each season are also used in the training data. The respective selection of days used for every given season is shown in table 4.6 followed by figure 4.8, 4.9, 4.10 and 4.11 representing the seasonal training sets data separated by generated power, air temperature and solar irradiance.

Spring			**Summer**		
Day	Month	Year	Day	Month	Year
21	March	2014	22	June	2014
24	March	2014	6	July	2014
31	March	2014	19	July	2014
2	May	2014	7	August	2014
3	May	2014	12	August	2014
6	May	2014	26	August	2014
21	May	2014	27	August	2014
1	June	2014	10	September	2014
13	June	2014	11	September	2014
14	June	2014	13	September	2014
15	June	2014	14	September	2014
17	June	2014	15	September	2014
22	June	2014	18	September	2014
Autumn			**Winter**		
Day	Month	Year	Day	Month	Year
30	September	2014	23	December	2014
1	October	2014	24	December	2014
7	October	2014	8	January	2015
9	October	2014	20	January	2015
10	October	2014	29	January	2015
8	November	2014	7	February	2015
17	November	2014	8	February	2015
19	November	2014	15	February	2015
28	November	2014	16	February	2015
2	December	2014	19	February	2015
13	December	2014	20	February	2015
15	December	2014	21	February	2015
16	December	2014	24	February	2015

Table 4.6: Calendar date for days of the training set for the individual seasonal training sets.

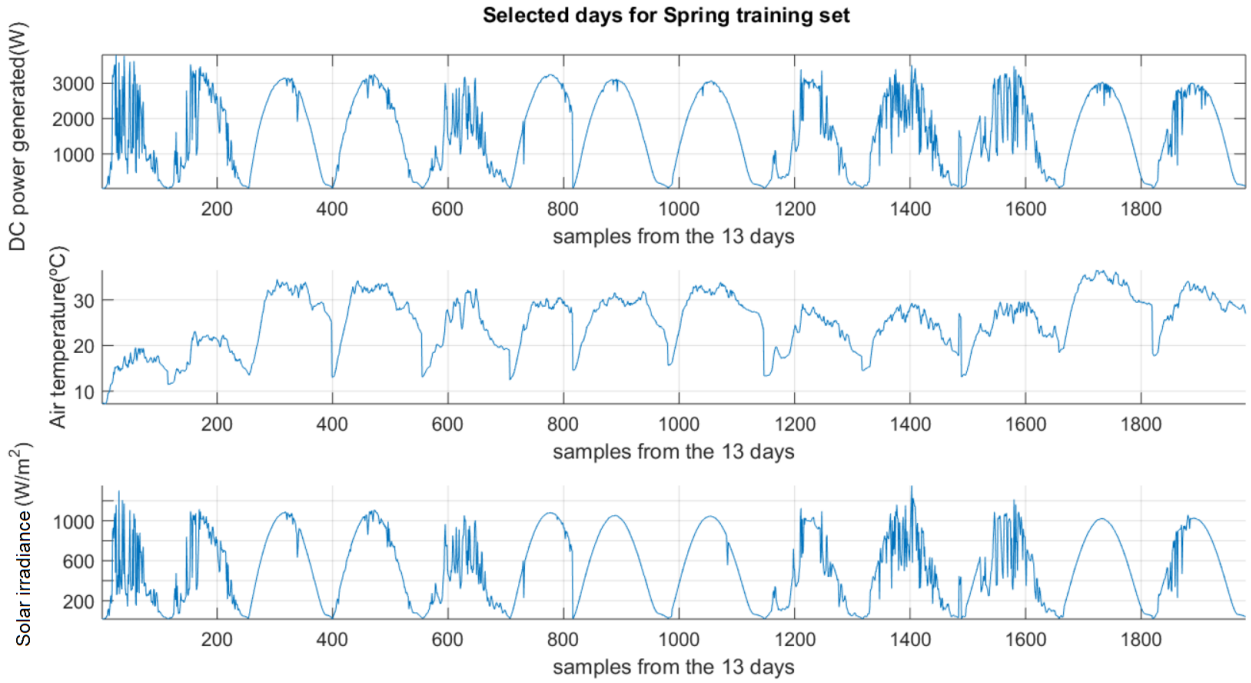

Figure 4.8: Training set data for Spring model exhibiting generated power, air temperature and solar irradiance.

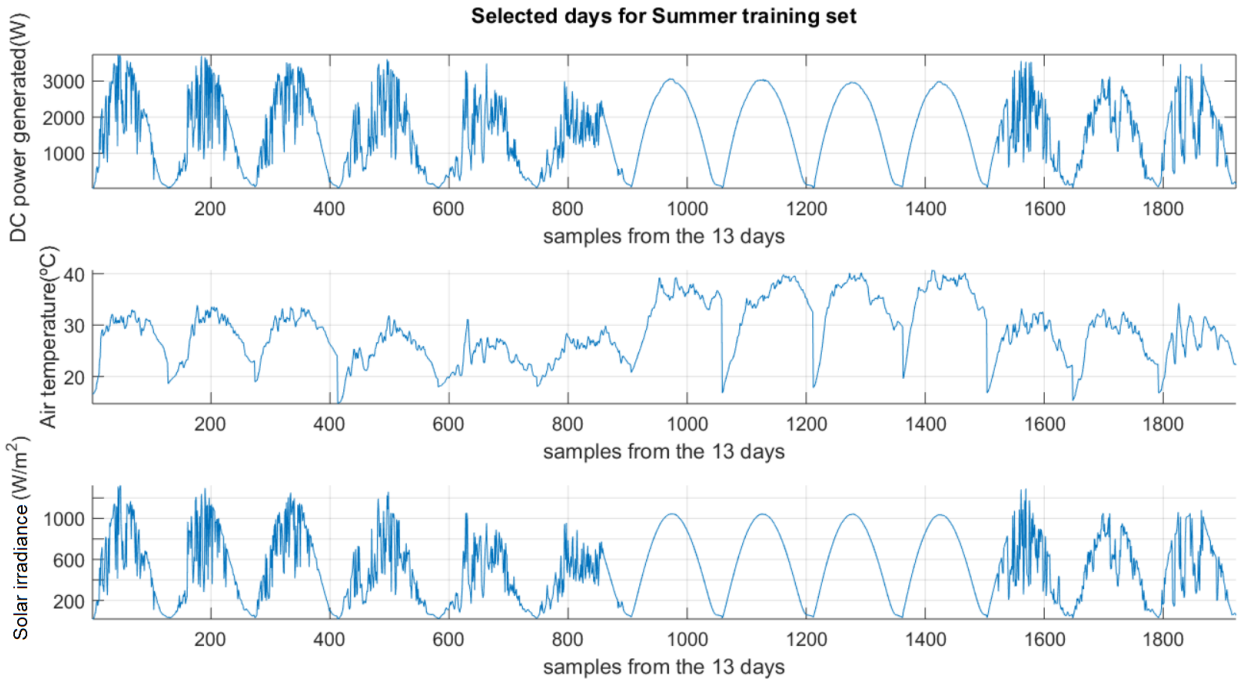

Figure 4.9: Training set data for Summer model exhibiting generated power, air temperature and solar irradiance.

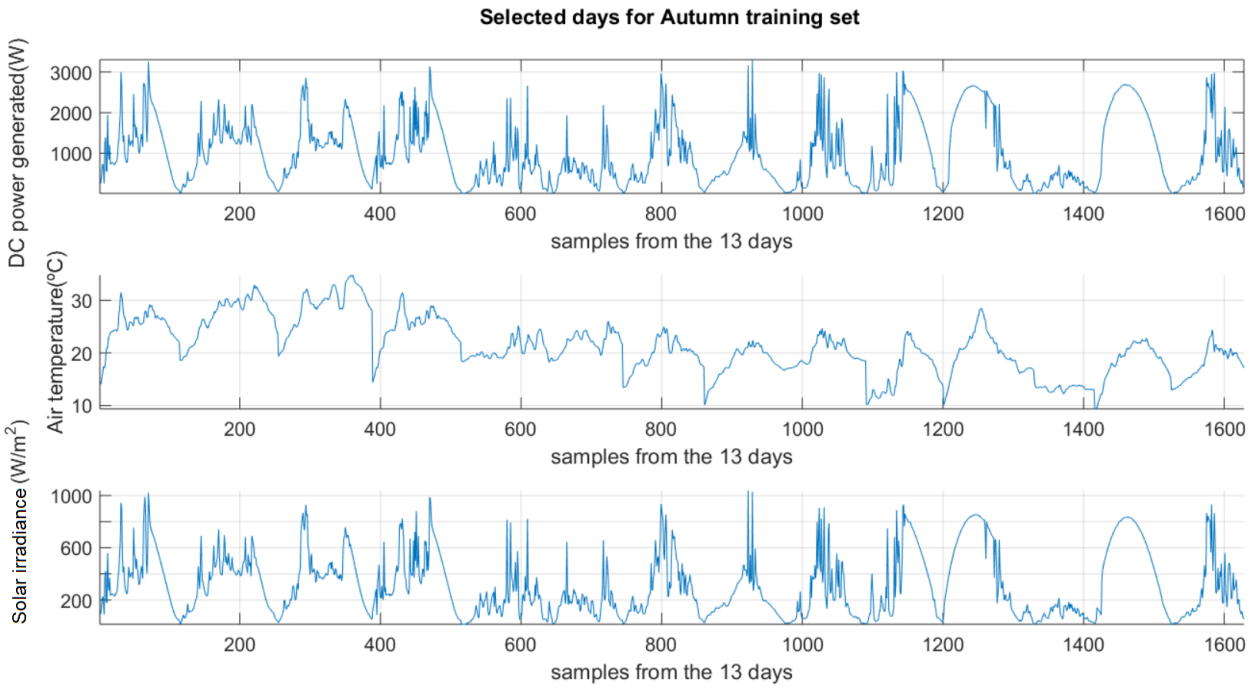

Figure 4.10: Training set data for Autumn model exhibiting generated power, air temperature and solar irradiance.

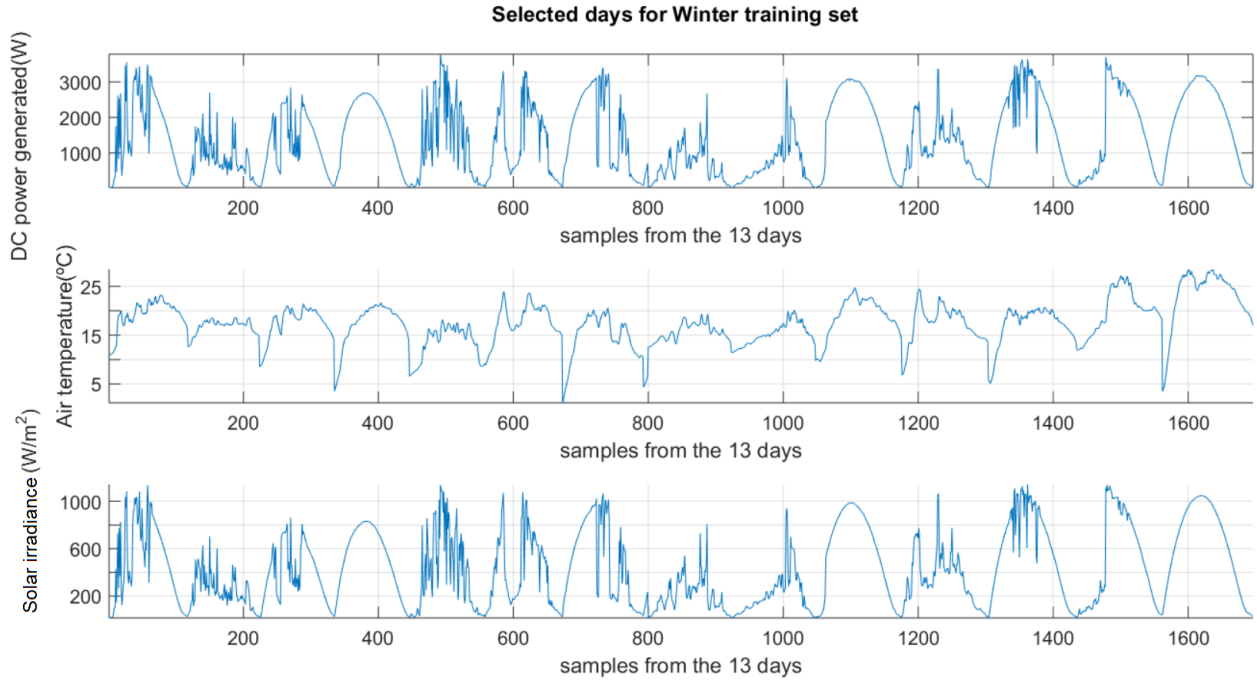

Figure 4.11: Training set data for Winter model exhibiting generated power, air temperature and solar irradiance.

4.4.3 Yearly Approach

In the yearly model approach the data selection process was different. The reason for this is because the application of the seasonal data selection process to the global dataset would wrongly distinguish seasonal patterns as specific characteristic days. That process is based on the separation of days with higher average of generated power; as such the chosen days would mostly be summer days, which represent the days with higher temperature and higher solar irradiance, by accepting this, the input space would not be maximized.

The approach applied was a selection of four specific days from the four different seasons. The criteria used for this selection was the choice of three distinct characteristic days from each season complemented by the day with the maximum generated power from each respective season. The training selection is shown in table 4.7, along with their data representing generated power, air temperature and solar irradiance shown in figure 4.12.

Yearly data					
Spring			**Summer**		
Day	Month	Year	Day	Month	Year
24	April	2015	19	July	2014
3	May	2015	7	August	2014
21	May	2015	13	September	2014
14	June	2015	15	September	2014
Autumn			**Winter**		
Day	Month	Year	Day	Month	Year
1	October	2014	23	December	2014
7	October	2014	28	January	2015
10	October	2014	21	February	2015
15	December	2014	27	February	2015

Table 4.7: Calendar date for days of the training set for yearly model.

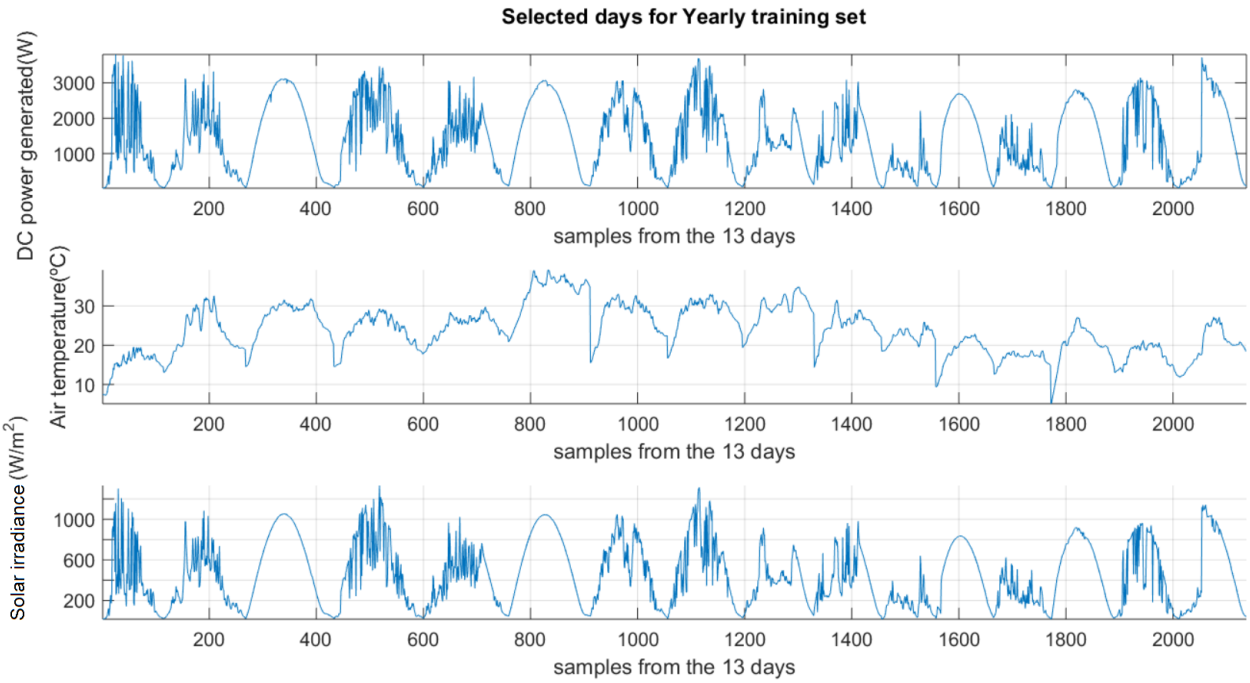

Figure 4.12: Training set data for yearly model exhibiting generated power, air temperature and solar irradiance.

