Electricity Load Demand Forecasting in Portugal Using Least-Squares Support Vector Machines

Isaura Denise Filipe Cuambe

M.Sc. in Informatics Engineering

Master thesis dissertation supervised by:

Professor Doctor Pedro M. Ferreira

2013

Electricity Load Demand Forecasting in Portugal Using Least-Squares Support Vector Machines

Isaura Denise Filipe Cuambe

M.Sc. in Informatics Engineering

Master thesis dissertation supervised by:

Professor Doctor Pedro M. Ferreira

2013

Electricity Load Demand Forecasting in Portugal Using Least-Squares Support Vector Machines

Declaração de autoria de trabalho:

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

(ISAURA DENISE FILIPE CUAMBE)

Ao meu companheiro...

Dário Amade

Contents

List of Figures

List of Tables

Acknowledgements

The work will be conducted on the framework of a research project, funded by *Fundação para a Ciência e a Tecnologia* (FCT)[1], in collaboration with *Rede Elétrica Nacional* (REN), where I am receiving a research grant. I would like to thank REN for the opportunity and theme, FCT for the funding, and the Algarve STP - Algarve Science and Technology Park and the University of Algarve for hosting this work.

To my supervisor, Professor Doctor Pedro M. Ferreira, for the opportunity to participate in this project, for the help on the definition of the object of study, the requirement of methodology and accuracy, tireless scientific guidance for the critical review, comments, clarifications, opinions and suggestions, for letting indication of relevant literature, the accessibility, warmth and friendliness shown by the confidence that has always given me and for following along this journey.

To all the Professors who followed my academic journey, for what they taught me and mainly by research methodology that instilled me.

To my mother and father for the solid education given till my youth, what provided the continuity of studies till this Masters.

To all my family members, in particular to my aunt, Carlota Cuambe, for the unconditional support during this master.

To my sister, Nércia Grete Cuambe.

To the friends I have made throughout life, who always attended and supported me direct or indirectly, advising and encouraging me with affection and dedication.

Finally to my husband Dário Amade, I appreciate all your love, affection, admiration, and with the presence tireless who supported me throughout the period of preparation of this dissertation.

To all, I reiterate my appreciation and my eternal gratitude.

[1] Projeto PTDC/SEN-ENR/115974/2009

Resumo

A produção de energia elétrica é um assunto de grande interesse principalmente para produtores e distribuidores e tem grande impacto na economia nacional. Nesta forma e à escala nacional não é viável armazenar energia e é difícil estimar o seu consumo com boa precisão de modo a poder gerir eficientemente a relação entre procura e demanda, de forma a não gerar grandes desperdícios.

Sendo assim, investigadores de diversas áreas abordaram este assunto de forma a facilitar a tarefa das empresas produtoras de energia em ajustar os níveis de produção à demanda do consumo. Com o decorrer dos anos, foram testados vários algoritmos preditivos e as Redes Neuronais de Função de Base Radial (RNFBR) foram até hoje uma das abordagens mais testadas com resultados satisfatórios. O facto de a adaptação em linha não ser uma tarefa fácil nesta abordagem, levou à procura de novas formas de efetuar a previsão, prometendo resultados melhores ou tão bons quanto os das RNFBR conseguindo também superar as dificuldades encontradas pelas RNFBR na adaptação em linha.

O presente trabalho pretende introduzir uma nova abordagem, ainda pouco explorada, para a previsão do consumo de energia. As Máquinas de Vetores de Suporte por Mínimos Quadrados (MVSMQ) poderão ser uma alternativa em relação às RNFBR e a outras abordagens, uma vez que têm muito menos parâmetros a ajustar, o que pode diminuir consideravelmente a sensibilidade daquelas máquinas aos problemas bem conhecidos relacionados com o ajuste de parâmetros, e tornar a adaptação em linha mais estável ao longo do tempo.

Palavras chave: Previsão do consumo de energia, máquinas de vectores de suportes de mínimos quadrados, função densidade de probabilidade, métodos de pruning, adaptação em linha.

Abstract

Electricity Load Demand (ELD) forecasting is a subject that is of interest mainly to producers and distributors and it has a great impact on the national economy. At the national scale it is not viable to store electricity and it is also difficult to estimate its consumption accurately enough in order to provide a better agreement between supply and demand and consequently less waste of energy.

Thus, researchers from many areas have addressed this issue in a way to facilitate the task of power grid companies in adjusting production levels to consumption demand. Over the years, many predictive algorithms were tested and the Radial Basis Function Artificial Neural Network (RBF ANN) was up to now one of the most tested approaches with satisfactory results. The fact that the on-line adaptation is not an easy task for this approach, led demand for new ways to make the prediction, promising better results, or at least as good as those of RBF ANN, and also the ability to overcome the difficulties founded by RBF ANN in on-line adaptation.

This work aims at introducing a new approach still little explored for electricity consumption prediction. Least-Squares Support Vector Machines (LS-SVMs) are a good alternative to RBF ANN and other approaches, since they have fewer parameters to adjust, hence, allowing significant decrease in the sensitivity of those machines to well-known problems associated with parameter adaptation, making the on-line model adaptation more stable over time.

Keywords: Electricity load demand forecasting, least-squares support vector machines, probability density function, pruning methods, on-line adaptation.

1 Introduction

Over the years, electricity consumption and its cost have been increasing significantly, thus, raising major concerns to both producing and supplying companies, and consumers. Since electricity cannot be easily and efficiently stored and conserved, the supplying and producing companies draw special attention towards the forecasting systems because they want to produce and/or buy not more or less than what they need to supply, and be able to avoid waste and lack that result from non-satisfaction of their customers' needs. As part of efforts to ensure that these aspects are taken into consideration, researchers from various fields of knowledge joined efforts and created ways to improve the agreement between the production and consumption of energy. The prediction of the electricity consumption is one of those ways.

The problem has gained even more relevance and this task became harder with the increase of micro-generation because it became more difficult to optimise the management between micro-generation, power-plants production and importation of electricity. Over time, this task became harder because of the increase in electricity consumption, derived from the increase of electric and electronic equipments usage, the global warming and the proliferation of the micro-generation of energy. Furthermore, another factor that hampers the achievement of results closer to reality is the fact that in many tested approaches, other variables that contribute to the variations of the electricity consumption like temperature, humidity, season of the year, weekday, holidays and others, are not taken into account.

1.1 Framework

This work will be conducted on the framework of a research project, funded by *Fundação para a Ciência e a Tecnologia* (FCT)[1] in collaboration with *Rede Elétrica Nacional* (REN). The goal of the project is the improvement of the accuracy of the predicted electricity consumption profile at the Portuguese national scale, by means of computational intelligence methods.

The requirement is an ELD forecast within an horizon of 48 hours in order to identify the need of reserves to be allocated in the Iberian Market.

[1] PTDC/SEN-ENR/115974/2009

1.2 Situation at REN

Currently, the approach used at REN consists of a RBF ANN one-step-ahead predictive model, executed at all half hours, which is iterated to obtain the predictive consumption profile for the next 48 hours. This model was identified using Multi-Objective Genetic Algorithms (MOGA) to select the inputs and the number of neurons to use [5, 8, 9]. The model parameters were determined by the Levenberg-Marquardt (LM) algorithm using a modified training criterion [4, 5, 7]. As the consumption time-series varies with time, the use of this approach requires retraining the model or the on-line adaptation of the model parameters [6, 10].

1.3 Proposal of this dissertation

The motivation for this work is to improve the ELD predictive performance either by improving existing Radial Basis Function (RBF) models or by the application of Least-Squares Support Vector Machines (LS-SVMs). Additionally, as RBF Artificial Neural Networks (ANNs) are, to some extent, difficult to adapt on-line, another goal is to conclude if the LS-SVM is more adequate in this respect. As sparse LS-SVMs are usually preferable to non-sparse LS-SVMs, it will be ascertained if an information-theoretic algorithm is efficient in providing sparseness to the LS-SVM model.

1.4 Main contributions of the thesis

The use of LS-SVM will bring a new way of looking to on-line adaptation without using computationally expensive models. This will be a big step on on-line adaptation models because of the avoidance of difficult retraining procedures or frequent readjust of model parameters. Also the application of LS-SVM models to iteratively obtain a multi-step forecast of ELD has not been done.

This will make that the goals of REN, or other companies who will choose to use this methodology, are achieved faster because LS-SVM models will probably require less maintenance over time when compared to other optimised models.

The introduction of an information-theoretic criterion based on multivariate kernel density estimation to provide sparseness to the LS-SVM model will be an original contribution of this work.

An efficient way has been found in order to initialize the hyper-parameters of LS-SVMs. Its

efficiency has been demonstrated experimentally.

1.5 Dissertation outline

The following chapter presents the state-of-the-art of ELD forecasting using RBF ANNs and LS-SVMs. Chapter 3 will show the methodologies employed in this dissertation regarding LS-SVMs. In chapter 4 the research methodology which was followed in this dissertation regarding ELD forecasting will be presented. The remaining of the dissertation follows with the description of the experiments and their objectives in chapter 5, the presentation of corresponding results in chapter 6, and finally the conclusions in chapter 7.

2 State-of-the-art

This chapter will present the main concepts addressed in this master thesis, focusing in:

1. ELD forecasting in general;
2. RBF ANNs applied to ELD;
3. LS-SVM applied to ELD.

2.1 Electricity load demand

In the past years, ELD forecasting conquered a big interest from power grid companies and researchers from different areas. For power companies, this is a big issue because they need to estimate the amount of electricity required to satisfy their customers. For them, this is not an easy task because the electricity demand has been increasing over the years and because electricity consumption patterns vary with many factors including time. Therefore, one of the goals of power companies is to get forecasts very close to the reality in order to prevent lack or waste of electricity. Forecasting became more difficult because electricity consumption varies over the years depending on factors like weather. Recently, the introduction and massification of the use of new electric and electronic technologies, the global warming and the micro-generation of energy, made this task even harder.

Finding the best forecast methodology is not an easy task because many variables, like temperature, humidity, wind, demographics, average number of domestic electric appliances, season of the year, day of the week, holidays, among others, have to be taken into account. Nevertheless, many studies are not taking some of these variables into account, therefore obliterating the chances of improving results.

Historical data may be of extreme importance in demand forecasting and its preparation is an important aspect [1]. The data is provided by power companies and the preparation is a task for the researchers, which is specific to the methodology each researcher will apply.

Demand forecasting is concerned with the prediction of hourly, daily, weekly and annual values of consumption and peak demands [16]. These forecasts are categorised in general as short-term, medium-term and long-term forecasting depending on the prediction horizon [1].

1. Short-term forecasting is normally made from few hours to a few weeks ahead.

2. Medium-term forecasting is made in the range of a few weeks to a few months and even up to a few years.

3. Long-term forecasting is made from 5 to 25 years. This type of forecast is important in deciding on the system generation and transmission expansion plans.

In short-term load forecasting, generally, weather conditions (particularly temperature) have significant influence on peak loads, and in long-term forecasts economic factors play an important role [12].

Forecast methods may be broadly classified into qualitative and quantitative techniques [18]:

1. Qualitative methods are sometimes referred to as subjective or judgemental methods. They are based on intuition and opinion and may or may not depend on past data. Generally they are used when data is limited, unavailable or not relevant. To use this type of forecast in the best way, the forecasters must have experience and skills on the subject and on available information. Consumers opinion can also be taken into consideration.

2. Quantitative methods are based on statistical or mathematical approaches. These methods depend on historical data and can be grouped into several types :

   (a) Causal methods are based on the identification of input variables that can predict values of the output variable in question. To use this type of method, the forecasters should have high statistical skills and large data requirements. These methods tend to work best for revenues that are heavily influenced by economic factors, such as business license fees, income taxes, and retail sales taxes.