4.5 Training

This section introduces the training methodology applied to both considered neural networks architectures, MLPs and RBFs. Although both networks have a similar training procedure, there are several steps which greatly differ based on their architecture. There are several conditions that change the capacity of generalization and the learning capability of these networks such as initial parameters, number of free parameters and data used in training. Training set was chosen as described in section 4.4. In the following sections the training procedure will be described for the two types of neural networks used and also the process of selection for the validation and testing sets.

4.5.1 Multi-Layered Perceptron learning procedure

The training procedure for Multi-Layered Perceptrons has some optimization procedures based on a comparison between generated models. To select the proper topology of the network, a window of acceptable topologies was created. This window allows the algorithm to try several configurations, under the window of available configurations. The window range comprises in the number of hidden layers and the allowable number of neurons per hidden layer. Knowing that networks with fewer neurons usually achieve a better generalization, the window ranged from two hidden layers with one neuron each, to two hidden layers with four

neurons each. In order to accurately choose the best model, a common test set needs to be used. The creation of this test set is described in section 4.5.3. The selected model will be the one with lowest MSE for test set.

To avoid the problem of local minima, each network is trained with random initial conditions and tested fifteen times. The choice of fifteen repetitions of the procedure was based on a time consumption and success rate.

The training algorithm was MATLAB® Levenberg-Marquardt implementation, based on its stability and efficiency. To allow a proper training with early stopping property, cross-validation was used , dividing further the data in a training subset with 70% of the original training data and a validation subset with the remaining 30%, allowing a maximum of 6 increasing values of the error in the validation before interrupting training. The data separation between these two sets was random, without any order, to avoid separation of characteristic days which would penalize training performance. The learning algorithm will stop after 300 epochs if it is not interrupted by the early-stoping condition. From the available topology of two hidden layers, the transfer functions used were hyperbolic tangent transfer function for the hidden layers and linear transfer function for the output layer.

4.5.2 Radial Basis Function learning procedure

In the Radial Basis Function learning procedure, the preprocessing of data is the same. However in RBFs the free parameters are the location of the centers and spread of the given radial functions requiring an initial selection.

There are training schemes that allow a proper positioning of these centers a-priori. In this specific case, an approach using k-means clustering algorithm was used to determine the initial position of the centers. This is followed by the training algorithm, where the Levenberg-Marquardt was selected based on the same criteria has the MLPs, stability and efficiency. In a similar fashion to the MLP network the data selected for learning procedure was further divided into two subsets of data, one for training with 70% of the original data set and one for validation with 30%. Sample division was random respecting only the proportions of the division.

To determine an optimum number of centers, the learning process was applied from two centers up to ten centers for 15 trials, only stopping the training if the limit of 300 epochs is met or stopped by the cross-validation process. The repetition of trials is very significant since there is no proper form to consider the performance of the RBF network topology before applying the learning procedure, and bad initial parameters can have a negative effect in network performance.

4.5.3 Testing and Validation sets

In order to avoid over-training, the cross-validation procedure uses a validation subset acquired from 30% of the respective set from the respective seasonal models or yearly model, described in sections 4.5.1 and 4.5.2. Due to such a variable number of available topologies, to accurately confirm if the neural network has an efficient learning process employing generalization and to determine the best topology, a simulation is applied after the training process. This simulation is a test to measure the error of the estimation and consequently the quality of the learning process.

In order to assess the quality of the trained neural network, a test set must be chosen. Testing the network ability to generalize is only possible by presenting the neural network with different samples than the ones presented in the training set. To select a test set that properly identifies the best topology for an estimation of data sets for entire seasons or a complete year and being the objective of training and topology selection a good generalization, the selection criteria for the test set was all the data available not used in training for a specific season or year, depending on which model was used.

As it is shown in table 4.8, each season has a different number of days. For the test set, all the days from each data set, yearly or seasonal, were selected, after removing the days used for training. The number of days used for the test sets is shown in table 4.8.

Data sets	Number of days in seasonal data set	Number of days in test set
Spring	73	60
Summer	94	81
Autumn	91	78
Winter	87	74
Year	343	327

Table 4.8: Size of seasonal models test sets

4.6 Forecasting Models

As explained in section 2.6, forecasting depends on certain characteristics to perform efficiently. In this section those characteristics will be presented and explained. The presented forecasting models are radial basis function neural networks designed for prediction with several lags depending on the model and selected through a multi-objective evolutionary algorithm (MOEA) reducing the RMSE across all time steps within the prediction horizon.

Due to the usage of continuous data being required for an efficient forecast, and being the data collected limited to hours with significant solar exposure as shown in table 4.3, there was the need to add data corresponding for the night time which resulted in an arti-

ficial null solar irradiance properly flagged in the data set and taken in consideration in the training procedure. This artificial data implementation was only used in solar irradiance; for temperature, on the other hand, this could not be applied.

The MOEA used was developed in [22] with the purpose of introducing an alternative when designing forecasting and classification ANNs simplifying the network design process. The algorithm was computed using a computer cluster in the Electronics and Informatics Engineering Department (DEEI) due to its computational demand and computation time. The generic approach of this algorithm was explained in section 2.6.1 and is going to be further described in section 4.6.1. Also, in the same section the resulting forecasting model expressions will be presented.

4.6.1 MOEA approach

In multi-objective evolutionary algorithms the individuals are defined by a set of variables, which characterize them in their score or Pareto Front. These characteristics are contained in a structure denoted as chromosome which is different for every individual in the population. In this specific algortihm chromosomes are strings of integers composed by the information of number of neurons, u, for the network design confined within the range $[u_m < u < u_M]$ and the indices values, λ_j, from the features subset, $\mathbf{f}_{k\lambda_j}$, selected from the data set $\mathbf{F}$. The usage of feature subsets is due to the size of the data set $\mathbf{F}$ being larger than the prescribed maximum, the input vector dimension maximum range is confined between $[d_m, d_M]$. As a result of this dimension confinement the input data set $\mathbf{X}$ is created from a selection of d columns of $\mathbf{F}$. Creating multiple input vectors of the appropriate size defined by $\mathbf{x}_k$ on expression 4.3

$$\mathbf{x}_k = [x_{k1}, x_{k2}, x_{k3}, ..., x_{kd}] = [f_{k\lambda_1}, f_{k\lambda_1}, ..., f_{k\lambda_d}] \tag{4.3}$$

The input space used is composed of a_0 output delayed terms with a maximum lag of τ_y.

In order to effectively identify the individual performance, each individual is assigned a scalar value, the fitness. Being the evolutionary algorithm efficiency dependent of its chromosome variability, several operators and procedures are used to ensure chromosome fitness and variety converges in the progression of the generations. These procedures are designated as mating procedure and mutation being applied in the process of creating a succeeding generation.

The mating procedure is based on the idea that individuals with a better fitness have an increased chance of having more copies in the mating pool, consequently, individuals with lower fitness have less or no copies in the mating pool creating an adjustment for the new generation with individuals which are more likely to have a better fitness value. With a given probability, the matting process uses the recombination operator using two individuals

to generate two offspring. This is a two part process designated by full identity preserving crossover, where the first part is composed of the re-ordering of the chromosomes, adjusting common terms to the left-most positions and the remaining terms are shuffled, isolating common terms avoids duplicates in the offspring. After re-ordering and shuffling a random point is then chosen, the crossover point, and the elements to its right are exchanged [22].

The used algorithm applies mutation in different forms for the distinct parts of the chromosome. Firstly, the number of hidden neurons is mutated with a given probability, following boundaries described previously, either by adding or subtracting one neuron to the existing quantity. Secondly, the following input terms in the chromosome are mutated for a given probability in three different operations: replacement, addition or deletion. In the first operation each term is tested and is either deleted or replaced by another term from the same set not present in the chromosome. Deletion can only occur if the chromosome as more terms than the minimum specified earlier, d_m. Afterwards, if the chromosome is not in its maximum range, d_M, one term may be added from the terms of the same set not present in the chromosome.

After the completion of the described procedures, the MOEA algorithm proceeds to the subsequent evaluation step repeating the cycle for the new generation of individuals. This process is cyclic and iterative only stopping to certain conditions. These conditions can be defined as a predefined number of generations, if an individual with satisfactory performance is found, or through the stagnation of the population, which is checked by the absence of change in the Pareto front approximation for a specified number of consecutive generations. Figure 4.13 represents the model design cycle separated in three different groups associated with the difference in colors: problem definition, solution generation and result analysis. $\mathbf{F}^p$ and $\mathbf{F}^v$ represent two partitions of the input search space, the first is intended for ANN training procedure and the second to validate the results obtained by the Pareto set of individuals, respectively.

$$\mu(\mathbf{F}, \Lambda, u, \mathbf{w}) = [\mu^p, \mu^s] \tag{4.4}$$

$$\mu^p = \left[\mu_1^p, \mu_2^p, \ldots, \mu_{v_1}^p\right] \tag{4.5}$$

$$\mu^s = \left[\mu_1^s, \mu_2^s, \ldots, \mu_{v_2}^s\right] \tag{4.6}$$

$$\Lambda = [\lambda_1, \lambda_2, \ldots . \lambda_d] \tag{4.7}$$

As shown in expression 4.4 a dependence to μ from several variables is created, being these:

$\mathbf{F}$, the vector Λ that identifies the indices of columns of $\mathbf{F}$ that define the input features considered , the number of neurons, u, and the ANN parameter vector, $\mathbf{w}$. The complementary parameters μ^p and $\mu^s$are vectors contain the quality measures v_1 and v_2, respectively. μ^p represents the mapping obtained by the parameters computed through the root-mean-square error on the training and generalisation testing data sets. As a restriction, a vector given by the 2-norm of the linear parameters was employed to guarantee good numerical properties and parameter convergence. Additionally, this vector also acts as a penalty term for the complexity of the model. Regarding μ^s, the component of μ related to model structure selection and specific model application, one objective was considered expressing the final goal of the model application: establishing the prediction profile for the models within an horizon of 4 hours optimizing the models prediction error taken from the multi-step model simulation over the prediction horizon ph.

Assuming a given simulation data set, $\mathbf{D}$, has p data points and for each data point the model is used to make predictions up to ph steps ahead. The error matrix is formed in expression 4.8.

$$\mathbf{E}(\mathbf{D}, ph) = \begin{pmatrix} e[1,1] & e[1,2] & \dots & e[1,ph] \\ e[2,1] & e[2,2] & \dots & e[2,ph] \\ \vdots & \vdots & \ddots & \vdots \\ e[p-ph,1] & e[p-ph,2] & \dots & e[p-ph,ph] \end{pmatrix} \tag{4.8}$$

being $e[i,j]$ the model prediction error taken from the instant i of $\mathbf{D}$, at step j within the prediction horizon. Denoting the root-mean-square function operating over the i^{th} column of the argument matrix by $\rho(\cdot\,, i)$ then the prediction performance measure is defined by expression 4.9.

$$\varepsilon(\mathbf{D}, ph) = \sum_{i=1}^{ph} \rho(\mathbf{E}(\mathbf{D}, ph), i) \tag{4.9}$$

which translates into the summed root-mean-square of the columns of $\mathbf{E}$. Becoming this way the single objective in μ^s the minimization of $\varepsilon(\mathbf{D}^s, ph)$ being ph the 48 multi-step prediction horizon and $\mathbf{D}^s$ a simulation data set [22].

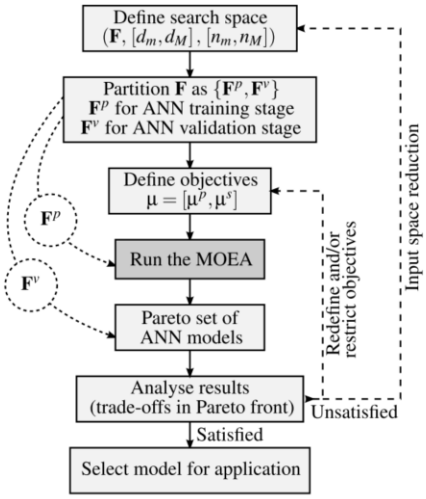

Figure 4.13: MOEA Model design cycle [22].

When it is required to repeat the MOEA process a problem redefinition should be applied. This results in two major actions: redefining the input space by adding or removing one or more features (variables and lagged input terms) and improving the trade-off surface coverage by changing objectives or redefining goals. These actions can result in a reduction of the number of input terms and limits the range for the number of neurons in face of the results obtained in a single run. This narrows the search space in a subsequent MOEA run, possibly achieving a faster convergence and better approximation on the Pareto front.

The training procedure of the individual RBF networks, in similar fashion to the networks described in section 4.5.2 uses the Levenberg Marquardt training algorithm, using a-priori a clustering algorithm spreading the centers in distinct regions of the input feature space and calculating the center spread according to expression 4.10 taking in consideration the maximum euclidean distance, z^{max}, between centers, c_i, and the number of neurons, u. In order to avoid over-training, the training procedure uses a cross-validation process splitting further the training input space $\mathbf{F}^p$ into two data sets, the training data set, $\mathbf{F}^t$, and the generalization data set, $\mathbf{F}^g$ [22].

$$\sigma_i = \frac{z^{max}}{\sqrt{2n}} \tag{4.10}$$

The resulting forecasting models are described by the following expressions being the selected air temperature forecasting model presented in expression 4.11 with 13 centers and 9 lags.

$$y(k) = f(y(k-1), y(k-2), y(k-3), y(k-4), y(k-5), y(k-7), \\ y(k-8), y(k-17), y(k-24)) \quad (4.11)$$

The Solar irradiance forecasting model uses 13 centers and 10 lag samples as shown in expression 4.12.

$$y(k) = f(y(k-1), y(k-12), y(k-21), y(k-24), y(k-27), y(k-39), \\ y(k-41), y(k-42), y(k-44), y(k-290)) \quad (4.12)$$

4.7 Evaluation criteria

The criteria used for evaluation of the static models relies on the MSE for the test set. The lowest mean squared error value for the test after training ensures the topology with the best generalization capability. However, to determine the best topology using a trial and error approach it should be noted that networks with more neurons are more likely to be overparameterized as discussed in section 2.4.1.

In chapter 5, the topologies selected will be based on the lowest MSE, although when several topologies have a similar mean squared error, topologies with less neurons were chosen to avoid overparameterization.

Forecasting models represented in section 4.6 differ in evaluation criteria from static models in the sense that performance in forecasting models is more consistently measured with RMSE. Additionally, the NRMSE criterion was also used.

5 Results

In this chapter the results for the topology selection of the neural networks used and the performance for the models in consideration is presented. The models presented are the seasonal models and annual models implemented with MLP and RBF neural networks, which will be compared in their capability of generalization based on the mean squared error. Note that these models estimate the power produced by the PV installation, whose inputs are the measured solar irradiance and atmospheric temperature. This chapter will also present results for the forecasting models for air temperature, solar irradiance and for the global model using the three models resulting in the forecast of generated power of the photovoltaic array, based on the forecast of air temperature and solar irradiance.

This section is separated into evaluation of the static model, and forecasting model. The static models evaluation are first differentiated in time scale (Seasonal and Yearly) and then in the type of network used for the given results.

The performance of the forecasting models is evaluated through a comparison between forecasting capability in the different seasonal data sets, annual data set and compared across different characteristic days.

5.1 Seasonal models

5.1.1 Multi-Layer Perceptrons Results

As mentioned in section 4.5.3, to select the best topology for the MLP network several topologies were tested being the criteria of selection the minimum MSE when tested with the respective test data set. The topologies presented in figure 5.1 and table 5.1 show the number of hidden layers and respective number of hidden neurons in those layers for the selected models. As is shown in table 5.1, the models selected have different topologies being the Summer and Autumn models the ones with the lowest number of hidden neurons, having two neurons in the two hidden layers, resulting in the models with less complexity.

Figure 5.1 illustrates the value of the MSE with the range of topologies for the four color coded seasons. Table 5.1 represents the selected topologies with the respective MSE and MAPE.

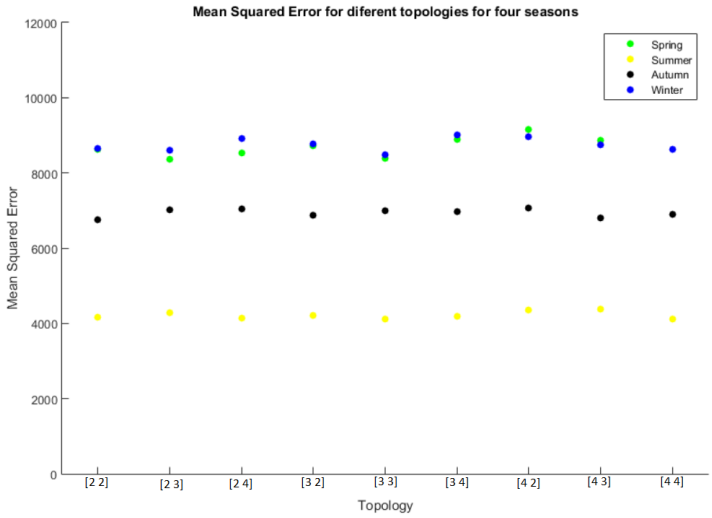

Figure 5.1: Variation of MSE across tested topologies for MLP seasonal models.

Seasonal models	Topology	MSE	MAPE (%)
Spring	[2 4]	8356	4.1
Summer	[2 2]	4153	2.2
Autumn	[2 2]	6753	3.8
Winter	[2 3]	8607	4.0

Table 5.1: Selected topologies for seasonal MLP models

The topologies in table 5.1 were selected based on the criteria described in section 4.7. Lowest mean squared error indicates a model with better generalization capability.

Although there is a significant discrepancy between the overall mean squared error for the different seasons, those are explained with the number of characteristic days per season, and how those days vary from each other. The model for summer presents, according to figure 5.1, the lowest mean squared error from all available topologies. This can be explained with the quantity of characteristic sunny days present in a season like summer and the relative simplicity of those days over partially cloudy days or overcast cloudy days. This is illustrated in figure 4.9, where the results for training and testing of the artificial neural networks are presented.

The performance results for the training and test set for the different MLP seasonal models are presented in table 5.2, and subsequently the results are illustrated in figure 5.2, 5.3, 5.4 and 5.5 for Spring, Summer, Autumn and Winter respectively. For each one of these figures, the top graphic represents the performance in the training set, the middle graphic the absolute error in the test set and the bottom graphic the performance in the test set.

In table 5.2 the displayed performance values were measured using the training sets defined in section 4.4.2 and corresponding test sets explained in section 4.5.3.

Seasonal models	Spring	Summer	Autumn	Winter
MSE (training)	13629	13663	5552	14258
RMSE (training)	116.7	116.8	74	119
MAPE % (training)	4.3	3.1	3.9	3.8
MSE (test)	8356	4153	6753	8607
RMSE (test)	91.4	64.4	82.2	92.7
MAPE % (test)	4.1	2.2	3.7	4.0

Table 5.2: Performance of MLP Seasonal models for training and test sets

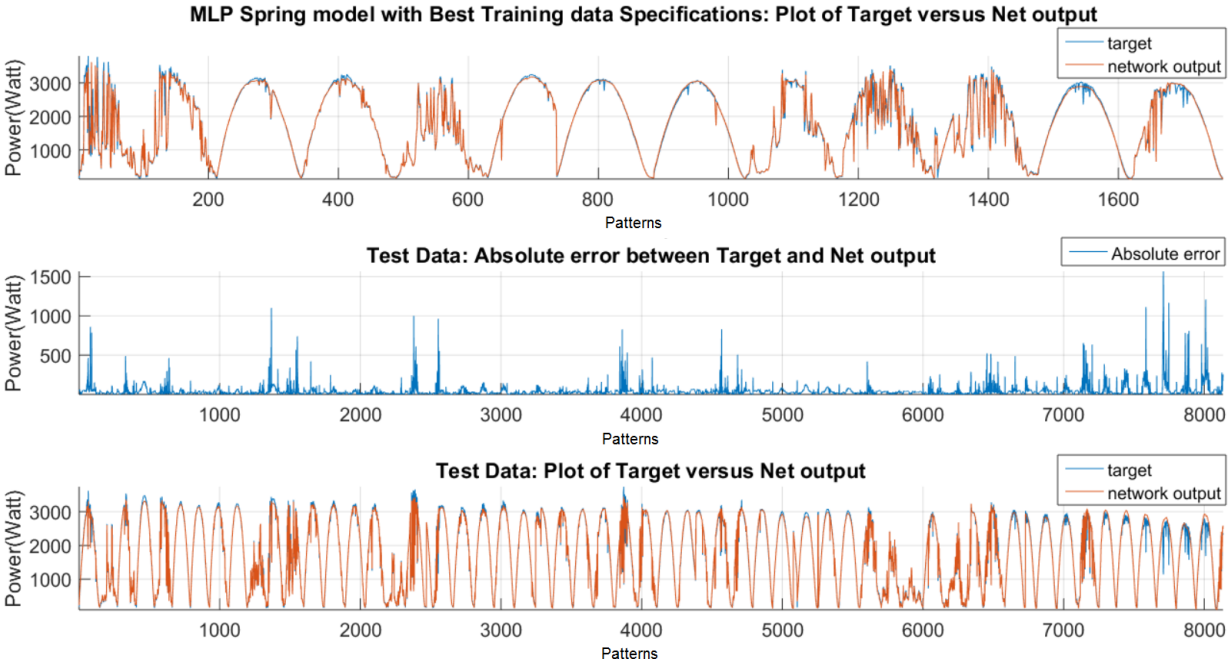

Figure 5.2: MLP Spring model evaluation.

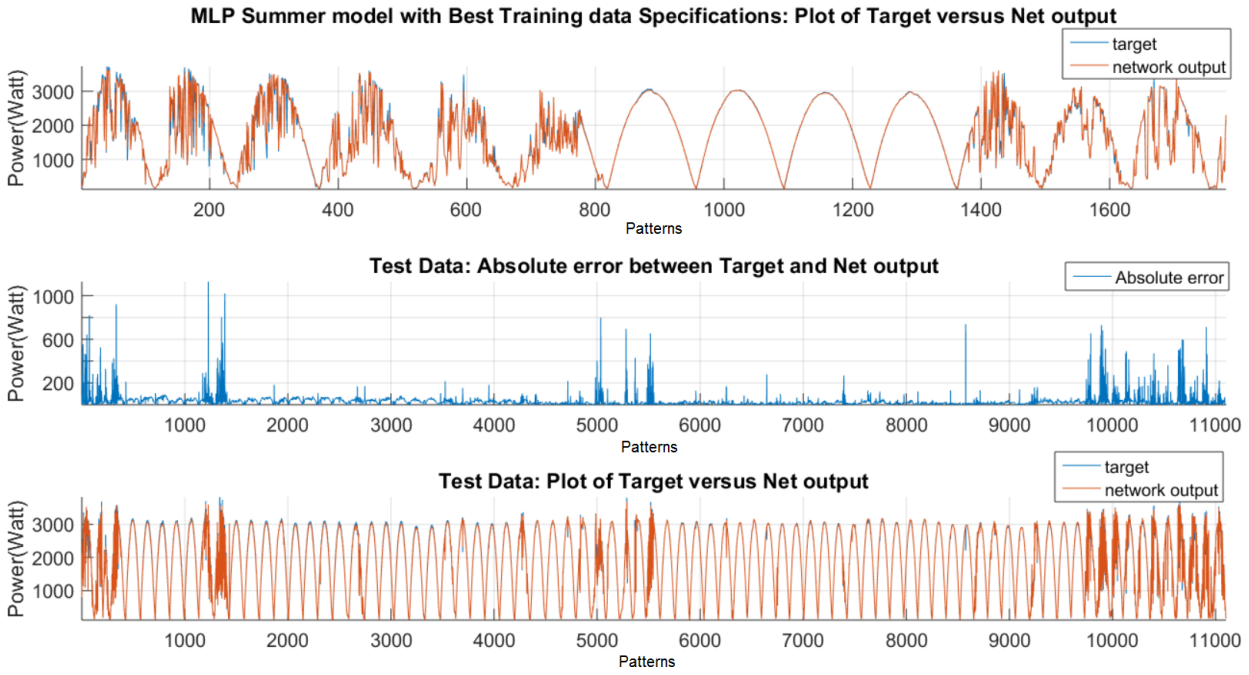

Figure 5.3: MLP Summer model evaluation.

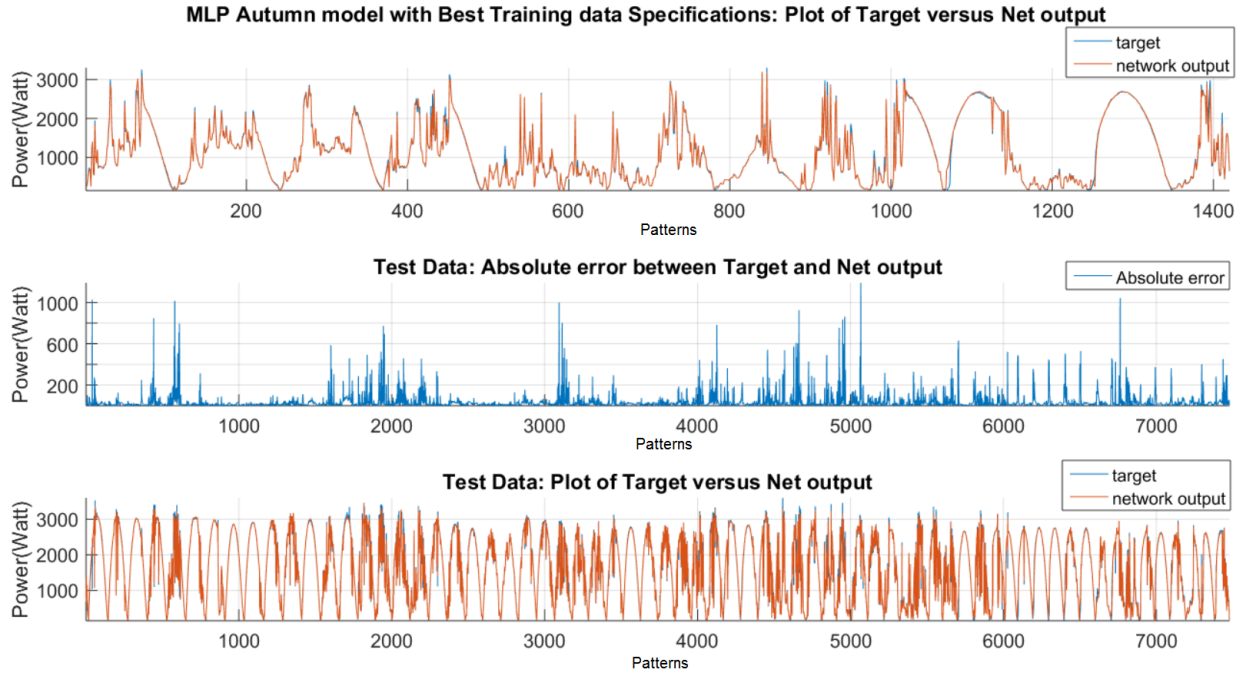

Figure 5.4: MLP Autumn model evaluation.

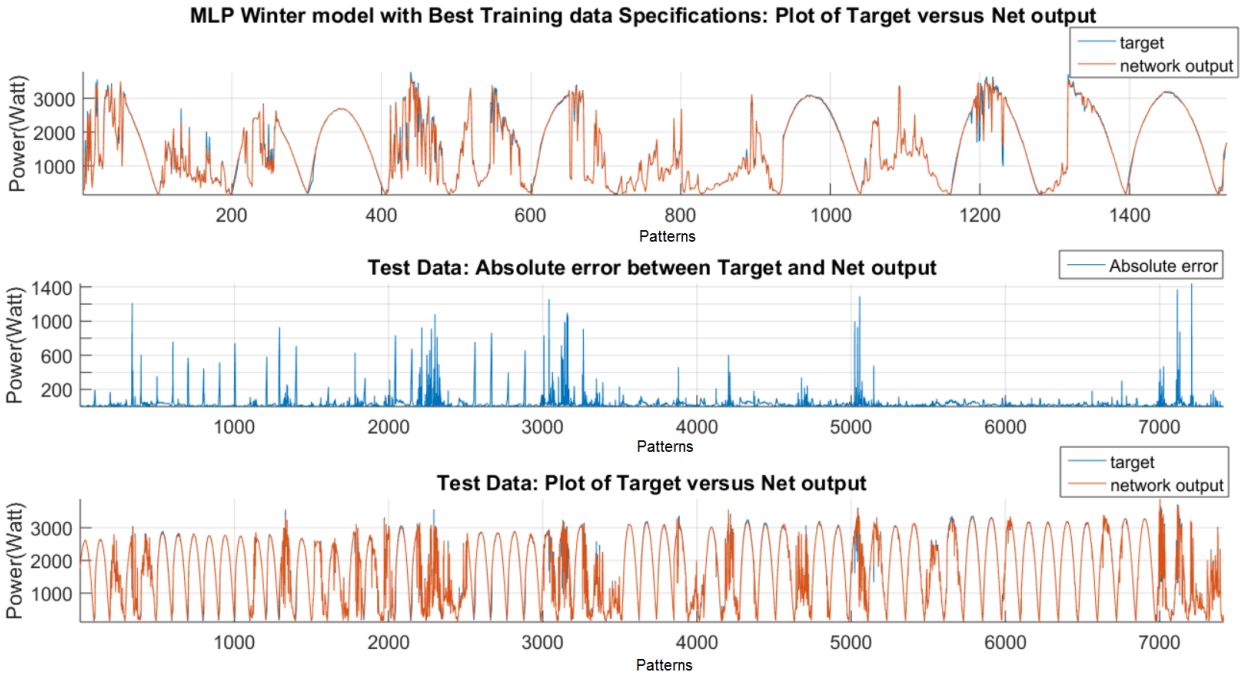

Figure 5.5: MLP Winter model evaluation.

5.1.2 Radial Basis Functions Results

The selected number of hidden neurons for the RBFNN models for the four seasons is presented in table 5.3. In figure 5.6 the mean squared error for the different topologies is shown.

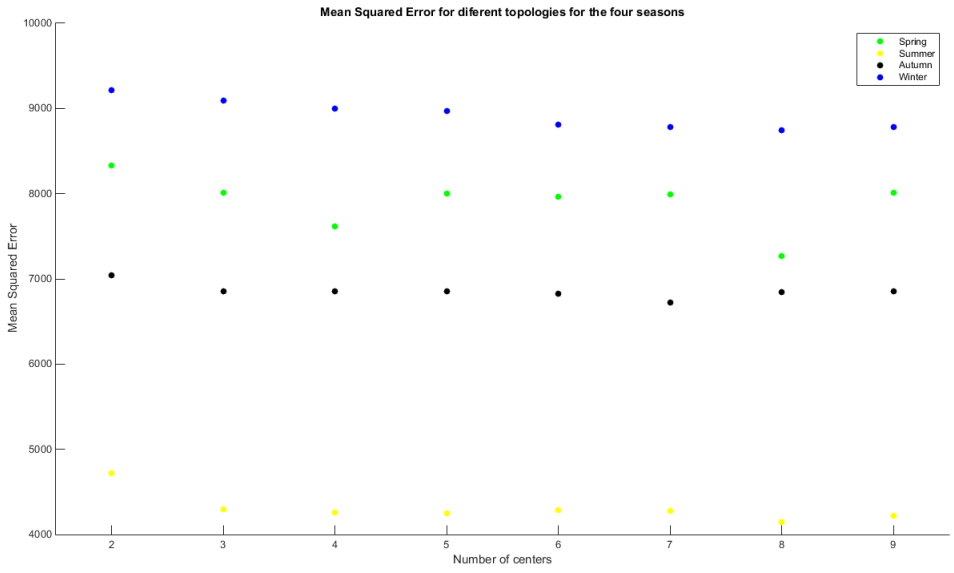

Figure 5.6: Variation of MSE across tested topologies for RBF seasonal models.