   (b) time-series models are the most frequently used among all forecast models. They use historical previous data as a forecast basis. In this type of method, the forecasters make the assumption that previous data can be used to forecast new data and factors that had many influences in the past will have the same influences in the future. For the best use of this method, is important to know how long the time-series data is required to identify patterns. It is important that the data covers at least a period of several years and include some observations of the variation of data over the years.

   (c) Neural Network models. Bishop [2], considered interesting the use of neural networks for forecasting problems. Neural networks use a system of highly interconnected nodes or neurons, resembling biological neural networks, which may be able

to perform complex operations. The simplicity of designing a neural network and train it, attracted many researchers. Despite that, many times the training can take long time but usually achieving better results.

2.2 Radial basis functions artificial neural networks applied to electricity load demand forecasting

The application of RBF ANNs ELD forecasting have attracted the attention of researchers over the past decade, including researchers from REN and the University of Algarve (UAlg).

Mamun and Nagasaka [20] presented the use of RBF ANNs in a long-term prediction model, in a period where the electricity demand increased in a considerable way (average of 3% for year). The forecasting was made in Japan, where 9 interconnected power companies are operating. They made a careful selection of the economic factors to use as inputs and also used two types of data (monthly and yearly). The data was obtained from all the power companies from the year 1975 to 2000. They presented two types of forecasting. Monthly forecasting for the years 2001 and 2010 and yearly forecasting for the years 2001 to 2015.

For the training set, the root mean squared error (RMSE) was 0.138 MegaWatts (MW), the mean absolute error (MAE) was 0.17 MW and the mean absolute percentage error (MAPE) was 1.150%. For the test set, the RMSE was 1.056 MW, MAE 0.999 MW and the MAPE was 3.465%. They conclude that, the average annual incremental rate was about 1.3% up to year 2015.

Ghods and Kalantar [12] presented the use of RBF ANNs to long time forecasting in Iran, where 16 interconnected power companies are operating, taking past and present economic situations and power demand into account. The predictions were done for target years 2007 to 2011 and the training set was taken from 1989 to 2006.

The least-squares error (LSE) for the training set was 4.45% after 2087 iterations. These studies also determined that loads are increasing with an average annual incremental rate of about 5.35% up to year 2011.

These teams [12, 20] used similar approaches and their results showed that long-term forecasting is not very reliable because an alteration on the used input variables can cause a big impact on the results. The long-term forecasting can be made to get a rough idea of the electricity consumption but one must create an alternative way to get more precise predictions to avoid major disruptions.

Gontar, Sideratos, and Hatziargyriou [14] used RBF ANN for a hourly forecasting of the next 24 up to 48 hours ahead, a situation similar to that of REN and this work.
They used two distinct architectures. The first one was developed in the National Technical University of Athens (NTUA) and consists in five RBF ANNs which accept the same input vector. Four of them have the same structure and each one is trained with load time-series corresponding to each season of the year, in order to give a better prediction in the period for which they have been trained. The fifth network is trained with the load values that correspond to weekends and special days. The second architecture was developed in the University of Londz (UL) and consists in a parallel model of 24 equations (for 24 hours ahead, 1 equation for each hour). The equations were modelled using separate neural networks with the same structure. In these models, holidays are treated like Sundays, the days after holidays like Mondays and the days two-days after holidays are treated like Tuesdays. The presented results showed that for the 24 hours-ahead model developed by NTUA the MAPE was $5,77\%$ while for the UL was 4.94%. For the 48 hours-ahead, NTUA presented a MAPE of 6.08% and UL 5.23%.
The strategy used for the creation of both models takes into consideration important variables for the electricity consumption prediction like season of the year, weekends and holidays. These variables contribute to the satisfactory results they got and expected with the use of RBF ANNs.
The UAlg team, also presented results on the application of RBF ANN to ELD forecasting. Those results were presented in three distinct research documents [8, 9, 10]. The work consists in considering four approaches:

- Obtain a predictive profile up to the specified prediction horizon using one-step-ahead predictive models that are iterated in a multi step fashion.
- The necessity to employ on-line model adaptation strategies, as the profile of electricity consumption tends to vary over time.
- The need to incorporate one input in the models to account for the effect of events that dramatically perturb the typical profile of load demand.
- Addressing the problem of model structure optimisation in the system identification methodology in order to meet the specified design requirements.

In summary the methodology consists in: train the ANN models using the LM [19, 21] algorithm using the modified training criterion, and the model structure (number of neurons and

input terms) is evolved using a MOGA. The data set used for the model identification experiments corresponds to the Portuguese electrical energy consumption measured at hourly intervals from around mid October 2007 to the end of 2008. For the model retraining experiments, the data set used was from 2001 to September 2010.

For the selected model, an analysis in an unpublished report regarding the model operation at REN between May 2010 and October 2011, showed that the MAPE was 3% for 24 hours ahead prediction horizon and 4% for 48 hours ahead.

2.3 Least-squares support vector machines applied to electricity load demand forecasting

The application of LS-SVMs to electricity consumption prediction has been barely explored. Chen et al. [3] were the only researchers who used models based on LS-SVMs and the Wavelet Transform (WT) for short term forecasting. The WT has been proposed for time-series forecasting but must be used in combination with other models such as Neural Networks. The forecasting was made in three steps:

1. Data preparation;
2. Feature selection by WT decomposition;
3. Forecasting model based on LS-SVM.

A radial basis kernel function with $\sigma = 0.005$ was used and the regularization parameter was $\gamma = 1000$. To verify this approach, the authors compared the prediction errors, number of correct prediction steps and computing time, with those of neural network predictors (in this case, Linear NN, RBF NN and back propagation (BP) NN). The comparisons were made with forecasting model for 3 days ahead. The same training set was used for all the NN predictors and results showed that WT-LS-SVM was faster and had a lower value of the average relative error ($\frac{|x_i - y_i|}{x_i}$) percentage (0.0191%).

3 Least-squares support vector machines

The Support Vector Machines (SVMs) are based on the learning theory and were created by Vapnik [31] in 1995 with the initial goal of solving binary classification problems. The initial idea was mapping an input vector $\boldsymbol{x}$ into a high dimensional feature space through non-linear mappings chosen a-priori and then use them to separate the hyperplane.

Over the years, SVMs were tested to solve regression problems. Smola et al. [25] proved that it can be done through the introduction of loss functions that can be quadratic, Laplace, Huber or ε-Insensitive as suggested by Vapnik which is an approximation of the Huber function and allows the achievement of a sparse support vector.

The construction of a SVM model for regression is made from,

$$f(x,w) = w^T \varphi(x) + b, \tag{1}$$

where $\varphi(\cdot)$ maps the data into a high dimensional feature space and w is the normal vector of the hyperplane. LS-SVMs are a least-squares version of SVM. They were developed by Suykens and Vandewalle [27] for solving pattern recognition and nonlinear function estimation problems. They work with a least-squares cost function and involve equality instead of inequality constraints as in the standard SVM, and therefore are easier to train.

The LS-SVMs were introduced to solve high computational work of the constrained optimisation programming problem found in the SVMs. However, the sparseness gained with the standard SVM is lost and the estimation of the support vectors is only optimal in the case of the Gaussian distribution of the error variables. LS-SVM uses the LSE function instead of the ε-insensitive error function used on the standard SVM. It makes the solution follow a linear Karush-Kuhn-Tucker (KKT) system instead of a computationally hard Quadratic Programming (QP) problem [29].

As LS-SVMs are the core methodology for this dissertation, their derivation is reproduced here from the work of Suykens, Lukas, and Vandewalle [28]. Let us consider a training set of N samples, $\{x_i, y_i\}_{i=1}^N$, with $x_i \in R^n$ and $y_i \in R$. In LS-SVMs for function regression the following optimisation problem is formulated:

$$min_{(w,\varepsilon)} J(w,\varepsilon) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{i=1}^{N} \varepsilon_i^2, \tag{2}$$

subject to the equality constraints

$$y_i = w^T \varphi(x_i) + b + \varepsilon_i,\ i = 1, ...N \tag{3}$$

This corresponds to a form of ridge regression [13]. From this, the Lagrangian is formed:

$$L(w, b, \varepsilon, \alpha) = J(w, \varepsilon) - \sum_{i=1}^{N} \alpha_i \{w^T \varphi(x_i) + b + \varepsilon_i - y_i\} \tag{4}$$

with Lagrange multipliers α_k. The conditions for optimality are,

$$\begin{cases} \frac{\partial L}{\partial W} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial L}{\partial \varepsilon_i} = 0 \rightarrow \alpha_i = \gamma \varepsilon_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + \varepsilon_i = 0 \end{cases}, \tag{5}$$

for $i = 1, ..., N$. After eliminating ε_i and w, the solution is obtained, which may be written as the following linear equations,

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{6}$$

where $y = [y_1, ..., y_N]^T$, $\vec{1} = [1; ...; 1]^T$, $\alpha = [\alpha_1, ..., \alpha_N]^T$ and $\Omega_{kj} = \varphi(x_k)^T \varphi(x_j)$ for $k, j = 1, ..., N$. As a result of Mercer's conditions there are certain kernel functions $\psi(\cdot, \cdot)$ such that:

$$\psi(x_k, x_j) = \varphi(x_k)^T \varphi(x_j),\ k, j = 1, ..., N\ , \tag{7}$$

which is also called the kernel trick in the literature.

Set A = $\Omega + \gamma^{-1} I$. For A, a positive-definite matrix, A^{-1} exists. Solving the linear equations we obtain the solution:

$$\alpha = A^{-1}(y - b\vec{1}) \quad ; b = \frac{\vec{1}^T A^{-1} y}{\vec{1}^T A^{-1} \vec{1}}. \tag{8}$$

Substituting w in (1) with the first equation of (5) and using (7) we have,

$$f(x, w) = y_x = \sum_{i=1}^{N} \alpha_i \psi(x, x_i) + b\ , \tag{9}$$

where α_i and b, presented in (8), are the solution to equation (6). The kernel function $\psi(\cdot, \cdot)$

can be chosen, among others, as:

(a) Linear function: $\psi(x,x_i) = x_i^T x$;

(b) Polynomial function: $\psi(x,x_i) = (\frac{x_i^T x}{c} + 1)^d$;

(c) Radial basis function: $\psi(x,x_i) = exp(-\frac{||x-x_i||^2}{\sigma^2})$.

3.1 Sparse and non-sparse least-squares support vector machines

The original LS-SVM, as presented above, does not yield sparse solutions as SVMs. This happens because on LS-SVM models, the solution is obtained using all the regressors. To keep all the advantages of LS-SVM over SVM and at the same time get sparse solutions, pruning methods can be used with the goal of finding the best set of regressors. This can be done by removing gradually less important data from the training set and re-estimating the LS-SVM. The suggestion from Suykens, Lukas, and Vandewalle [28] consists in:

1. Train the LS-SVM based on a set of N samples;
2. Decrease the amount of data regressors, for example 5%, having smaller values of $|\alpha|$;
3. Retrain the LS-SVM based on the reduced training set;
4. Go back to 2 only if the performance increases.

The α is mostly chosen because $|\alpha_i|$ support values are proportional to the errors at the data points ($\alpha_i = \gamma\varepsilon_i$).

3.2 On-line adaptation of least-squares support vector machines

When a RBF kernel function is used, the LS-SVM has only two parameters, σ and γ, also called hyper-parameters.

The γ is used on the optimization problem formulation given by (2), and σ controls the width of the Gaussian kernel function. As these parameters influence the results obtained they should be optimized in some way. There are two classes of methods for the estimation of these parameters:

- Experimental methods: in practice, most researches have been using cross-validation and searching on a (σ,γ) grid.

- Theoretical methods: global or local optimization methods can be used like genetic algorithms, simulated annealing and Bayesian inference framework.

In this work the ideas introduced in [33] are followed in order to optimize the hyper-parameters for a given data set. The cost function employed uses an estimate of the variance of the effective noise in a data set. The next sub-section presents the estimation method.

3.3 The Gamma Test

To evaluate the quality of a model to approximate an unknown function, the Gamma Test (GT) can be used [17]. For building a model, we need a N-sample data set of the following type:

$$D = \{(x_i, y_i)_{i=1}^{N}\} \ , \tag{10}$$

as presented earlier, now denoted as D.
In practical terms, the GT allows the estimation of the effective noise variance in y by means of the data set D. The effective noise variance, denoted by Γ, may be computed as follows:

1. Compute $\delta_N(k)$:

$$\delta_N(k) = \frac{1}{N}\sum_{i=1}^{N} ||x_{i,k}^{v} - x_i||^2 \ , \tag{11}$$

   where $||\cdot||$ represents the Euclidean distance between $x_{i,k}^{v}$, which is the k^{th} nearest neighbour of x_i, and x_i. With that, we get the sequence of points,

$$\{\delta_N(1), \delta_N(2), ..., \delta_N(p)\} \ , \tag{12}$$

   where p is usually 10.

2. Compute $\beta_N(k)$:

$$\beta_N(k) = \frac{1}{2N}\sum_{i=1}^{N} (y_{i,k}^{v} - y_i)^2 \ , \tag{13}$$

   where $y_{i,k}^{v}$ represents the corresponding output of $x_{i,k}^{v}$. Note that $y_{i,k}^{v}$ is not necessarily the nearest neighbour of y_i. From here, we get the sequence of points,

$$\{\beta_N(1), \beta_N(2), ..., \beta_N(p)\} \ . \tag{14}$$

3. With the pairs of points from (12) and (14),

$$\{\delta_N(k), \beta_N(k)\}_{k=1}^{p} , \tag{15}$$

using linear regression, we obtain a linear approximation:

$$\beta_N(k) = \Gamma + A\delta_N(k) , \tag{16}$$

where Γ is the estimation of the effective noise in y and the value of A gives information about the complexity of the function to be modelled. Any model that produces a mean squared error less than Γ will be, in principle, modelling the noise component of y. Also, any model whose mean squared error does not approximate to Γ will not be modelling all the dynamic features of the process or system being modelled that are present in D.

3.4 Objective function for optimization of the hyper-parameters

Based on [33] and using the data set in (10), the LS-SVM can be presented as:

$$\hat{y} = f(X, \gamma, \sigma). \tag{17}$$

Using Γ in (16), an objective function can be formulated, that depends on γ and σ. The goal is to optimize that objective function using gradient based methods. For a given model trained using the set D, the training error ε should have a variance Γ_T near Γ, so that ε is only the noise component in X. If $\Gamma_T > \Gamma$, the model does not represent completely the system, in other words there is in ε a dynamic component of the system beyond the noise. If the opposite is found, ($\Gamma_T < \Gamma$), the model reflects, beyond the system features, also noise features, which can be due to over-training. The error variance is given by:

$$\Gamma_T(\varepsilon) = \frac{1}{N}\sum_{i=1}^{N}(\varepsilon_i - \bar{\varepsilon})^2 , \tag{18}$$

where $\bar{\varepsilon}$ is the average error (ε). If we assume that $\bar{\varepsilon} = 0$,

$$\Gamma_T(\varepsilon) = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2 . \tag{19}$$

According to equation (5) of a LS-SVM, $\varepsilon_i = \frac{\alpha_i}{\gamma}$. Replacing this in (19) we obtain:

$$\Gamma_T(\varepsilon) = \frac{1}{N}\sum_{i=1}^{N}\frac{\alpha_i^2}{\gamma^2} = \frac{\alpha^T\alpha}{N\gamma^2}. \tag{20}$$

The following objective function can now be defined:

$$\mathscr{O}(H) = |\Gamma_T(\varepsilon) - \Gamma| = |\frac{\alpha^T\alpha}{N\gamma^2} - \Gamma| \,, \tag{21}$$

where H is the vector of parameters to be optimized, defined as,

$$H = [\gamma, \sigma] \,. \tag{22}$$

With that, (17) can be rewritten as,

$$\hat{y} = f(x, H) \,. \tag{23}$$

In order to use gradient based methods to incrementally optimize $\mathscr{O}$, the partial derivatives of (21) to γ and σ are required. These are given in [33] and are not reproduced here as the presentation would be extensive.

3.5 The gradient descent method

In this dissertation a simple Gradient Descent (GD) method will be employed to optimize the cost function presented in (21) in order to train LS-SVM models. Although several other methods are available, that in principle would be preferable, the focus of the work is not that subject. This will be compensated by executing different runs of the Gradient-Descent method with random starting points.

The goal of the gradient descent method is to find the nearest local minimum of a function, assuming that it can be calculated. The method proceeds from iteration k to k+1, taking steps of size ρ, according to the update rule:

$$H_{k+1} = H_k - \rho\Delta\mathscr{O}(H_k), \tag{24}$$

where ρ is the step and $\Delta\mathscr{O}(H_k)$ is the gradient in the iteration H_k. The update rule may be iterated until a given precision on the objective function is obtained or a certain number of iterations has been reached.

The following algorithm has been used:

Algorithm 1 The gradient-descent algorithm

θ ▷ Precision
NI ▷ Maximum number of iterations
ρ ▷ Gradient method step size
Γ ▷ Estimated noise variance
$H_k \leftarrow (\sigma, \gamma)$ ▷ Initial parameter vector
$f(H_k)$ ▷ Compute the LS-SVM for H_k to obtain the value of α
$\Gamma_T \leftarrow \frac{\alpha^T \alpha}{N\gamma^2}$ ▷ Error variance of LS-SVM for H_k
$k \leftarrow 0$
while $|\Gamma_T - \Gamma| > \theta$ **and** $k < NI$ **do**
    $\Delta\mathscr{O}(H_k)$ ▷ Compute gradient
    $H_{k+1} \leftarrow H_k - \rho\Delta\mathscr{O}(H_k)$
    $f(k_{k+1})$ ▷ Compute the LS-SVM for H_k to obtain the value of α
    $\gamma \leftarrow H_{k+1}[0]$
    $\Gamma_T \leftarrow \frac{\alpha^T \alpha}{N\gamma^2}$
    $k \leftarrow k + 1$
$\gamma \leftarrow H_k[0]$ ▷ Optimized value of γ
$\sigma \leftarrow H_k[1]$ ▷ Optimized value of σ

4 Research methodology

4.1 Advantages and disadvantages of the chosen methodology

The major disadvantage of using LS-SVMs is the difficulty of the algorithm to present sparse solutions since it uses all the training set to implement the network.

Besides that, there are many advantages on the use of LS-SVMs, the major one being the fact that the algorithm adjusts the support vectors automatically, therefore it only has to calculate the neuron propagation.

For the ELD prediction application, another significant advantage comes from the fact that, when using the Gaussian kernel, LS-SVMs have only two parameters, therefore it is expectable that they will be easier to adapt on-line when compared to other models with large number of parameters.

4.2 Technical approach and data used

This work will use LS-SVMs and RBF ANNs to obtain one-step-ahead predictive ELD models and employ these models to generate a forecast of the consumption profile up to a 48 hours prediction horizon.

The historical data set was provided by REN and corresponds to the period from 2001 to 2011 in Portugal. The data was scaled and/or normalised to avoid working with different ranges among different variables. The sampling time is one hour, which results in 48 prediction steps to reach the 48 hours prediction horizon. Figure 4.1 shows a plot of the entire data set and examples of a 24 hours period and a week period.

After implementation and validation of algorithms and computational methods the research methodology will consist in using simulation to produce a number of comparable experimental results generated by the following different ELD predictive strategies:

- Static RBF ANN as identified off-line;
- Static non-sparse LS-SVM;
- Static sparse LS-SVM;
- On-line adapted RBF ANN;
- On-line adapted LS-SVM (the best of off-line sparse and non-sparse).

The results will allow the performance analysis of the methods and validation of contributed methods. In order to enable the comparison of the different methodologies, the models will be determined using the same data set. They will be validated using a different data set from the one used for the identification and the data sets will be the same for the different models. The evaluation of the models will use the same error metrics, for example the MAE and the MAPE. These error metrics are analysed along the prediction horizon, so the models are ultimately compared from the metric curves along the horizon.

The algorithms will be implemented in the Python language because it is a high-level and high performance free programming language. It is easy to use and we can explore the Python/C API, in this case, to write extension modules in C that can be used with the Python language. Modules are also available to enable distributed implementations of algorithms over multiple processors or processor cores.

4.3 Additional methodologies employed

Besides the LS-SVMs, other methodologies have been used. These are described in following subsections.

4.3.1 Multi-objective genetic algorithms

Genetic algorithms are an evolutionary computation approach that performs a search based on a population through a set of genetic operators such as, selection, crossover and mutation.

The MOGA [11] has been used in this work in order to evolve suitable RBF ANN ELD predictive models. Specifically it was used to select the number of neurons and inputs of models so that the model performance is maximized. The approach has been developed within the UAlg team.

In this work a Python implementation has been done considering a reference Matlab implementation. A direct translation has been accomplished where Matlab C Mex files have been translated to Python modules using the Python/C API.

4.3.2 Radial basis function artificial neural networks

The ANNs are used in many courses as neuroscience, mathematics, statistic, physics, computer science and engineering to solve forecasting, optimization, pattern recognition, associative memory and other problems. Haykin [15] define neural networks as a massively parallel