Seasonal models	Topology	MSE	MAPE (%)
Spring	[4]	7615	3.8
Summer	[5]	4249	2.5
Autumn	[7]	6723	3.7
Winter	[7]	8778	4.1

Table 5.3: Selected topologies for seasonal RBF models

The selection of the RBF topologies follows the criteria described in section 4.7. The performance results for the RBF seasonal models are displayed in table 5.4, and the corresponding graphics for the several models are shown in figures 5.7, 5.8, 5.9 and 5.10 for Spring, Summer, Autumn and Winter respectively.

In equivalent practice to MLP results in table 5.2, the training and test sets for RBFNN are in table 5.4 and were determined in sections 4.4.2 and 4.5.3, respectively.

Seasonal models	Spring	Summer	Autumn	Winter
MSE (training)	14004	12501	5760	11752
RMSE (training)	118.3	111.8	75.9	108.4
MAPE % (training)	3.8	3.4	4.4	3.5
MSE (test)	7615	4249	6723	8778
RMSE (test)	87.3	65.2	82	93.7
MAPE % (test)	3.8	2.45	3.7	4.2

Table 5.4: Performance of RBF Seasonal models for training and testing sets

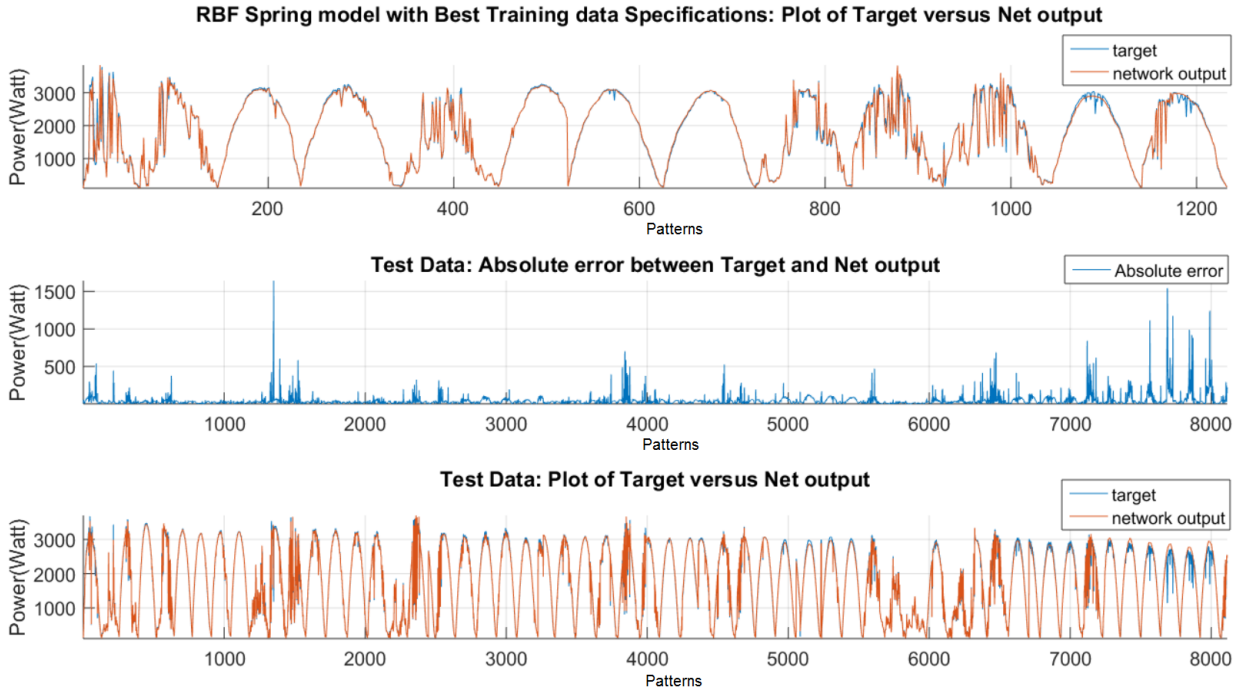

Figure 5.7: RBF Spring model evaluation.

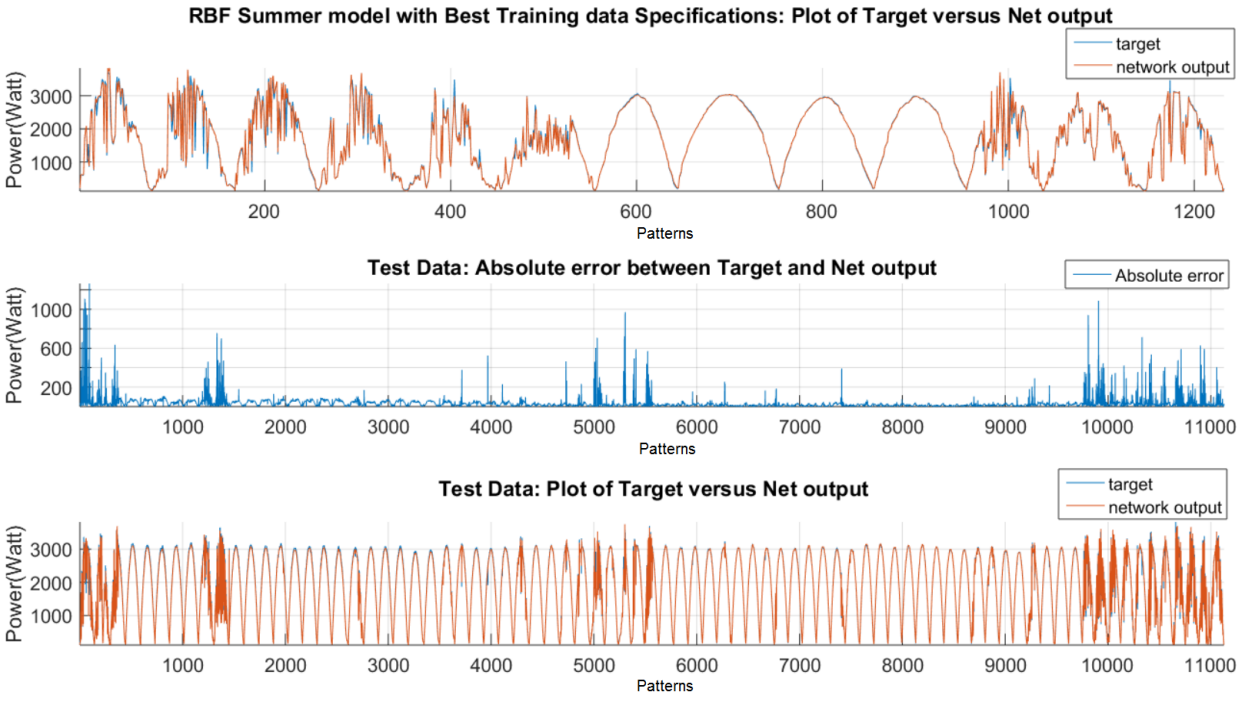

Figure 5.8: RBF Summer model evaluation.

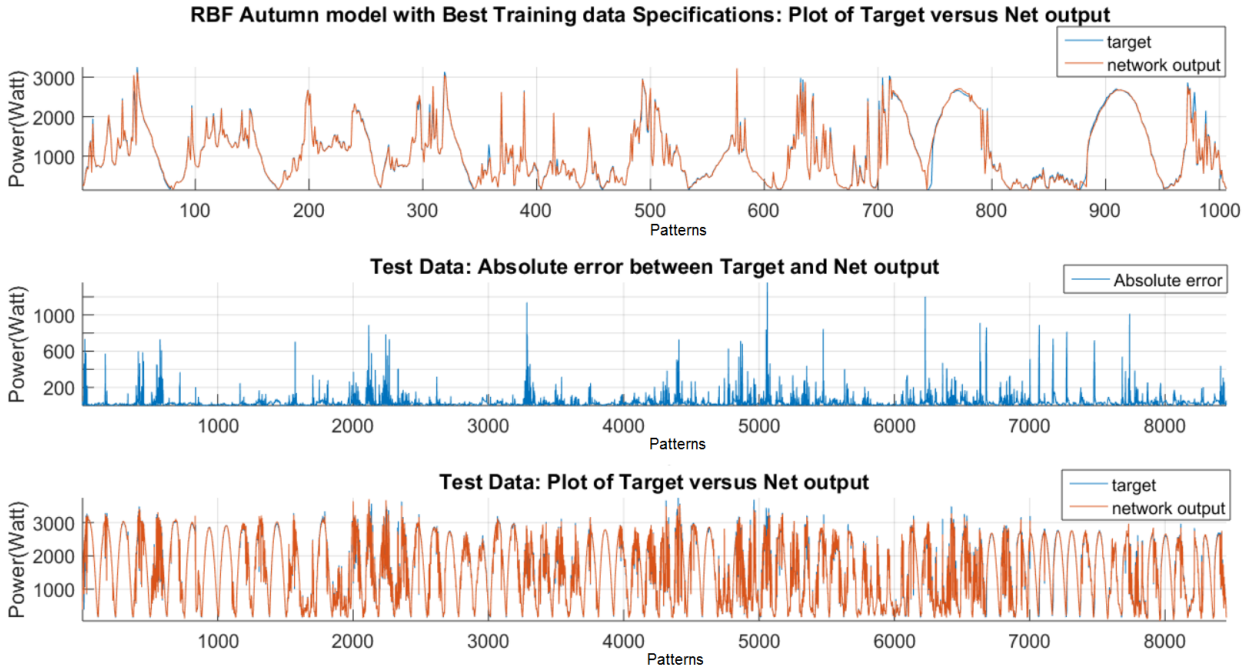

Figure 5.9: RBF Autumn model evaluation.

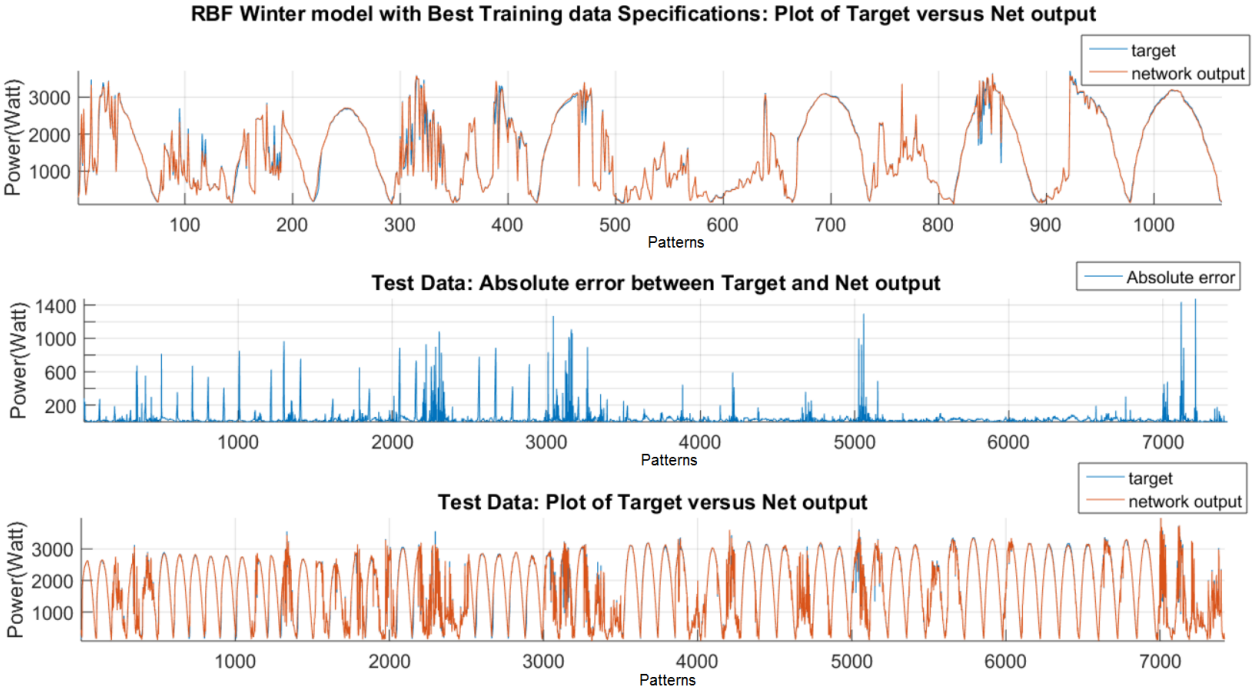

Figure 5.10: RBF Winter model evaluation.

5.2 Yearly model

The yearly models training set was developed using characteristic days from the different seasonal training sets. Considering the MAPE value, which translates an overall performance of the network, it is noticeable that the value of MAPE for the yearly models is an average value of the MAPE for the seasonal models for the MLPs and RBFs.

The results for the training and test for the selected topologies using MLPs and RBFs are presented in table 5.5 and 5.6, respectively. Also the illustrations of the training and test results are presented in figure 5.11 for the MLP and figure 5.12 for the RBF network.

	Yearly model
Topology	[2 4]
MSE (training)	16496
RMSE (training)	128
MAPE % (training)	3.8
MSE (test)	7531
RMSE (test)	86.7
MAPE % (test)	3.4

Table 5.5: Performance of MLP Yearly model for training and test sets

	Yearly model
Hidden neurons	7
MSE (training)	11504
RMSE (training)	107.3
MAPE % (training)	3.4
MSE (test)	7211
RMSE (test)	84.9
MAPE % (test)	3.4

Table 5.6: Performance of RBF Yearly model for training and test sets

Figure 5.11: MLP Yearly model evaluation

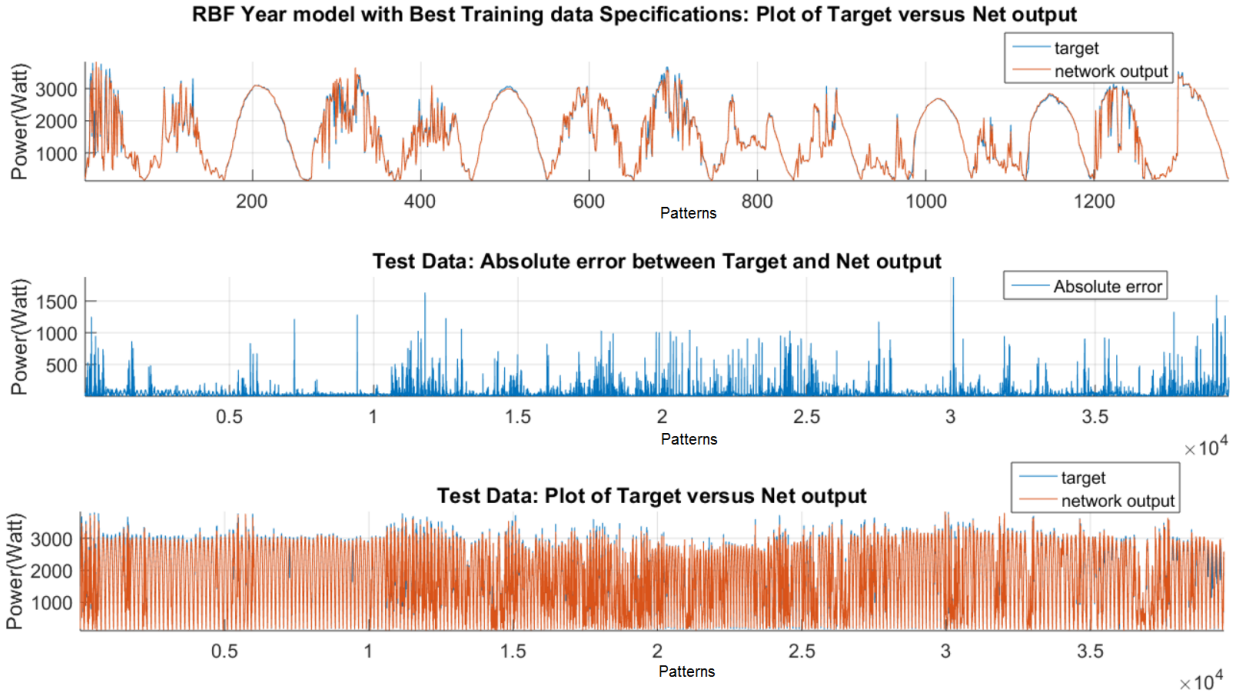

Figure 5.12: RBF Yearly model evaluation

5.3 Comparison between static models

In order to compare the static models, the complexity of the models should also be taken into account. In Artificial Neural Networks the usage of an increased quantity of hidden neurons increases their complexity. To compare models with different levels of complexity due to the different architecture of the networks, the number of free parameters is used. In table 5.7 the results from all the static models are presented.

Models		Spring	Summer	Autumn	Winter	Year	Weighted Average
MLP	Free parameters	23	15	15	19	23	x
	MSE (training)	13629	13663	5552	14258	16496	12174
	RMSE (training)	116.7	116.8	74	119	128	105.8
	MAPE % (training)	4.3	3.1	3.9	3.8	3.8	3.7
	MSE (test)	8356	4153	6753	8607	7531	6851
	RMSE (test)	91.4	64.4	82.2	92.7	86.7	81.95
	MAPE % (test)	4.1	2.2	3.7	4.0	3.4	3.45
RBF	Free parameters	12	15	21	21	21	x
	MSE (training)	14004	12501	5760	11752	11504	10809
	RMSE (training)	118.3	111.8	75.9	108.4	107.3	102.6
	MAPE % (training)	3.8	3.4	4.4	3.5	3.4	3.8
	MSE (test)	7615	4249	6723	8778	7211	6756
	RMSE (test)	87.3	65.2	82	93.7	84.9	81.5
	MAPE % (test)	3.8	2.45	3.7	4.2	3.4	3.5

Table 5.7: Comparison between static models

Analyzing the performance of the seasonal models based on the results of the test MSE shows similar results from the different architectures, although seasonal models show a better consistency when ranking performance among them. Since these models focus specific seasons of the year, the performance is comparable with the complexity of the data in each season. The model with best performance proves to be the Summer model and the worst performance

the Winter model when considering the network complexity. The difference in performance from the Summer model to the remaining seasonal models can be explained by the similarity between the majority of sampled days in the Summer dataset, as mentioned in section 5.1.

The test MAPE shows a better performance for the MLP architecture for the summer and winter models, being outperformed by the spring model of the RBF architecture. For the RMSE applied to the test set this is also verified, being the RBF architecture the one with better performance for the spring and autumn model and the MLP architecture for the summer and winter model.

The year models are very similar in terms of performance, although the RBF model shows a slightly better performance in the results of the RMSE test.

Comparing the seasonal models to the annual model using the weighted average with respect to the number of days in table 5.7, presented in the last column, a performance comparison can be possible when evaluating the performance for the duration of 1 year. The combined seasonal performance for each architecture proves to be slightly better than an annual model when considering the MSE and RMSE, being the MAPE the only indicator of better performance of the annual model.

The combination of seasonal performance can be enhanced using models from different architectures, choosing seasonal models with the best performance per season. In table 5.8 the best combination possible using the available seasonal models is presented.

	MSE (test)	RMSE (test)	MAPE (test)	Architecture
Spring	7615	87.3	3.8	RBF
Summer	4153	64.4	2.2	MLP
Autumn	6723	82	3.7	RBF
Winter	8607	92.7	4.0	MLP
Year	7211	84.9	3.4	RBF
Weighted Average	6687	81	3.39	x

Table 5.8: Combination of best seasonal models.

The results from the comparison between seasonal models and annual model were expected. By allowing each individual seasonal model to learn from different subsets of data enhances their performance, exploiting certain characteristics in each subset. This comparison results in a paradigm when analyzing the difference in number of data samples and characteristics that the annual model is intended to learn while sharing a similar network complexity with individual seasonal models. Additionally, increasing network complexity will result in the model learning unwanted characteristics from the data.

5.4 Forecasting models

The proper evaluation of the performance from forecasting models was made through the evolution of the RMSE, for several prediction steps. This allows the performance of forecasting models to be assessed across different steps.

As described in chapter 4.6, the forecasting models predict 48 time steps ahead, each step separated by a time interval of 5 minutes. In this section the results and performance of the Air temperature and Solar irradiance forecasting models presented in figure 4.1 are going to be evaluated. Due to the scope of this thesis the performance evaluation will be done for each season, and for the whole year. Forecast performance is presented in figures 5.13, 5.14, 5.15 and 5.16.

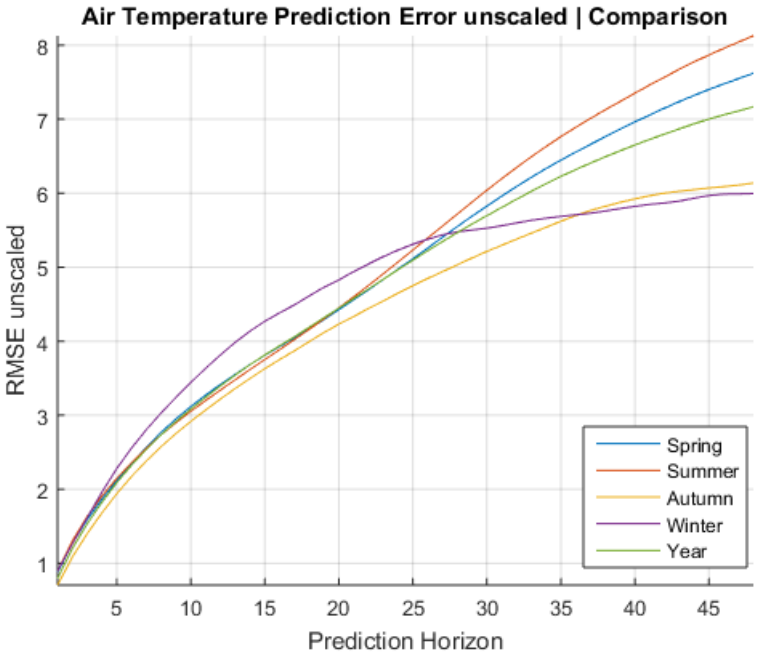

Figure 5.13: Air Temperature forecast RMSE unscaled

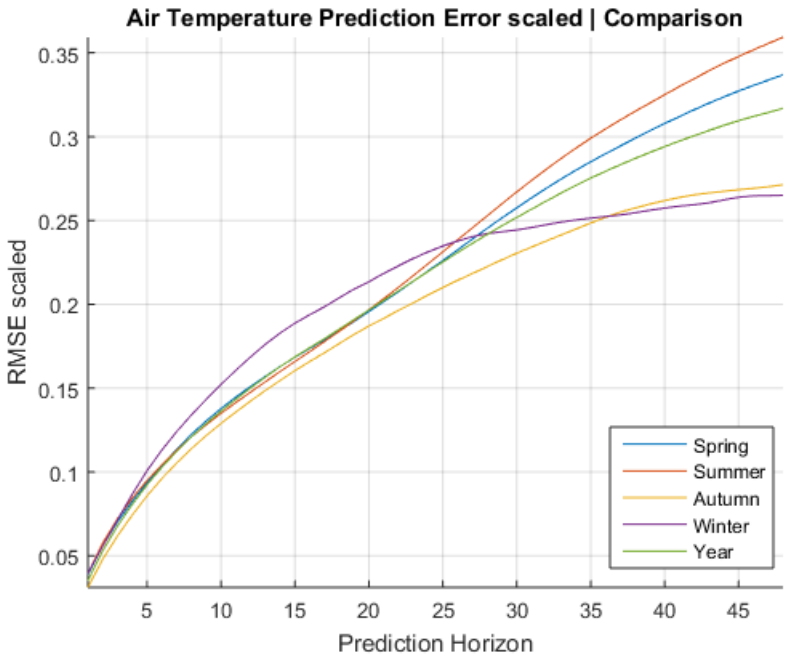

Figure 5.14: Air Temperature forecast RMSE scaled

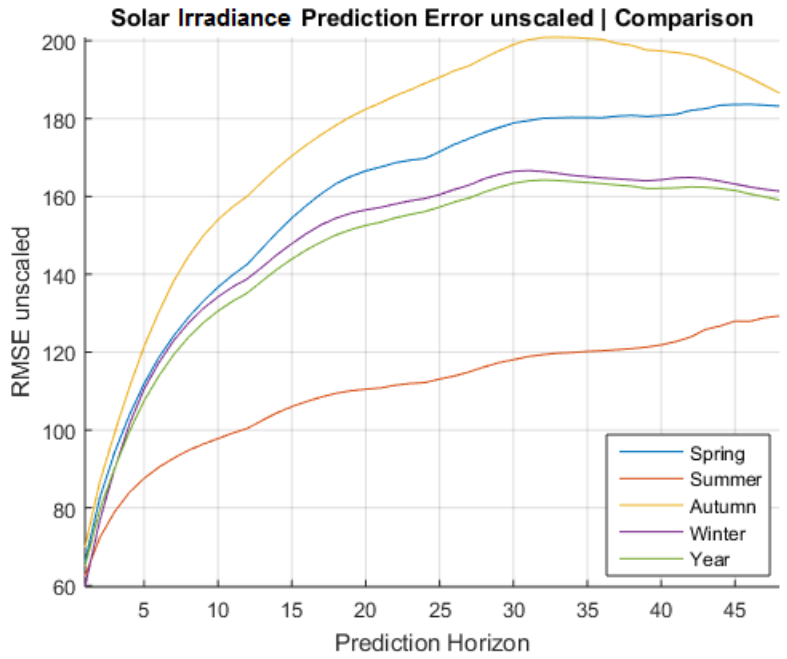

Figure 5.15: Solar Irradiance forecast RMSE unscaled

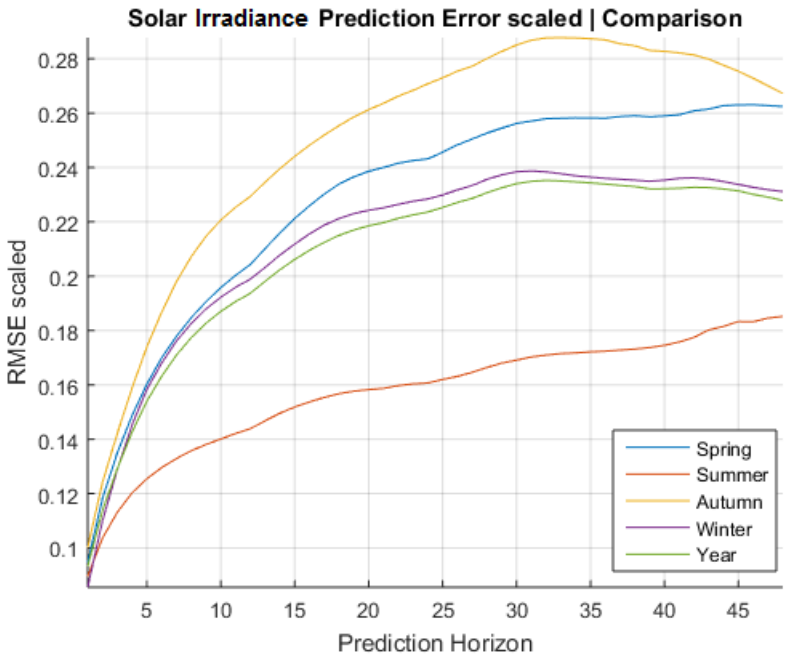

Figure 5.16: Solar Irradiance forecast RMSE scaled

In figures 5.15 and 5.16 the error associated with solar irradiance forecast shows better results for seasons like Summer and Winter. The reason for this might be that even considering typical sunny days as days with better performance, the solar irradiance curve exemplified in figures 4.8, 4.9, 4.10 and 4.11 changes its range across the year, due to the variation of solar exposition time. This range is due to the elliptic orbit of earth on the Solar plane maximizing the width variation of solar irradiance in transition seasons like the Spring and Autumn, decreasing the forecast performance. Another reason to explain the variation of performance in Solar Irradiance might be the different types of days already mentioned in chapter 4.4.1.

The air temperature forecasting performance follows a similar reasoning. Since the study region is geographically close to the Atlantic ocean, thermal amplitude is low. Evaluating the available data, there is a difference in temperature variance between Summer and Winter, having Winter a variance of $22.8^{o}C$ opposed to 27.9^{o}C in Summer. Analyzing from the artificial neural networks performance perspective, a lower range will facilitate a better performance.

To further evaluate the forecasting models performance using the characteristic days described in section 4.4.1, figures 5.17, 5.18, 5.19 and 5.20 show the performance of the forecasting models used for Solar Irradiance and Air Temperature for the characteristic days classified using the theoretical method described in the same section.

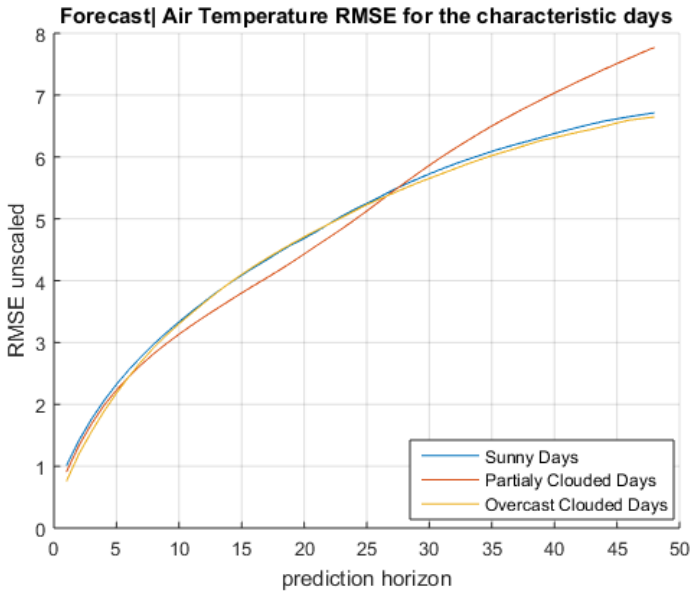

Figure 5.17: Air Temperature forecast unscaled RMSE for the characteristic days.

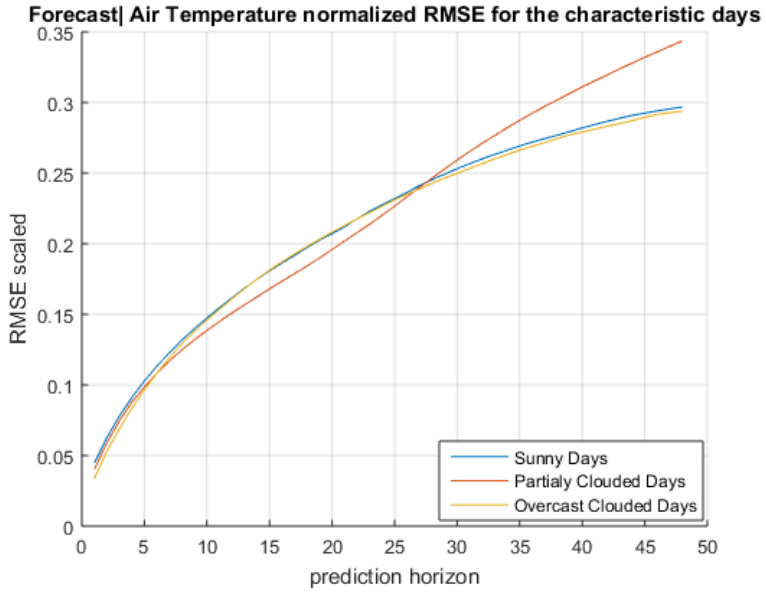

Figure 5.18: Air Temperature forecast scaled RMSE for the characteristic days.