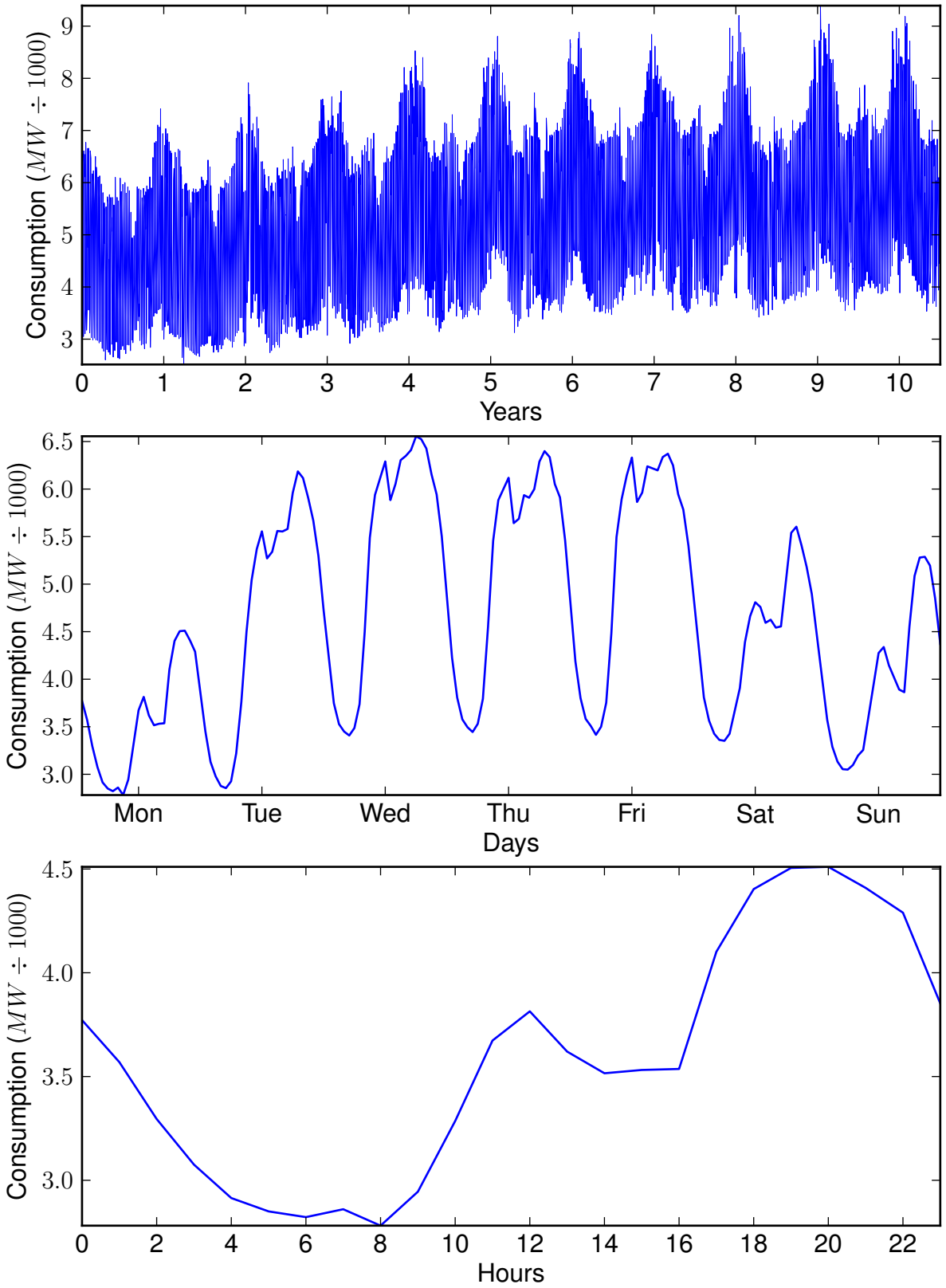

Figure 4.1: Top: Plot of the entire data set. Middle: An example of one week period. Bottom: An example of a 24 hours period.

distributed processor made up of simple processing units to store experimental knowledge and making it available for use.

ANNs can be seen as weighted directed graphs where the neurons are nodes and the directed edges (with weights in each) are connections between input and output neurons. They are used for non-linear mapping between the input data X and the output vector y in order to model relations or detect patterns between them.

The RBF neural network consists in three layers fully interconnected:

- The input layer that has nodes to connect the inputs to the network;
- The hidden layer, normally high dimensional, where each unit has an activation function;
- The output layer is a weighted linear combination of the activation functions of the hidden layer.

RBF ANNs are a type of ANNs that use radial basis as an activation function [22]. They were designed to solve classification problems but studies showed that they can also be perfectly used to solve regression and time-series problems [26].
Designing RBF ANNs results from the achievement of the following steps:

- Selecting the number of neurons of the hidden layer;
- Selecting the relevant inputs to the network;
- Adjusting the coordinates of the centre of each radial basis function of the hidden layer;
- Adjusting the weights applied to the radial basis functions output;
- Adjusting the radius of each radial basis function.

A RBF ANN is mathematically defined as:

$$f(x_k) = \sum_{i=0}^{N} \alpha_i \psi_i(x_k) \; ; \; \psi_0 = 1 \; , \tag{25}$$

where $\psi_i(x_k)$ is, for instance, a radial basis function given by

$$\psi_i(x) = \exp\left(-\frac{||x - c_i||^2}{2\sigma_i^2}\right) \; , \tag{26}$$

where c_i is the i^{th} centre and $||\cdot||$ is a norm, usually Euclidean.

The design approach developed by the UAlg team consists in using the MOGA to choose RBF ANNs where the number of neurons and selected inputs optimize predefined goals [5]. During this optimization, the RBF ANNs are trained using the LM algorithm minimizing a criterion that reflects the RBF non-linear/linear topology [4, 7].

4.3.2.1 Comparing radial basis function artificial neural networks to least-squares support vector machines

RBF ANNs present good results concerning prediction but they have a slow design process which implies selecting the structure to use including the inputs, number of neurons and all the parameters (weights, centres and widths) of the network in a way to maximize performance. The on-line adaptation is complicated because it is necessary to update all the model parameters over time. If adaptation is not done, the model deviates from the modelled system which may cause the accumulation of errors over time to be very large, therefore making the model unreliable. Comparatively, in LS-SVMs the algorithm adjusts automatically the support vectors, leaving only two parameters to design (assuming a Gaussian kernel). This decreases the computational complexity of the design process and possibly makes the on-line adaptation more stable and reliable over time.

The LS-SVM looks promising for the ELD prediction problem due to three main reasons:

- They have been successfully applied to the prediction of time-series [30, 32];
- Their design does not require computationally expensive identification experiments in order to obtain an optimised model;
- They are expected to present improved results and more stable behaviour regarding the on-line adaptation of the model;

4.3.3 Multi-variate kernel density estimation

In a following subsection an algorithm will be presented in order to select a data subset using information quantification.

To quantify the information in a data set organized in multi-dimensional points we need to estimate the probability density function (PDF). One family of PDF estimation methods is the kernel density estimation. The probability of each point x in a data set of N points using this

method is given by [23, 24]:

$$Pb(x) = \frac{1}{n}\sum_{i=1}^{N}\left[\prod_{k=1}^{d} K_{h_k}(x^{(k)} - x_i^{(k)})\right] \tag{27}$$

where $K(\cdot)$ is a kernel function, d is the sample dimension, and h_k is the bandwidth in dimension k. The superscript (k) indicates the dimension k of points.

When we are using a Gaussian kernel, h_k corresponds to the spread of the Gaussian kernel.

For the estimation of h_k the simple rule introduced by Scott [23] has been adopted:

$$h_k = \hat{\sigma}_k N^{-\frac{1}{d+4}}, \tag{28}$$

where $\hat{\sigma}_k$ is the standard deviation of dimension k.

For the application of ELD forecasting, the points x, or regressors, will be very similar along their d dimensions, which means that the d distinct parameters will be extremely similar, therefore allowing the simplification:

$$h_k = h = \hat{\sigma} N^{-\frac{1}{d+4}}, \tag{29}$$

where $\hat{\sigma}$ is estimated on the entire data set.

4.3.4 Pareto efficiency analysis

The goal of multi-objective optimization is to find a set of acceptable solutions and the concept of Pareto optimally or Pareto dominance is introduced to compare those candidate solutions. For that reason, it can be said that a solution belongs to the Pareto set or Pareto frontier if there is no other solution that can improve at least one of the objectives without degradation of any other objective.

The Pareto frontier can be chosen according to two different goals, maximize or minimize the objectives. In the context of this work the performance of different models has been analysed according to a few pairs of error metrics, (m_1, m_2), which form Pareto fronts where the goal was to minimize. Therefore the selection of models has been done by analysing the trade-offs along the Pareto fronts. The following simple procedure has been used:

Algorithm 2 The Pareto front algorithm

```
function PARETO(P)        ▷ P is an array of points (m1,m2) sorted in ascending order of m2
    P' = {}                                                          ▷ Pareto set
    N                                                                ▷ Dimension of P
    for i = 0 → N do
        (m11,m12) ← P[0]                          ▷ First pair of P has lowest value of m2
        P' ←add (m11,m12)
        REMOVE (m11,m12) from P
        while any (m1,m2) in P has m1 >= m11 and m2 >= m12 do
            REMOVE (m1,m2) from P
    return P'                                    ▷ P' has the points in the Pareto front
```

4.4 Subset selection based on information quantification of data

Assuming X is a random variable, each point or outcome $x_i \in X$ has a quantity associated, called self-information, defined as,

$$I(x_i) = -log(Pb(x_i)) \tag{30}$$

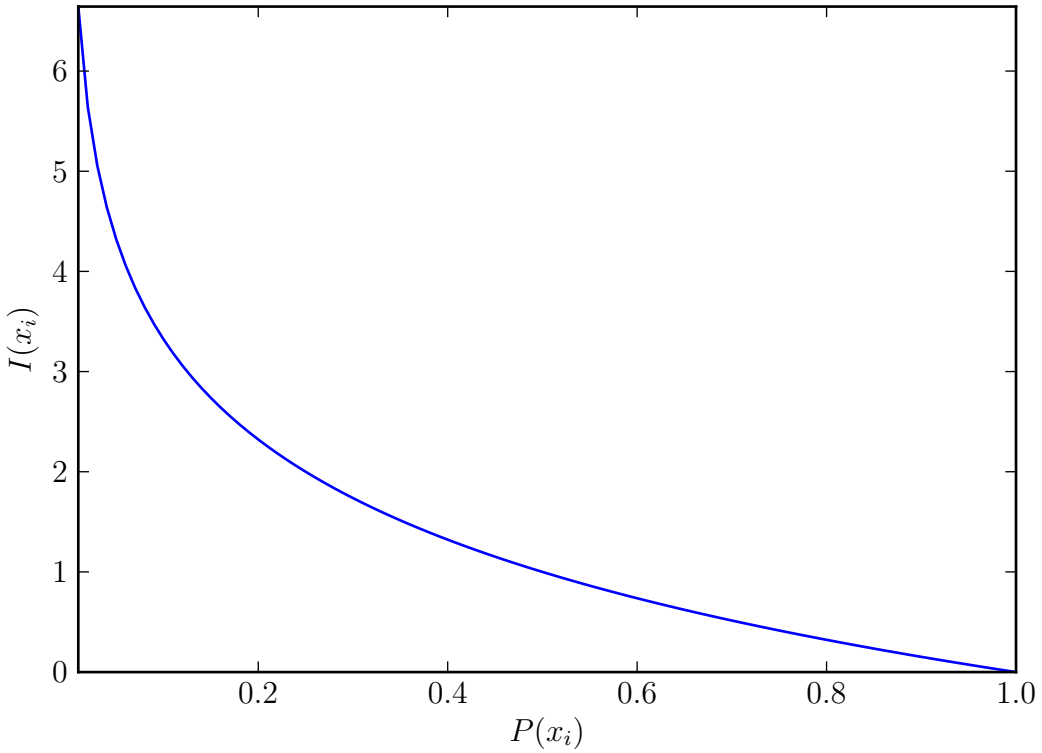

Figure 4.2: Relation of $I(x_i)$ to $Pb(x_i)$ (considering base 2 logarithm function).

Figure 4.2 presents the relation of $I(x_i)$ to $Pb(x_i)$.

In this work we will analyse an algorithm in order to select a subset of X according to some

proportionality of $I(x_i)$. The main idea is to select points to train a model that have an appropriate information content. This does not mean we should select the subset with the highest sum of information because in that way only less frequent points would be selected and the resulting model would be oblivious to that input region.

In order to maintain some diversity and still maximize information content we will sample X proportionally to $I(x_i) \forall x_i \in X$, i.e., the probability of selecting one point is proportional to $I(x_i)$. One way of making such selection is by using Stochastic Universal Sampling (SUS) using $I(x_i)$ as the fitness indicator. Such subset selection strategy may be applied to pruning LS-SVMs, besides other applications, with potential advantages:

1. The pruning is done a-priori, therefore saving computational time;

2. If a Gaussian kernel is used to estimate $Pb(x_i)$ the kernel values can be stored in memory as they will be useful for the LS-SVM (assuming a common value of h and σ has been found, that serves well both methods);

3. There is an objective of maximizing information content of the regression matrix, for a given degree of diversity, which might translate to better performing models.