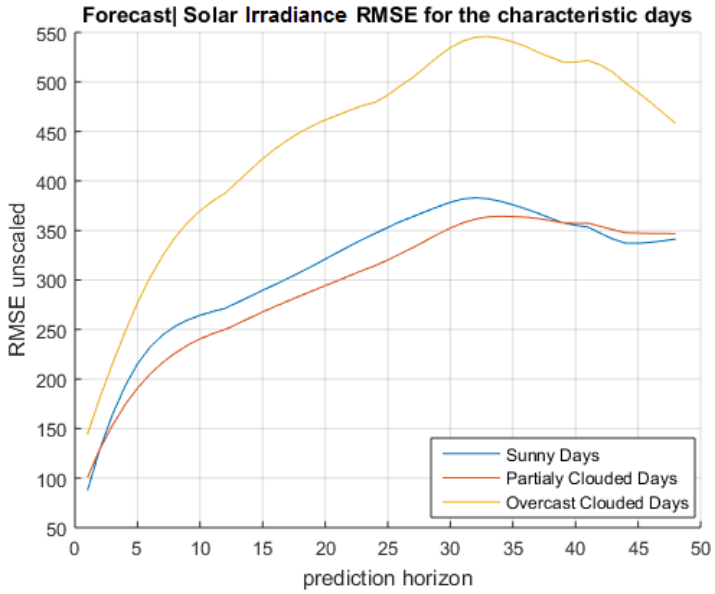

Figure 5.19: Solar Irradiance forecast unscaled RMSE for the characteristic days.

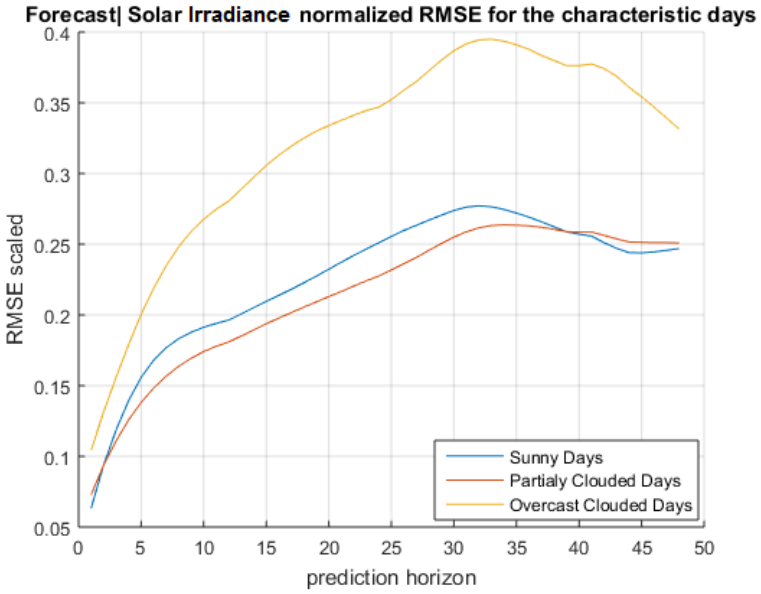

Figure 5.20: Solar Irradiance forecast scaled RMSE for the characteristic days.

In figure 5.19 and 5.20 it is shown that days classified as "sunny" or "partially clouded" when used by the forecasting models have a better performance in comparison with "overcast clouded" days. This is expected as "overcast clouded days" are characteristic for having a high frequencies in the data becoming very complicated to accurately forecast. This also justifies the discrepancy in performance between summer and the rest of the seasons evaluated in figure 5.1, 5.6, 5.15 and 5.16 being Summer the season with the highest concentration of characteristic sunny and partially clouded days.

Additionally, to analyze the performance of the forecasting models, figure 5.21 and 5.22 show forecasted values of air temperature and solar irradiance, for a selection of three different characteristic days from the seasonal data sets, for different time steps. In figures 5.21 and 5.22, target refers to the acquired data values and step refers to the resulting forecasts, for the number of steps indicated. In tables 5.10 and 5.11 the values for RMSE and normalized RMSE are presented for the characteristic days in figures 5.21 and 5.22, respectively. The calendar date of those same characteristic days is shown in table 5.9

		Day	Month	Year
Spring	Sunny	31	5	2015
	Partially clouded	21	6	2015
	Overcast clouded	22	3	2015
Summer	Sunny	21	7	2014
	Partially clouded	2	8	2014
	Overcast clouded	31	7	2014
Autumn	Sunny	25	9	2014
	Partially clouded	19	10	2014
	Overcast clouded	14	11	2014
Winter	Sunny	25	1	2015
	Partially clouded	28	2	2015
	Overcast clouded	31	1	2015

Table 5.9: Dates of characteristic days used in the results.

		RMSE (ºC)				NRMSE (%)			
Step		1	12	24	48	1	12	24	48
Spring	Sunny	0.59	2.6	4.99	6.3	1.53	6.68	12.8	16.2
	Partially clouded	0.6	2.45	3.73	4.91	1.53	6.29	9.58	12.6
	Overcast clouded	0.57	2.93	3.36	4.92	1.47	7.53	8.64	12.6
Summer	Sunny	0.91	2.31	4.84	7.77	3.23	8.21	17.21	27.6
	Partially clouded	0.7	2.86	5.54	6.49	2.49	10.14	19.69	23
	Overcast clouded	0.56	2.40	4.1	5.01	1.98	8.54	14.43	17.78
Autumn	Sunny	0.84	3.22	5.88	7.7	2.14	8.23	15.02	19.67
	Partially clouded	0.54	2.47	4.33	7.18	1.38	6.31	11.05	18.34
	Overcast clouded	0.53	2.41	3.75	4.01	1.35	6.15	9.59	10.43
Winter	Sunny	0.60	2.80	3.36	4.45	1.56	7.32	8.78	11.65
	Partially clouded	1.06	2.14	3.87	4.7	2.78	5.6	10.12	12.31
	Overcast clouded	0.49	2.78	2.82	3.63	1.29	7.28	7.38	9.49

Table 5.10: Air temperature forecast results for selected characteristic days within seasons.

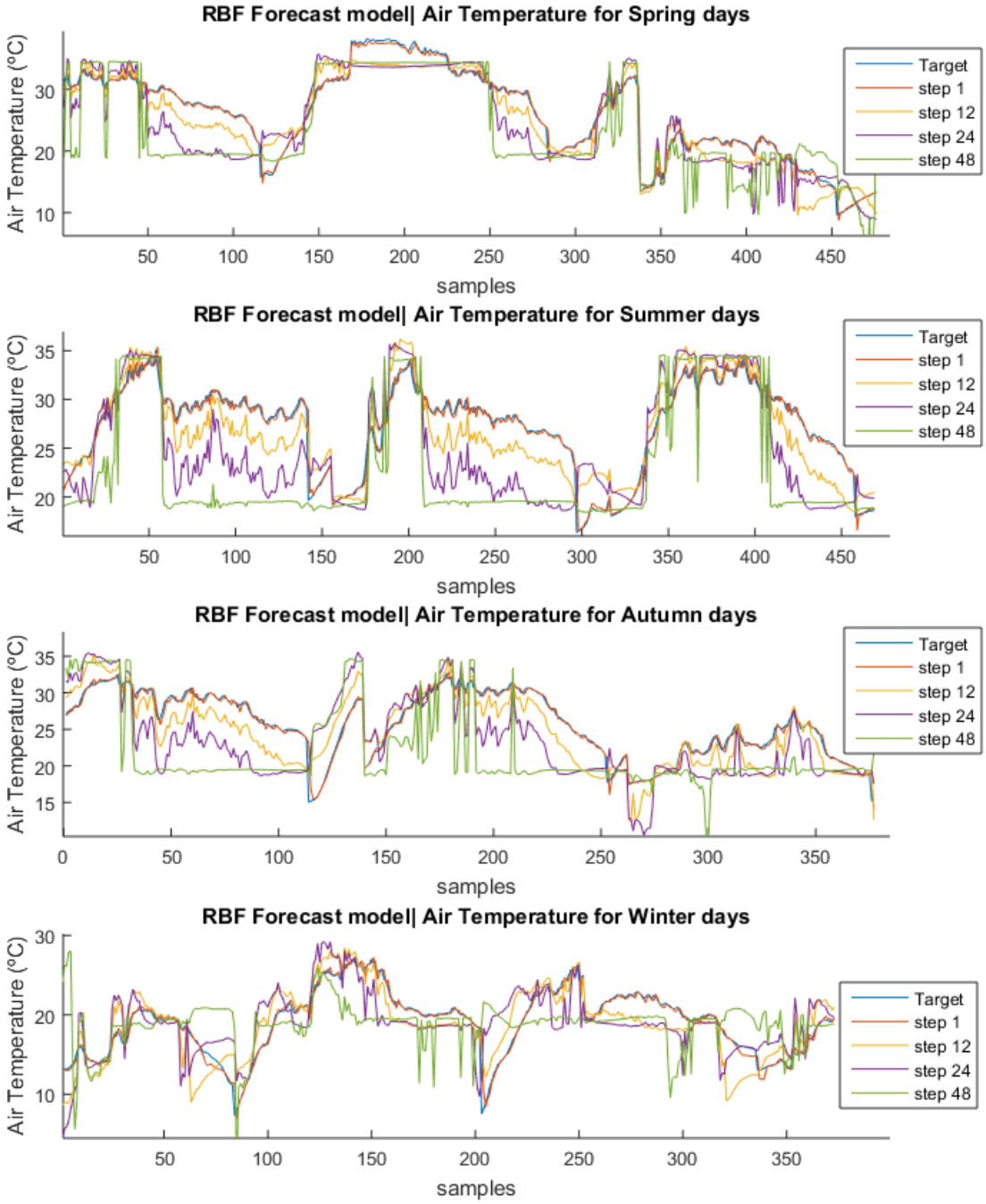

Figure 5.21: Forecast of air temperature for characteristic days across all seasonal data sets.

		RMSE ($Watt/m^2$)				NRMSE (%)			
Step		1	12	24	48	1	12	24	48
Spring	Sunny	44.6	102	98.4	104	3.23	7.37	7.11	7.52
	Partially clouded	147.7	146	171	176.1	10.68	10.56	12.36	12.74
	Overcast clouded	139.1	412.6	491.4	475.5	10.1	29.8	35.54	34.4
Summer	Sunny	39.7	83.2	78.8	92.1	2.87	6	5.7	6.66
	Partially clouded	163.5	175.2	176.2	192.5	11.82	12.67	12.74	13.92
	Overcast clouded	98.1	189.4	279.2	328	7.1	13.7	20.19	23.72
Autumn	Sunny	48.8	91.2	100.3	116.15	3.53	6.59	7.25	8.4
	Partially clouded	131.3	119.8	111.88	121.8	9.5	8.5	8.1	8.8
	Overcast clouded	154.63	275.93	303.7	282.2	11.2	20	21.9	20.4
Winter	Sunny	42.35	97.9	117.2	133.2	3.1	7.1	8.5	9.6
	Partially clouded	162.3	172.3	174.12	171.9	11.7	12.5	12.6	12.4
	Overcast clouded	205.8	192.3	187.8	186.3	14.9	14.3	13.6	13.5

Table 5.11: Solar irradiance forecast results for selected characteristic days within seasons.

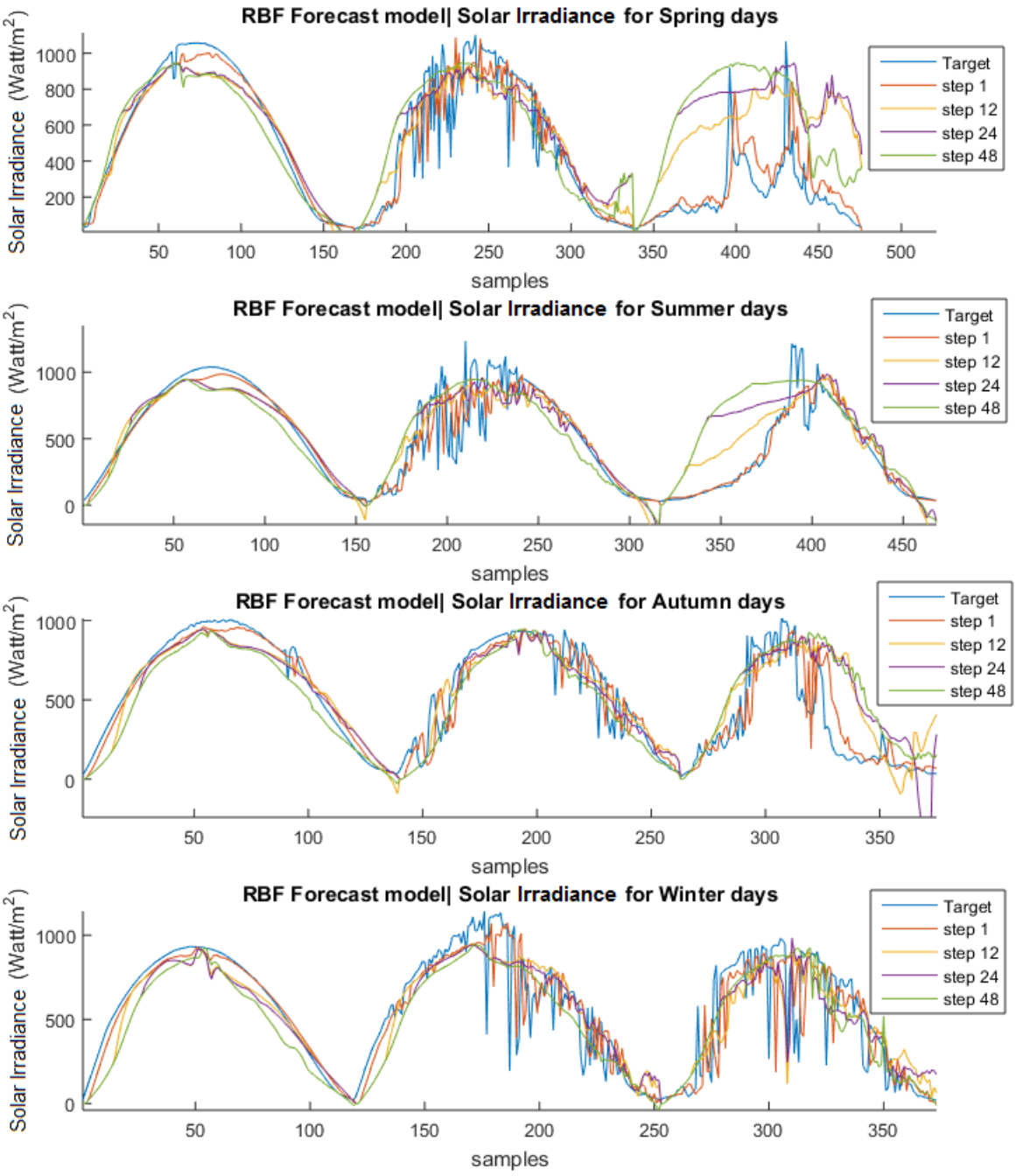

Figure 5.22: Forecast of solar irradiance for characteristic days across all seasonal data sets.

5.5 Global model results

In this section the results of the global model given by the combination of forecasting models with static models will be presented and evaluated, referring the global model in figure 4.1. The results are evaluated with a comparison between seasonal and annual models using the RMSE, and the normalized RMSE, over a selection of days.