5 Experiments and objectives

Several experiments were carried out in order to select the best LS-SVM modelling strategy in what concerns some aspects:

- Choosing the regressor type;
- Choosing dimensions for the regression matrix;
- Selecting a pruning strategy and the percentage of regressors to prune;
- Selecting a suitable model adaptation strategy.

These experiments will be described in following sections.

5.1 Regressors, data set and model evaluation

Consider the data set presented in (10) and the model relationship in (17). X is a $(N \times d)$ matrix where each row is a vector x_i of the form:

$$x_i = [y_k\ , y_{k-1}\ , \cdots , y_{k-d}]\ . \tag{31}$$

In order to build a LS-SVM model to relate $\hat{y}_i \equiv \hat{y}_{k+1}$ to x_i, the values of N (size of regressive data window) and d (size of each regressor) must be decided.

In some experiments an additional value was appended to x_i, which is an encoded number with a different value for the day of the week of y_k in x_i. This value also encodes the event of y_k belonging to a holiday. The complete description of the encoding scheme can be found in [9]. In this case each regressor has the form,

$$x_i^* = [y_k\ , y_{k-1}\ , \cdots , y_{k-d}\ , \zeta_k]\ . \tag{32}$$

The experiments employed one of the types of regressor, x_i or x_i^*.

The data set used to build each model was composed of the first 15 months of data, starting in January, 2001. The last 3 months were used for testing purposes, and the first 12 were to train the model.

For each modelling experiment the following results were collected: the training error (ε_t), testing or generalization error (ε_g) and two quantities related to the model predictive performance, denoted as ε_p and ε_p^*, defined as follows [5]:

Consider X_g is the regression matrix built from the test set. For each row the model is simulated in order to obtain the ELD predictive profile up to 48 hours ahead. Using these predictions an error matrix is constructed as:

$$E(X_g, ph) = \begin{pmatrix} e[1,1] & e[1,2] & \dots & e[1,48] \\ e[2,1] & e[2,2] & \dots & e[2,48] \\ \vdots & \vdots & \ddots & \vdots \\ e[N-48,1] & e[N-48,2] & \dots & e[N-48,48] \end{pmatrix}, \tag{33}$$

where $e[i, j]$ is the model predictive error taken from instant i of X_g, at step j within the prediction horizon. ε_p is the MAE for each instant j and is defined as,

$$\varepsilon_p(j) = \frac{1}{N-48} \sum_{i=1}^{N-48} |e[i, j]| , \tag{34}$$

and ε_p^* is the sum of $\varepsilon_p(\cdot)$ over the prediction horizon, defined as,

$$\varepsilon_p^* = \sum_{j=1}^{48} \varepsilon_p(j) . \tag{35}$$

5.2 Choosing the regressor, N and d

The first two sets of experiments were conducted in order to achieve two conclusions: to find the best regressing strategy and to decide if a second input variable was beneficial or not.

In the two sets of experiments N took discrete values corresponding to using between 6 and 12 months, i.e., $N \in \{6,7,8,9,10,11,12\}$. For d, corresponding to days, the values were $d \in \{1,2,3,4,5,6,7\}$. The actual values of N are multiplied by 31×24 and those of d are multiplied by 24. The number of months (for N) and days (for d) will be used for simplicity. Each set of experiments employed one of the regressors, x_i or x_i^*, and will be denoted by $E1$ and $E2$ respectively. Within $E1$ and $E2$ a number of modelling experiments were executed for each combination of $\{N, d\}$.

For each $\{N, d\}$ combination 11 modelling trials were executed, where the difference was the initial value of the hyper-parameters σ and γ. Using trial and error, the ranges to select random starting points for 10 of the modelling trials were defined as $[\frac{h}{5}, 2 \times h]$ for σ (h given by (29)) and $[0.5, 2.0]$ for γ. The 11^{th} trial was initialized with $\sigma = h$ and $\gamma = 1.0$. For each trial, algorithm 1 was executed using the parameters $\theta = 0.0001$, $\rho = 1.0$, $NI = 15$.

These two set of experiments, $E1$ and $E2$, allowed the selection of the regressor type and to restrict the set of combinations of $\{N, d\}$ to those where results were better. In the remaining experiments that were executed in this dissertation only the selected regressor was used.

5.2.1 Choosing N and d

For the restricted set of $\{N, d\}$ combinations a new set of experiments was conducted, using only the regressor type that was already selected. The experiments were parametrised as $E1$ and $E2$ except for the number of modelling trials initialized randomly, now 50, and for NI set to 30.

The set of experiments was denoted as $E3$ and allowed selecting specific values for N and d that resulted in better models.

5.3 Pruning the models

With $\{N, d\}$ chosen on $E3$, pruning methods were applied for different pruning percentages varying from 5% to 50% in steps of 5%. Two pruning methods were employed, the first based on $|\alpha|$ as explained in section 3.1, and the second using the quantification of information as explained in section 4.4. The two sets of pruning experiments, grouped by the method used, are denoted as $E4$ and $E5$. Both were parametrised as $E3$ except the number of modelling trials initialized randomly, now 10, and the test set now including all the data until 2011.

These two experiments, $E4$ and $E5$, contribute to evaluate which pruning method and percentage achieve better results. From the selected method, two percentages were chosen: the first one is the percentage that achieved better results when all simulation data was considered and the second percentage was selected by analysing only the first three months of the test set. This way we have the pruning percentages that are best more locally or more globally in time. For these percentages, the best modelling trials were chosen for on-line adaptation.

5.4 On-line adaptation of the models

The simulations of the models chosen from $E4$ and $E5$ were made using all the test set with data until 2011. For that, two experiments were conducted using three periodicities of 7, 14 and 30 days, for model parameter adjustment.

For the experiment $E6$, σ was always initialized with h given by (29) and γ with 1.0, and for $E7$ σ and γ were initialized with values from the previous parameter adjustment.
The objective of both experiments was to choose the best initialization for the hyper-parameters and the best periodicity for the parameter adjustment.

5.5 Evaluating the existing RBF model

The model currently in use at REN was trained and evaluated using the same data set and adaptation periodicity as LS-SVMs, on experiments E6 and E7, to facilitate the comparison between both methodologies. The methodology used for RBF ANNs was the same as the one presented in the UAlg team paper [10].

6 Results and Discussion

In this chapter, the results obtained from the experiments presented on the previous chapter will be presented. For each experiment or set of experiments, figures and tables will be presented to help clarifying the ideas behind each decision and conclusion.

6.1 Choosing the regressor, N and d

Tables 6.1 and 6.2 show MAE values, in MW, obtained from the training set, using regressors x_i and x_i^*, respectively. Each group of three values represents, from top to bottom, the minimum, average and maximum obtained over the 11 modelling trials for the corresponding days/months combination. Bold values denote the smaller value between both tables. Tables 6.3 and 6.4 present the corresponding results obtained on the test set.

As we can see on tables 6.1 and 6.2, the number of smaller values are slightly predominant on table 6.2. The difference is not so great, which complicates a choice between regressors based on those tables. For that purpose, tables 6.3 and 6.4, with results from the test set, are more elucidating. There is a clear predominance of smaller values on table 6.4, showing that the regressor x_i^* improves the prediction and the capacity of generalization to data different from the training set. Because of this, the regressor x_i^* was chosen for the continuation of this dissertation.

From tables 6.3 and 6.4, we also conclude that the best results are obtained from $\{N,d\}$ combinations with larger N and lower d.

Tables 6.5 and 6.6 show values of MAE, in MW,using regressor x_i^*, obtained from the training set and test set respectively. Each group of three values represents, from top to bottom, the minimum, the value obtained by the model initialized by $\sigma = h$ and $\gamma = 1.0$, and the average obtained for the corresponding days/months combinations over the 11 modelling trials.

The values given by $(\sigma,\gamma) = (h,1.0)$, in the middle, are almost always near or even above the average on the training set as it may be seen on table 6.5. Despite that, the corresponding results obtained from the test set, are the smallest obtained, or very close to that, specially when we have lower d and greater N where the best results are found. For this, we can anticipate that, $(h,1.0)$ are good initializers for σ and γ, which, for the on-line application, can be significant since we don't need to search randomly for these initial values.

Figure 6.1 shows the average curves of evolution of ε_p, over the prediction horizon. On the left,

Months \ Days	1	2	3	4	5	6	7
	58.9	64.5	**69.0**	80.3	86.0	96.8	**106.9**
6	72.3	73.1	89.4	98.3	109.6	115.1	**120.4**
	85.9	94.5	109.0	129.7	133.9	141.4	162.4
	58.0	60.6	71.7	77.1	83.3	**95.9**	**99.8**
7	66.7	**68.9**	**77.5**	87.4	**103.0**	114.7	122.0
	74.1	80.9	**94.2**	99.4	130.4	144.2	147.2
	54.3	62.2	69.3	73.6	**81.2**	90.1	**97.3**
8	65.7	**68.7**	77.6	91.1	**92.0**	**103.7**	**113.6**
	92.1	**79.3**	**91.5**	117.8	**105.7**	**123.9**	**139.4**
	53.9	60.7	65.8	77.8	84.1	89.4	98.3
9	**65.2**	68.2	**72.1**	89.3	92.7	**106.5**	**113.2**
	74.5	81.7	91.4	113.8	**119.0**	**123.3**	**137.0**
	56.7	**59.5**	65.6	77.6	78.1	**83.9**	92.0
10	64.0	65.9	**70.8**	92.6	93.9	99.3	**108.8**
	76.6	74.6	**82.9**	106.3	109.0	124.7	**131.0**
	55.7	59.5	66.6	71.3	77.3	86.7	96.7
11	**62.6**	67.1	**71.7**	**81.3**	95.3	106.0	111.5
	75.5	85.3	**79.6**	**93.9**	112.6	124.4	136.6
	52.3	63.7	**64.8**	**67.6**	73.3	80.7	**90.0**
12	**62.5**	68.1	**73.1**	78.7	96.7	**94.2**	**103.9**
	75.2	80.0	**79.8**	92.9	116.1	**118.1**	**128.5**

Table 6.1: Values of training MAE, in *MW*, for regressor x_i. For each combination of days and months, the three values are, from top to bottom, the minimum, average and maximum obtained over the 11 modelling trials. Bold values indicate a smaller value when compared to the corresponding of table 6.2.

Months \ Days	1	2	3	4	5	6	7
	60.4	**56.5**	71.9	**77.9**	**84.2**	**94.7**	108.8
6	**65.5**	**72.1**	**85.7**	**89.2**	**106.3**	**113.4**	127.4
	77.9	**84.8**	**97.1**	**106.9**	**131.4**	**135.8**	**144.2**
	58.8	**59.5**	**67.0**	**75.5**	**82.0**	99.7	100.3
7	**65.5**	69.2	80.4	**85.1**	103.7	**114.6**	**119.8**
	83.3	**78.4**	100.1	**97.1**	**128.5**	**128.7**	**145.2**
	54.1	**59.7**	**63.9**	**72.8**	87.6	**88.6**	103.5
8	**61.4**	69.8	**74.3**	**89.2**	102.6	106.6	122.9
	73.2	85.5	92.0	**110.5**	120.6	133.2	148.3
	56.7	**59.8**	**64.2**	**75.7**	**77.7**	**88.1**	**95.4**
9	72.0	**64.1**	74.1	**84.7**	**92.2**	111.0	118.0
	122.4	**81.2**	**91.0**	**91.3**	122.0	134.9	144.5
	52.5	60.3	**63.9**	**68.7**	**76.0**	84.4	**91.4**
10	**59.8**	**63.4**	75.6	**81.2**	**87.7**	**99.0**	118.1
	81.0	**68.9**	93.1	**102.6**	**99.7**	**112.0**	142.2
	59.0	**55.0**	**57.9**	**69.9**	**76.4**	**83.1**	**90.9**
11	67.3	**64.2**	72.4	87.1	**88.5**	**98.1**	**104.1**
	83.7	**75.8**	87.3	104.9	**107.4**	**119.4**	**129.4**
	55.8	**59.9**	66.4	68.9	**72.7**	**80.4**	93.2
12	64.0	**65.6**	74.0	**75.4**	**84.0**	94.7	107.2
	90.4	**76.3**	90.6	**84.7**	**107.9**	118.6	132.4

Table 6.2: Values of training MAE, in *MW*, for regressor x_i^*. For each combination of days and months, the three values are, from top to bottom, the minimum, average and maximum obtained over the 11 modelling trials. Bold values indicate a smaller value when compared to the corresponding of table 6.1.