In the following figures 5.23, 5.24, 5.25 and 5.26 RMSE comparison, scaled and unscaled will be presented across all available static models for the global model. From these figures a similar performance across seasonal models and annual models independent of network architecture used is observed. Additionally, tables 5.12, 5.13, 5.14 and 5.15 present a more detailed comparison in the performance of seasonal and annual networks from different architectures for the same seasonal periods.

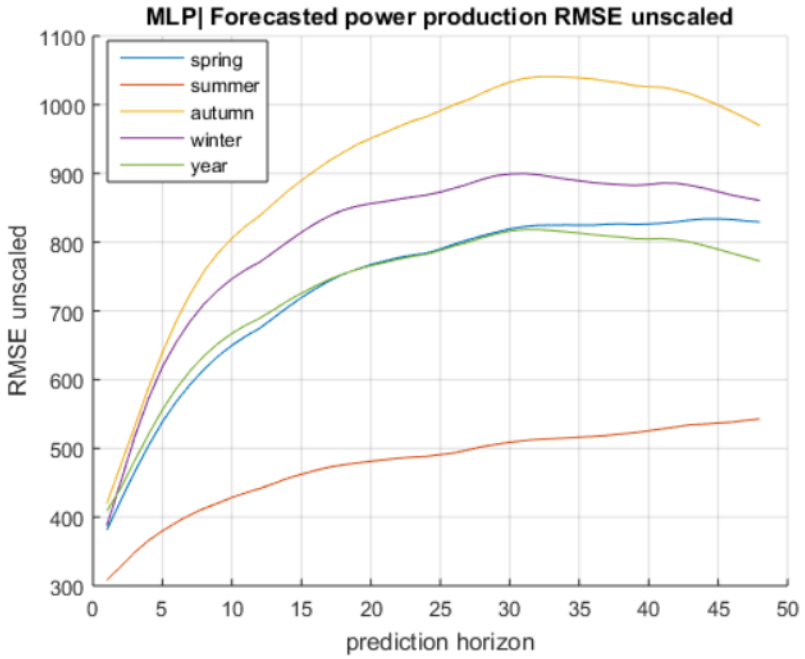

Figure 5.23: Comparison of unscaled RMSE using the MLP model.

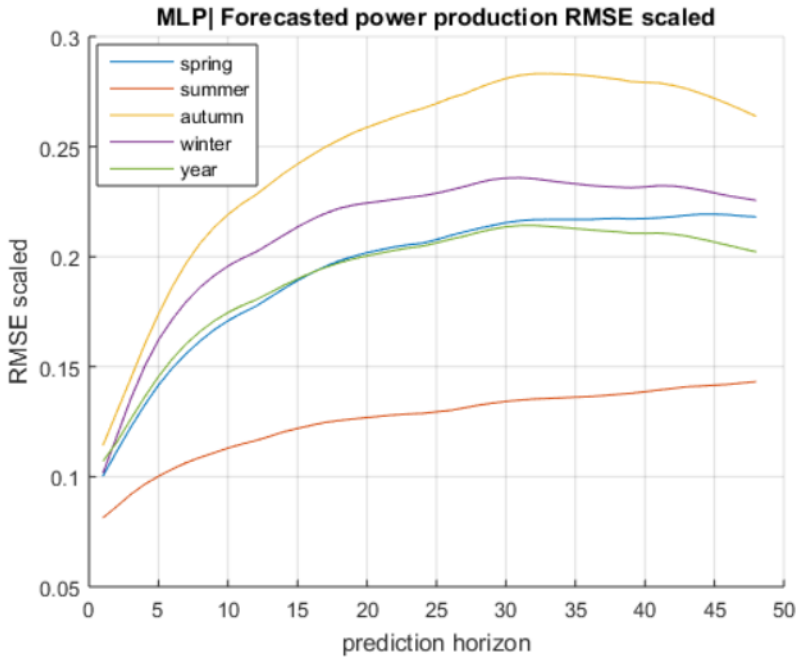

Figure 5.24: Comparison of scaled RMSE using the MLP model.

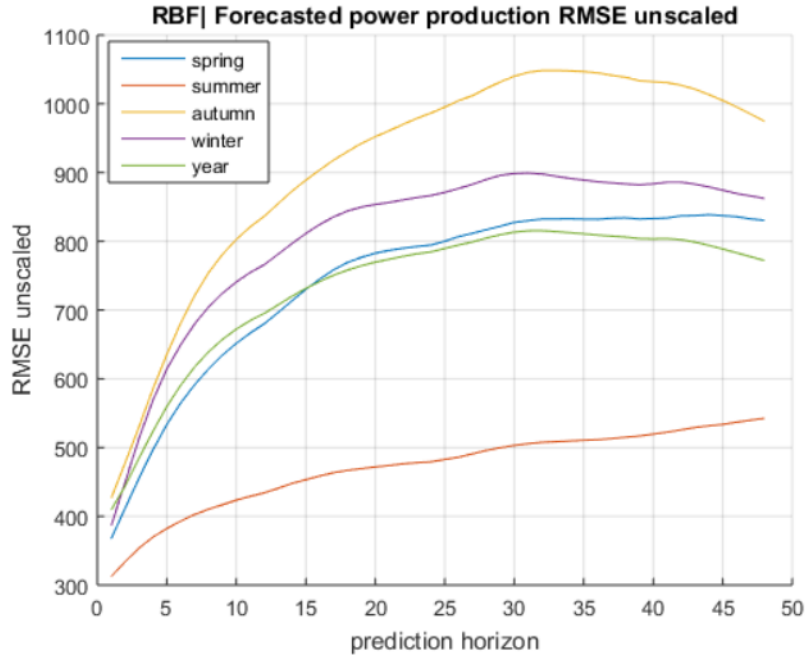

Figure 5.25: Comparison of unscaled RMSE using the RBF model.

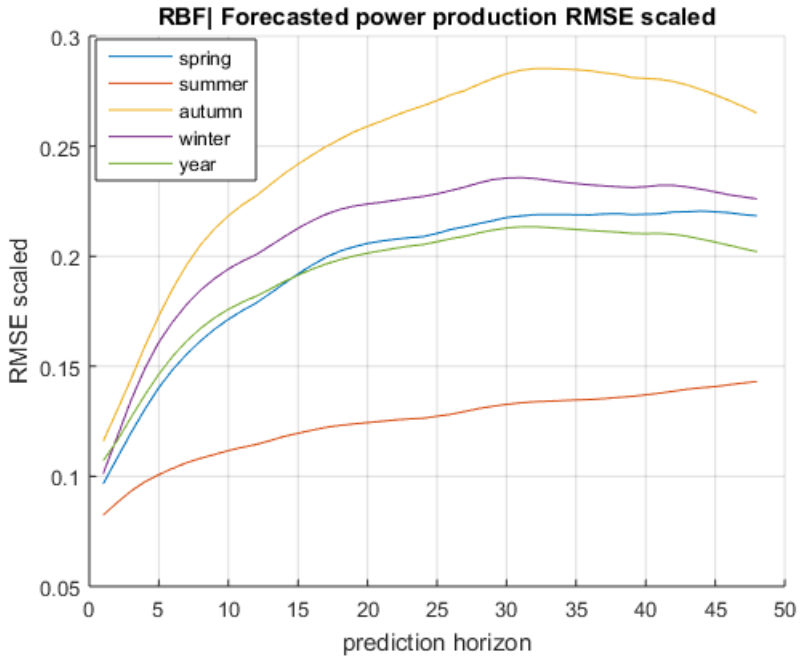

Figure 5.26: Comparison of scaled RMSE using the RBF model.

		RMSE				NRMSE			
steps		1	12	24	48	1	12	24	48
Spring	MLP	381	674.8	783.9	829.4	10	17.7	20.6	21.8
	RBF	367.5	679.9	794.5	830.5	9.7	17.9	20.9	21.8
Year	MLP	368	673.8	784	826.3	9.7	17.7	20.6	21.7
	RBF	368.1	682.5	791	826.4	9.7	17.9	20.8	21.7

Table 5.12: Performance comparison between Spring model and Annual model for the Spring seasonal period.

		RMSE				NRMSE			
steps		1	12	24	48	1	12	24	48
Summer	MLP	308	441	488	542.9	8.1	11.6	12.8	14.3
	RBF	312	434	479.2	542.5	8.2	11.4	12.6	14.3
Year	MLP	309.4	434.8	482.7	536.3	8.2	11.5	12.7	14.1
	RBF	313.4	448.5	493.5	543.3	8.3	11.8	13	14.3

Table 5.13: Performance comparison between Summer model and Annual model for the Summer seasonal period.

		RMSE				NRMSE			
steps		1	12	24	48	1	12	24	48
Autumn	MLP	419.7	838.4	982.6	969.6	11.4	22.8	26.7	26.4
	RBF	426.5	836.1	986.5	974.4	11.6	22.7	26.8	26.5
Year	MLP	434.3	877.6	1025.5	1011.1	11.8	23.9	27.9	27.5
	RBF	435.8	881.9	1028.8	1011.9	11.9	24	28	27.5

Table 5.14: Performance comparison between Autumn model and Annual model for the Autumn seasonal period.

		RMSE				NRMSE			
steps		1	12	24	48	1	12	24	48
Winter	MLP	467.7	833.4	921.8	907	12.3	21.8	24.2	23.8
	RBF	386.6	765.85	866.5	862.3	10.1	20.1	22.7	22.6
Year	MLP	399.3	791.4	891.9	873.7	10.5	20.7	23.4	22.9
	RBF	401.4	796.8	892	875	10.5	20.9	23.4	22.9

Table 5.15: Performance comparison between Winter model and Annual model for the Winter seasonal period.

These tables show that the performance of the annual model is very similar to the Spring and Summer models when compared for those same seasonal periods. Compared with the Autumn and Winter models, the annual model is slightly outperformed by both architectures of the Autumn model and by the RBF architecture of the Winter model.

To compare the global model performance, figures 5.27, 5.28, 5.29 and 5.30 show the forecasted power production and its measured values. In those figures the target represents the real power production values, and the steps represent the forecasted values, for that particular step. Those same figures also intend to compare the ANN architecture and the model performance across the seasonal and annual models over the selection of the same characteristic days chosen in section 5.4, properly dated in table 5.9.

Although the three characteristic days were introduced to assess the performance of the global model, it is worth mentioning the distribution of these days, as shown in figure 4.7, standing out that sunny and partially clouded days prove to be more frequent in the data set.

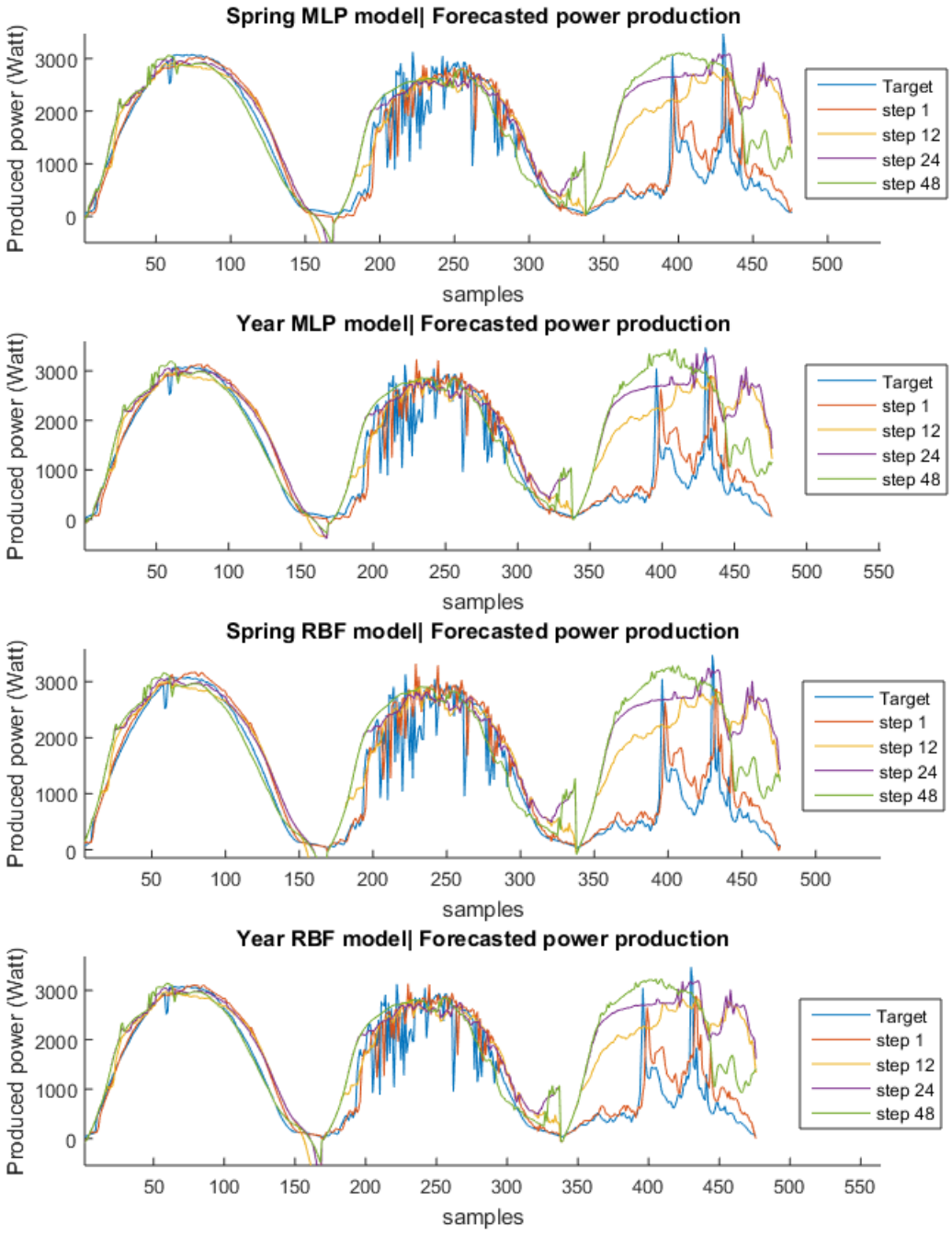

Figure 5.27: Comparison across available models for a selected Spring day.

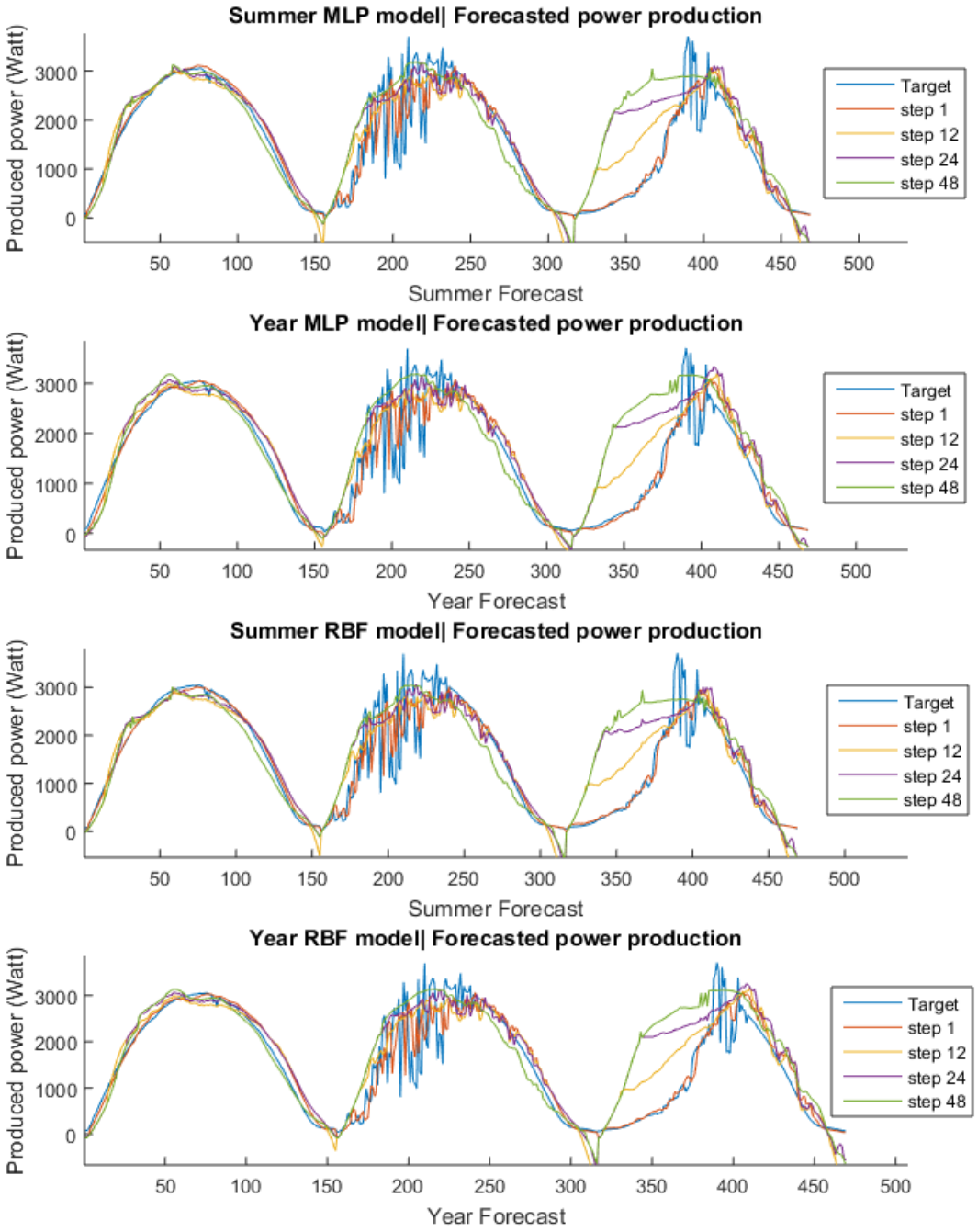

Figure 5.28: Comparison across available models for a selected Summer day.

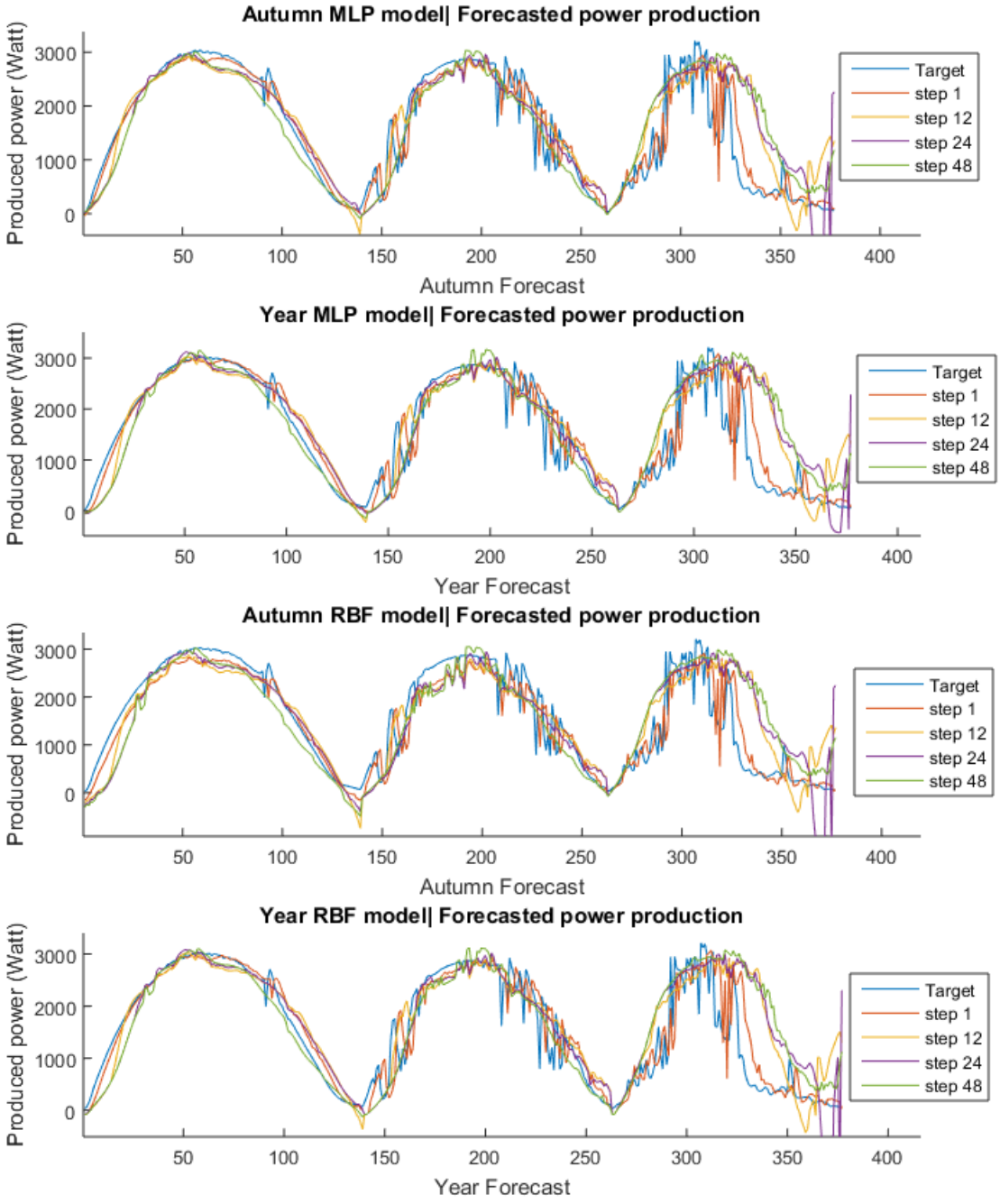

Figure 5.29: Comparison across available models for a selected Autumn day.

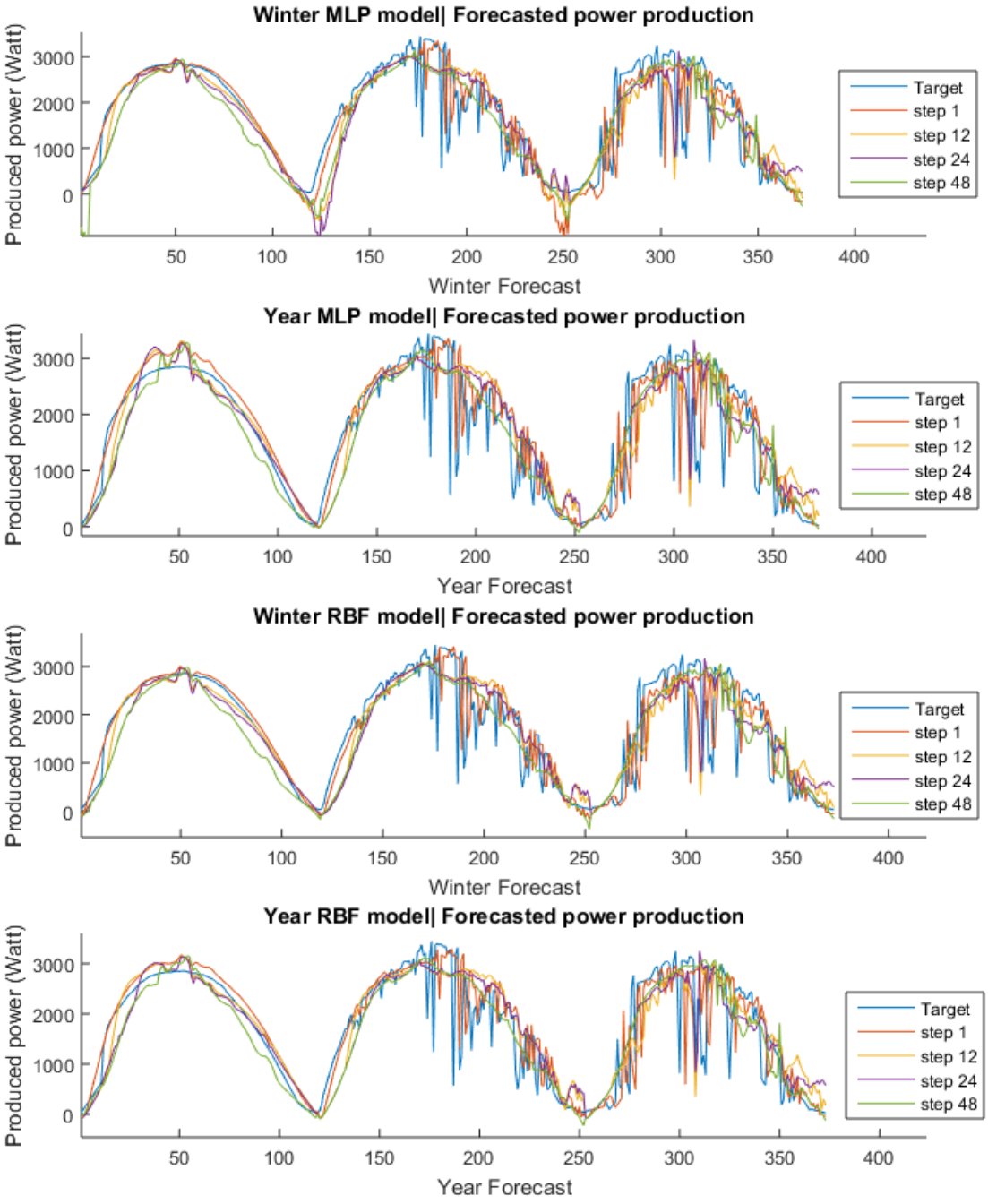

Figure 5.30: Comparison across available models for a selected Winter day.

6 Conclusion and Future work

6.1 Conclusions

In this thesis a methodology for short-term forecasting of generated power for a photovoltaic on grid structure used at the University of Algarve was presented. The methodology used the available data collected by the referenced structure, corresponding to day light hours of 343 days allowing the creation of several models and the comparison between them.

The methodology applied focuses in a short-term forecasting model for PV generated power using artificial neural networks, specifically MLPs and RBFs. The presented system consists in a two part model, using two forecasting models for a 48 multi-step prediction of solar irradiance and air temperature within a range of 4 hours, and a static model performing the estimation of generated power based on the two forecasts.

For the case of solar irradiance forecasting, the model performance was better in Summer and worse for Autumn. The differences in performance suggest a correlation with the similarity of the recorded values for specific seasons, being one example the smoothness of sunny days in the summer. On the other hand the air temperature forecasting model has a similar effect with respect to the registered air temperature range for different seasons, having a better performance in seasons like Winter with lower air temperature range and worse results in the Summer with a wider air temperature range.

When using characteristic days as a comparison set, the air temperature forecasting model shows a significant discrepancy in performance between partially clouded days and the rest of the characteristic days. The solar irradiance forecasting model show an expected difference between overcast clouded days and the remaining characteristic days, with sunny and partially clouded days being more predictable and, as shown before, also more frequent in the data set.

For the static estimation model, several models were designed, varying the architecture, topology and data scope in order to determine the model with the best performance. Although, the results from seasonal forecasting models compared separately suggested very distinct performance from the annual models, when compared in the same data the results are similar. Overall the combination of seasonal static models obtains a slightly better performance than the annual model.

Artificial Neural Networks proved to be a very powerful non-linear approximator as expected, and very efficient predictors.

6.2 Difficulties and future work

The restriction of data availability to day time proved to be very harmful, in terms of performance, to the forecasting models. Although an alternative solution was applied to the solar irradiance forecasting model, the air temperature model was based only in day time values.

Future work would be the application of this methodology using data from a 24 hour cycle for air temperature and solar irradiance which is expected to improve performance of the forecasting models. As a complementary work, the creation of a complementary neural system to predict the building power consumption and adding an energy storage capability to the present photovoltaic system in order to maximize energy price-load efficiency would also be interesting. Finally, it would be very interesting to upgrade the forecasting structure with forecasting models for individual seasons and forecasting models defined by characteristic days.

Bibliography

References

[1] Daniel G Nocera and Nathan S Lewis. For the "In This Issue" summary. 104(42), 2007.

[2] E. Sortomme and M. a. El-Sharkawi. Optimal Power Flow for a System of Microgrids with Controllable Loads and Battery Storage. *2009 IEEE/PES Power Systems Conference and Exposition*, pages 1–5, 2009.

[3] F. Almonacid, C. Rus, P. J. Pérez, and L. Hontoria. Estimation of the energy of a PV generator using artificial neural network. *Renewable Energy*, 34(12):2743–2750, 2009.

[4] I. a. Basheer and M. Hajmeer. Artificial neural networks: Fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3–31, 2000.

[5] Simon Haykin. *Neural Networks and Learning Machines*. Pearson Education, third edition, 2014.

[6] D. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex systems*, 2:321–355, 1988.

[7] António Ruano. Artificial Neural Networks. Technical report, Centre for Intelligent Systems, University of Algarve, Faro,Portugal, 2014.

[8] J. Park and I. W. Sandberg. Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation*, 3(2):246–257, 1991.

[9] a K Jain, J C Mao, and K M Mohiuddin. Artificial neural networks: A tutorial, 1996.

[10] I V Tetko, D J Livingstone, and A I Luik. Neural-Network Studies .1. Comparison of Overfitting and Overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995.

[11] R.E. Bellman. Adaptive control Processes, 1961.

[12] Ananth Ranganathan. The Levenberg-Marquardt Algorithm. *Tutoral on LM algorithm*, 142(June):1–5, 2004.

[13] E Radziemska. The effect of temperature on the power drop in crystalline silicon solar cells. 28:1–12, 2003.

[14] Antonio Luque and Steven Hegedus. *Handbook of Photovoltaic Science and Engineering*. Wiley, second edition, 2011.

[15] F Kasten and a T Young. Revised optical air mass tables and approximation formula. *Applied optics*, 28(22):4735–4738, 1989.

[16] E. G. Laue. The measurement of solar spectral irradiance at different terrestrial elevations. *Solar Energy*, 13(1):43–57, 1970.

[17] M G Villalva, J R Gazoli, and E R Filho. Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays. *Power Electronics, IEEE Transactions on*, 24(5):1198–

1208, 2009.

[18] K. Hersch, P. ; Zweibel. Basic Photovoltaic Principles and Methods. Technical report, United States, 1982.

[19] Michael Boxwell. *Solar Electricity Handbook: A simple, Pratical Guide to Solar Energy - designing and installing photovoltaic solar eletric systems.* Greenstream Publishing, sixth edition, 2009.

[20] Fengxia Zheng and Shouming Zhong. Time series forecasting using a hybrid RBF neural network and AR model based on binomial smoothing. *World Academy of Science, Engineering and Technology*, 5(3):419–423, 2011.

[21] Cheng-Ming Lee and Chia-Nan Ko. Time series prediction using {RBF} neural networks with a nonlinear time-varying evolution {PSO} algorithm. *Neurocomputing*, 73(3):449–460, 2009.

[22] Pedro M. Ferreira and António E. Ruano. New advances in intelligent signal processing. In António Ruano and Annamária Várkonyi-Kóczy, editors, *Studies in Computational Intelligence*, volume 372, pages 21–53. Springer Berlin Heidelberg, 2011.

[23] Ajith Abraham and Lakhmi Jain. Evolutionary Multiobjective Optimization. pages 1–6, 2005.

[24] Changsong Chen, Shanxu Duan, Tao Cai, and Bangyin Liu. Online 24-h solar power forecasting based on weather type classification using artificial neural network. *Solar Energy*, 85(11):2856–2870, 2011.

[25] Badia Amrouche and Xavier Le Pivert. Artificial neural network based daily local forecasting for global solar radiation. *Applied Energy*, 130:333–341, 2014.

[26] a. Mellit, a. Massi Pavan, and V. Lughi. Short-term forecasting of power production in a large-scale photovoltaic plant. *Solar Energy*, 105:401–413, 2014.