Months \ Days	1	2	3	4	5	6	7
	132.4	126.5	189.5	258.4	283.5	326.7	**350.1**
6	182.7	198.9	228.9	280.4	**311.4**	362.0	413.3
	272.2	260.3	260.6	**310.2**	359.0	**392.9**	457.0
	108.1	**115.4**	**159.5**	**225.7**	286.7	**323.4**	358.4
7	188.4	186.0	**205.4**	282.7	316.0	355.0	**395.8**
	287.6	265.9	**238.9**	**313.8**	**355.6**	393.0	**445.8**
	114.7	**120.4**	**145.7**	236.6	**263.1**	343.0	366.2
8	205.2	211.7	**221.9**	276.7	325.5	377.8	418.0
	299.7	286.5	267.0	318.9	360.4	**413.4**	460.4
	100.2	**98.1**	**118.3**	215.6	271.5	299.1	333.3
9	159.7	165.5	215.8	257.0	298.0	329.6	373.0
	216.6	255.8	253.5	284.0	**319.1**	**361.4**	**410.3**
	104.4	108.3	**104.6**	140.4	216.3	272.0	299.1
10	154.1	174.6	**165.1**	**205.7**	249.4	290.8	321.6
	228.8	237.7	213.7	**244.7**	**276.0**	**320.8**	**356.9**
	90.1	88.1	**98.3**	**128.6**	163.6	194.6	222.1
11	122.0	129.5	132.1	153.8	180.5	205.3	**228.7**
	141.1	168.3	160.3	169.6	192.0	217.0	**238.6**
	76.4	84.5	95.7	**98.8**	**144.7**	**175.0**	202.1
12	99.3	118.2	123.5	**133.6**	164.9	187.3	209.6
	114.7	147.6	133.1	156.2	**174.4**	199.8	220.3

Table 6.3: Values of testing MAE, in *MW*, for regressor x_i. For each combination of days and months, the three values are, from top to bottom, the minimum, average and maximum obtained over the 11 modelling trials. Bold values indicate a smaller value when compared to the corresponding of table 6.4.

Months \ Days	1	2	3	4	5	6	7
	106.6	**119.4**	**165.6**	**222.4**	**283.1**	**324.0**	357.7
6	**153.3**	**179.2**	**214.2**	**279.8**	314.7	**356.0**	**391.3**
	212.2	**211.2**	**254.2**	310.5	**355.4**	397.6	**430.8**
	106.1	120.9	188.8	238.8	**281.0**	325.8	**356.8**
7	**179.7**	**178.9**	223.6	**282.3**	**313.5**	**347.0**	400.1
	248.5	**220.4**	255.9	305.8	357.2	**379.0**	451.9
	106.3	135.6	193.4	**195.3**	270.0	**318.5**	**356.7**
8	**164.6**	**177.6**	227.7	**274.9**	**308.9**	**365.9**	**396.9**
	257.3	**219.1**	**262.9**	**316.9**	**339.3**	418.8	**445.1**
	92.6	113.8	164.5	**171.3**	**260.8**	**288.1**	**324.0**
9	**141.5**	**159.9**	**211.9**	**239.7**	**293.6**	**320.7**	**362.2**
	227.9	**191.1**	**236.8**	**273.5**	327.8	367.0	422.7
	90.9	**100.5**	106.6	**134.0**	**182.4**	**271.8**	**293.2**
10	**139.4**	**169.2**	169.4	215.8	**248.7**	**288.8**	**310.9**
	205.3	**215.6**	**206.9**	249.0	285.5	321.1	362.1
	73.8	**81.8**	99.8	130.1	**146.8**	**191.2**	**219.0**
11	**95.2**	**113.8**	**120.5**	**148.1**	**175.0**	**203.4**	232.8
	123.5	**144.6**	**139.4**	**166.7**	**187.4**	**216.2**	250.2
	74.9	**72.4**	**84.2**	110.7	147.3	175.7	**200.3**
12	**97.3**	**99.0**	**114.0**	136.1	**164.5**	**185.6**	**207.0**
	122.9	**118.2**	**131.0**	**152.3**	174.8	**195.8**	**216.6**

Table 6.4: Values of testing MAE, in *MW*, for regressor x_i^*. For each combination of days and months, the three values are, from top to bottom, the minimum, average and maximum obtained over the 11 modelling trials. Bold values indicate a smaller value when compared to the corresponding of table 6.3.

Months \ Days	1	2	3	4	5	6	7
	60.4	56.5	71.9	77.9	84.2	94.7	108.8
6	70.5	68.9	81.0	97.9	107.1	119.6	131.5
	65.5	72.1	85.7	89.2	106.3	113.4	127.4
	58.8	59.5	67.0	75.5	82.0	99.7	100.3
7	67.4	66.1	77.9	93.7	102.5	114.4	125.8
	65.5	69.2	80.4	85.1	103.7	114.6	119.8
	54.1	59.7	63.9	72.8	87.6	88.6	103.5
8	64.6	64.1	75.7	89.8	98.2	109.6	120.5
	61.4	69.8	74.3	89.2	102.6	106.6	122.9
	56.7	59.8	64.2	75.7	77.7	88.1	95.4
9	65.3	63.6	74.5	88.5	97.0	108.1	119.6
	72.0	64.1	74.1	84.7	92.2	111.0	118.0
	52.5	60.3	63.9	68.7	76.0	84.4	91.4
10	65.1	62.3	70.6	82.0	91.7	103.9	115.1
	59.8	63.4	75.6	81.2	87.7	99.0	118.1
	59.0	55.0	57.9	69.9	76.4	83.1	90.9
11	66.1	62.2	68.3	78.6	88.7	101.1	112.0
	67.3	64.2	72.4	87.1	88.5	98.1	104.1
	55.8	59.9	66.4	68.9	72.7	80.4	93.2
12	64.0	61.4	67.3	76.1	84.8	97.2	108.4
	64.0	65.6	74.0	75.4	84.0	94.7	107.2

Table 6.5: Values of training MAE, in MW, for regressor x_i^*. For each combination of days and months, the three values are, from top to bottom, the minimum, the value obtained by the model initialized by $\sigma = h$ and $\gamma = 1.0$ and average obtained for the corresponding days/months combinations over all the 11 modelling trials.

Months \ Days	1	2	3	4	5	6	7
	106.6	119.4	165.6	222.4	283.1	324.0	357.7
6	107.8	131.7	192.6	259.0	305.1	343.6	382.0
	153.3	179.2	214.2	279.8	314.7	356.0	391.3
	106.1	120.9	188.8	238.8	281.0	325.8	356.8
7	106.1	131.3	191.4	257.9	305.7	344.8	382.7
	179.7	178.9	223.6	282.3	313.5	347.0	400.1
	106.3	135.6	193.4	195.3	270.0	318.5	356.7
8	106.3	135.6	193.4	257.3	308.1	350.7	391.3
	164.6	177.6	227.7	274.9	308.9	365.9	396.9
	92.6	113.8	164.5	171.3	260.8	288.1	324.0
9	97.5	113.8	164.5	226.0	274.2	312.4	348.4
	141.5	159.9	211.9	239.7	293.6	320.7	362.2
	90.9	100.5	106.6	134.0	182.4	271.8	293.2
10	90.9	100.5	134.5	185.0	235.6	273.6	306.1
	139.4	169.2	169.4	215.8	248.7	288.8	310.9
	73.8	81.8	99.8	130.1	146.8	191.2	219.0
11	81.1	81.8	99.8	131.1	165.0	195.0	221.6
	95.2	113.8	120.5	148.1	175.0	203.4	232.8
	74.9	72.4	84.2	110.7	147.3	175.7	200.3
12	77.2	79.3	93.7	117.7	147.3	175.7	200.3
	97.3	99.0	114.0	136.1	164.5	185.6	207.0

Table 6.6: Values of testing MAE, in MW, for regressor x_i^*. For each combination of days and months, the three values are, from top to bottom, the minimum, the value obtained by the model initialized by $\sigma = h$ and $\gamma = 1.0$ and average obtained for the corresponding days/months combinations over all the 11 modelling trials.

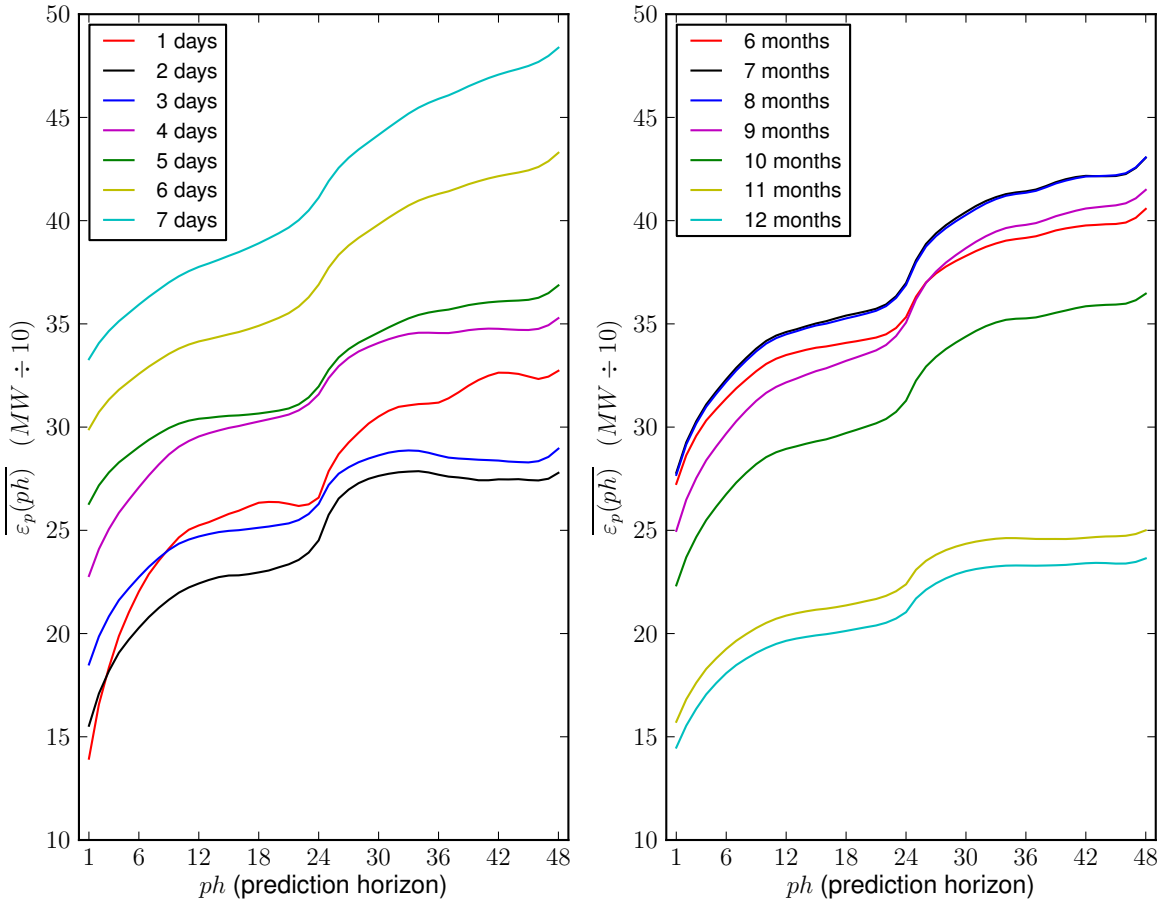

Figure 6.1: Average curves of evolution of ε_p, in MW, for the test set over the prediction horizon. Left: The average is computed for all models with the same value of days. Right: The average is computed for all models with the same value of months.

the average is computed for all models with the same value of days and on the right, the average is computed for all models with the same value of months.

As we can see on this figure, and concluded before, the combinations with 11 and 12 months, and with 1, 2, and 3 days on the regressor dimension are better than the others.

Figure 6.2 shows ε_p^* values for all modelling trials of each combination $\{N, d\}$. Analysing figure 6.2, we can clearly confirm that combinations with 1, 2 and 3 days are better than the others for long-term simulation, which is the final application, the regressor with 3 days being the best.

Figure 6.3 shows the evolution of the objective function (21) over the iterations of the gradient descent method. The evolution is analysed based on averages over d and N as in figure 6.1. On left, the average is computed according to the days and on right according to the months. As we can see, in some cases, it would be possible to make more than 15 iterations to achieve better convergence.

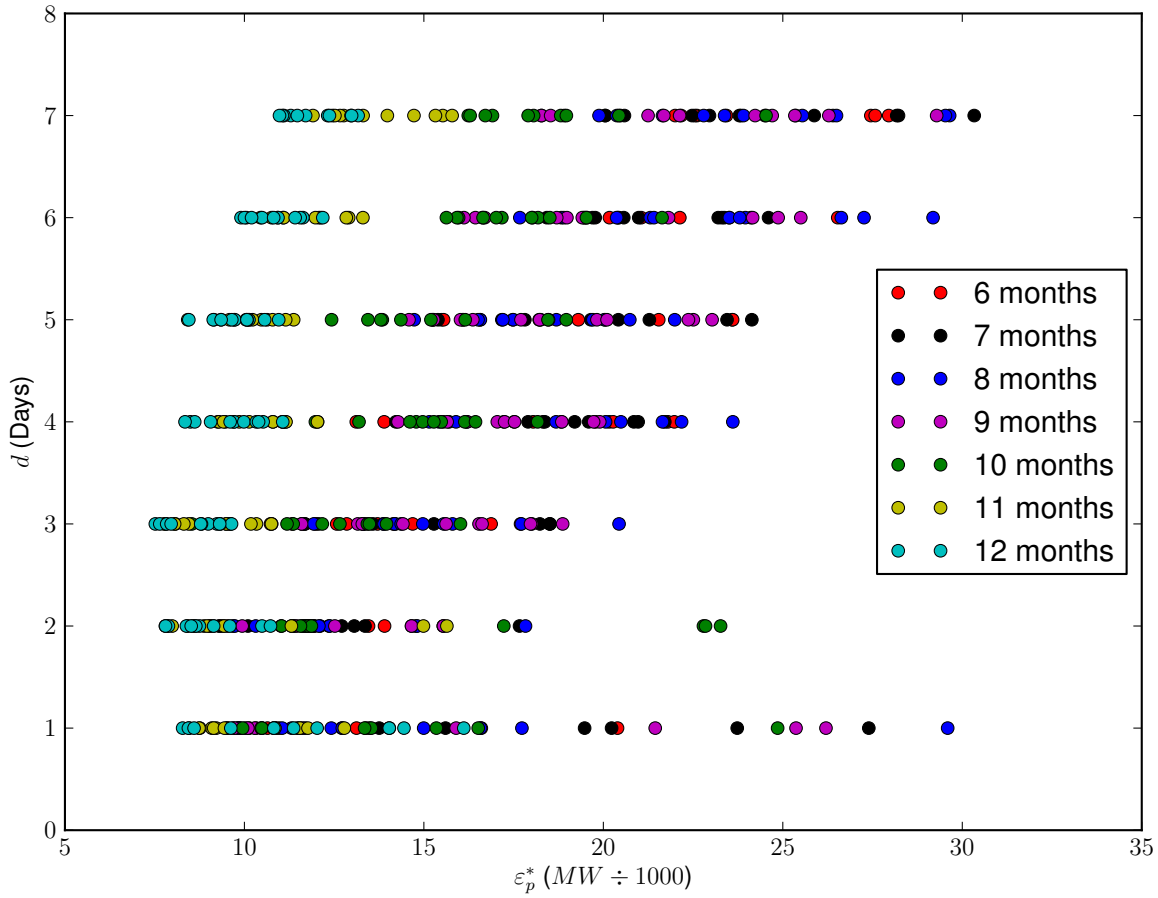

Figure 6.2: ε_p^* values for all modelling trials of each combination $\{N, d\}$.

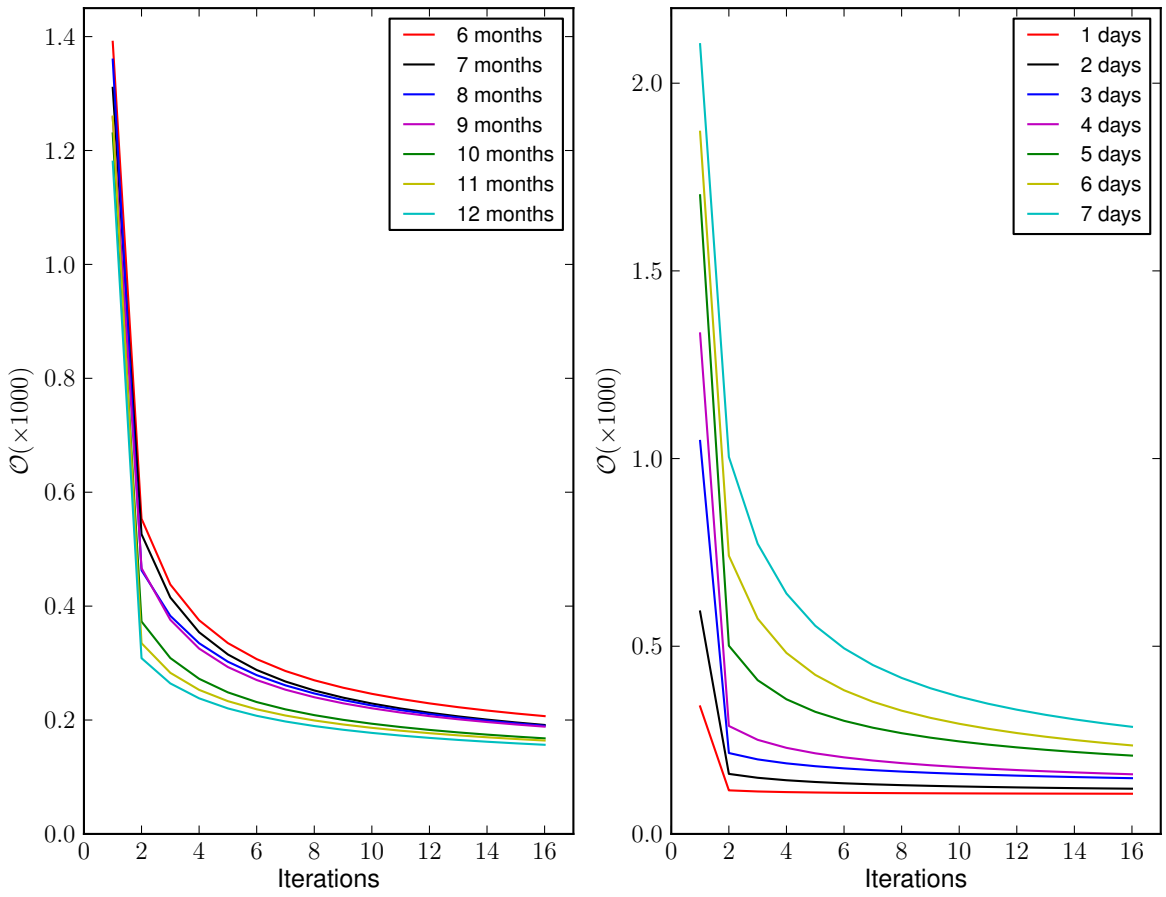

Figure 6.3: Evolution of the objective function over the iterations of the gradient descent. Left: The average is computed according to the days. Right: The average is computed according to the months.

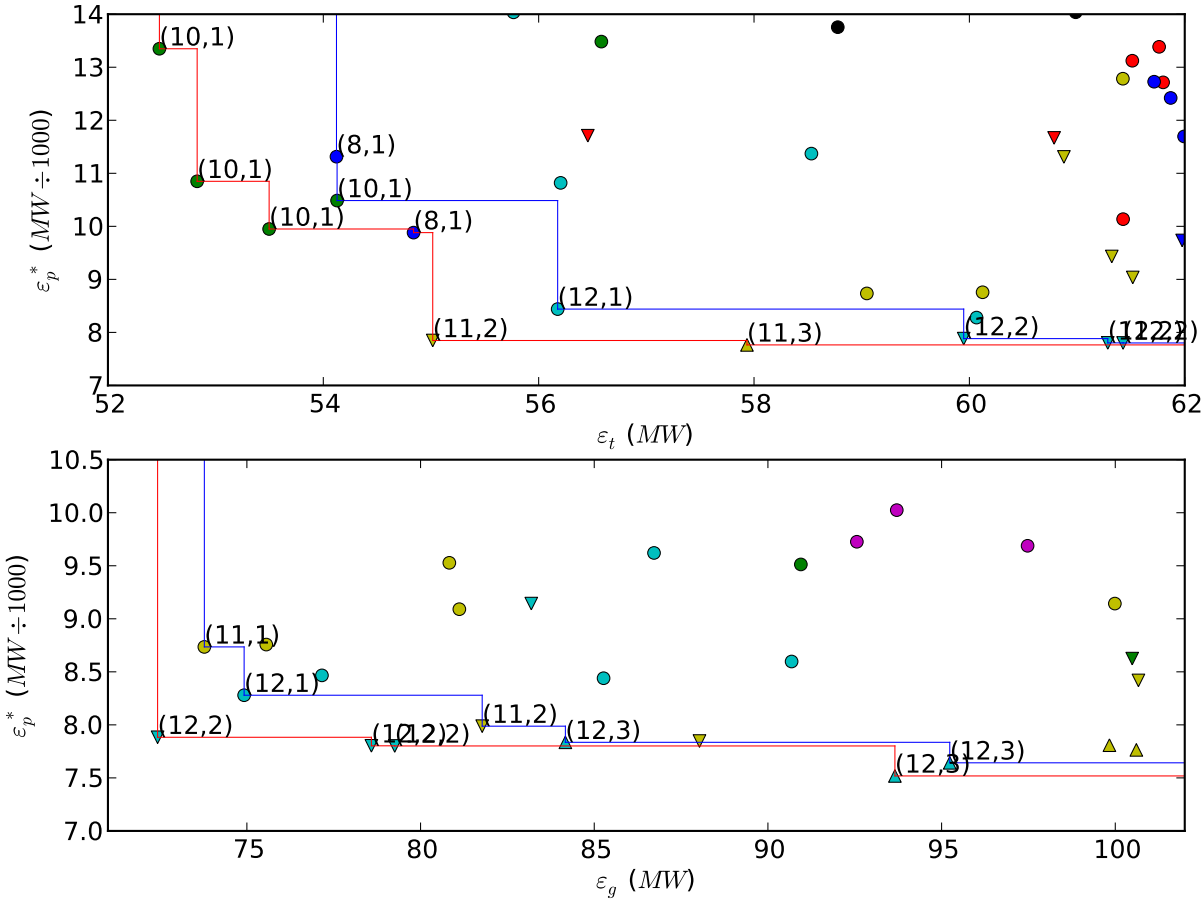

Figure 6.4: Training and testing average error versus ε_p^*. Top: Pareto fronts of ε_t versus ε_p^*. Bottom: Pareto fronts of ε_g versus ε_p^*. The inner line represents the Pareto fronts obtained by removing the outer line.

6.1.1 Choosing N and d

Figure 6.4 presents a partial view of the results achieved on all modelling trials of all $\{N,d\}$ combinations, organized in two plots where the *MAE* of the training set and test set is plotted against the long-term prediction performance measure ε_p^*. The two lines shown in each plot constitute Pareto fronts that were used to select relevant $\{N,d\}$ combinations. The inner lines are the Pareto fronts that would be achieved if the points on the outer lines were removed.

By looking at these lines a number of combinations achieves good results, but no definite conclusion can be made. Therefore the combinations having points on the lines were used for the experiment $E3$ in order to obtain more results and restrict the selection.

The inner Pareto front was determined to avoid leaving out important combinations, that can be also promising but would be discarded by the outer Pareto front.

Table 6.7 presents all the combinations obtained from the Pareto fronts represented on figure 6.4. The combinations obtained show, as said before, that combinations with greater N and lower d,

Months	Days
8	1
10	1
11	1
11	2
11	3
12	1
12	2
12	3

Table 6.7: Combinations of $\{N,d\}$ from the Pareto fronts from figure 6.4.

achieve better results.

Figure 6.5 shows the results from experiment $E3$ with a presentation similar to 6.4. On top, we have the representation of the Pareto front for the training set and on bottom for the test set. In this case, only the $\{N,d\}$ combinations shown in table 6.7 were considered. By using the Pareto fronts shown, it was concluded that only the combinations $\{(12,2),(12,3)\}$ were common to all the fronts which means they achieved a good balance between training set and test set. As combination $\{12,3\}$ achieved a smaller ε_p^* result, this combination was chosen for the continuation of the work in this dissertation. Figure 6.6, similar to figure 6.1 but for the results of experiment $E3$, also confirms that on average the combination $\{12,3\}$ is preferable to $\{12,2\}$.

6.2 Pruning the models

Figures 6.6 and 6.7 present results from pruning methods, for each pruning percentage, given from experiments $E4$ and $E5$, based on $I(Pb(x_i))$ and $|\alpha|$ respectively. On both figures, on the first plot the percentages are plotted against ε_t, on the second plot the percentages are plotted against ε_g, and on the last two plots the percentages are plotted against ε_p^*. On the third plot, ε_p^* is computed using all the test set and on the last plot the data includes only the first three months of the test set. This way the modelling results are analysed both from a localized and global perspective, regarding time. On all plots, the larger circle marker represents the average obtained over the 11 modelling trials, the smaller circle markers represent all the modelling trials with random initialization of (σ,γ) and the star marker represents the modelling trial with $(\sigma,\gamma) = (h,1.0)$ for each percentage.

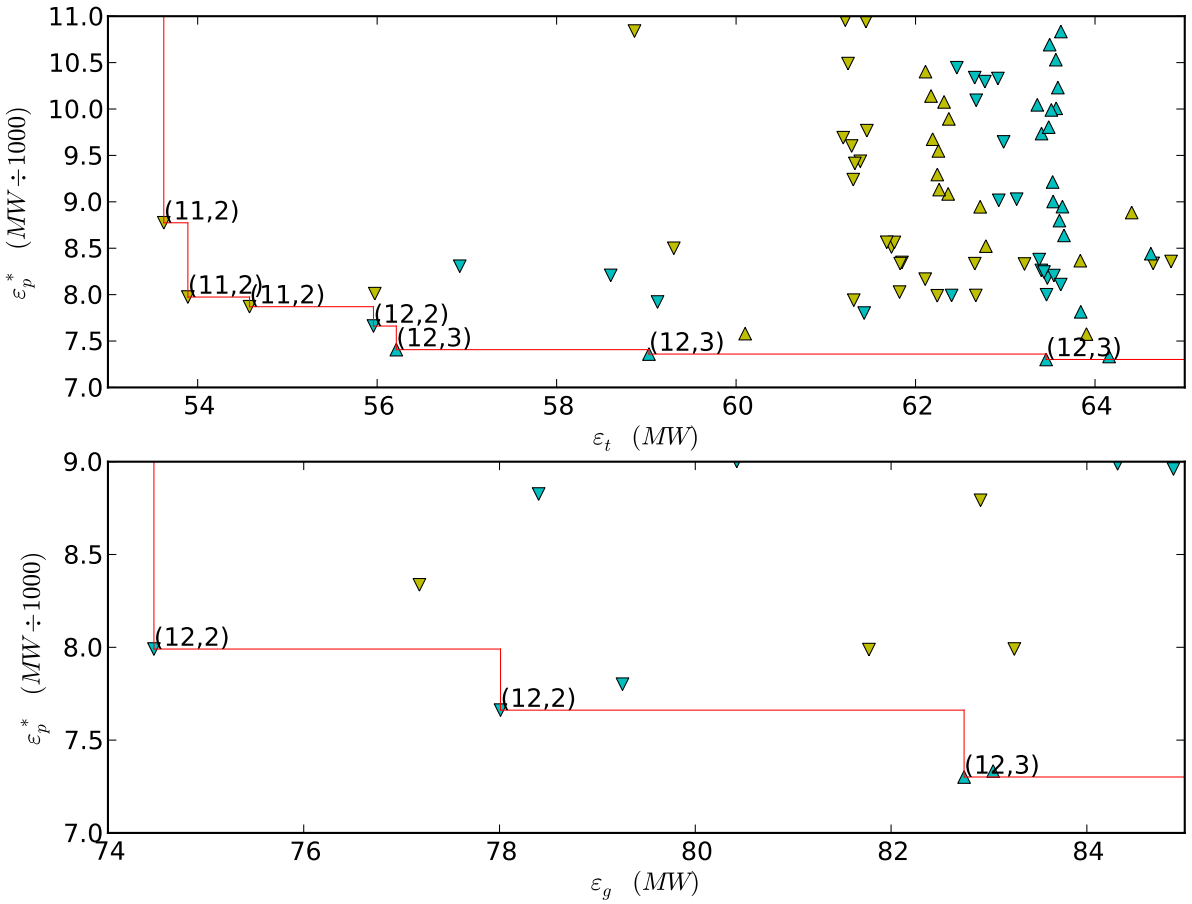

Figure 6.5: Training and testing average error versus ε_p^*. Top: Pareto fronts of ε_t versus ε_p^*. Bottom: Pareto fronts of ε_g versus ε_p^*.

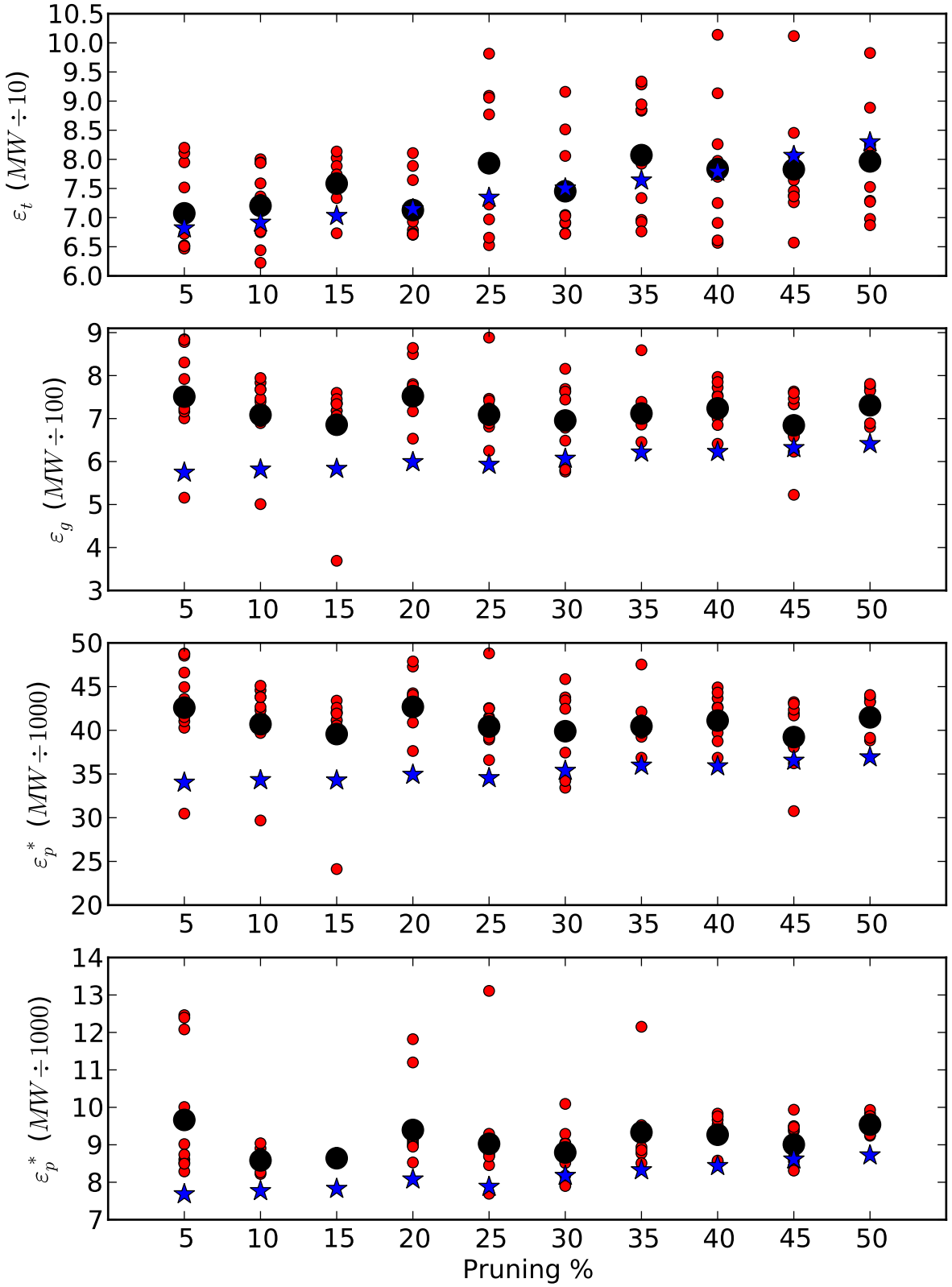

Figure 6.6: Pruning method based on $I(Pb(x_i))$, as explained on 4.4. On all plots, the larger circle marker represents the average obtained over the 11 modelling trials, the smaller circle markers represent all the modelling trials with random initialization of (σ, γ), and the star marker represents the modelling trial with $(\sigma, \gamma) = (h, 1.0)$ for each pruning percentage.

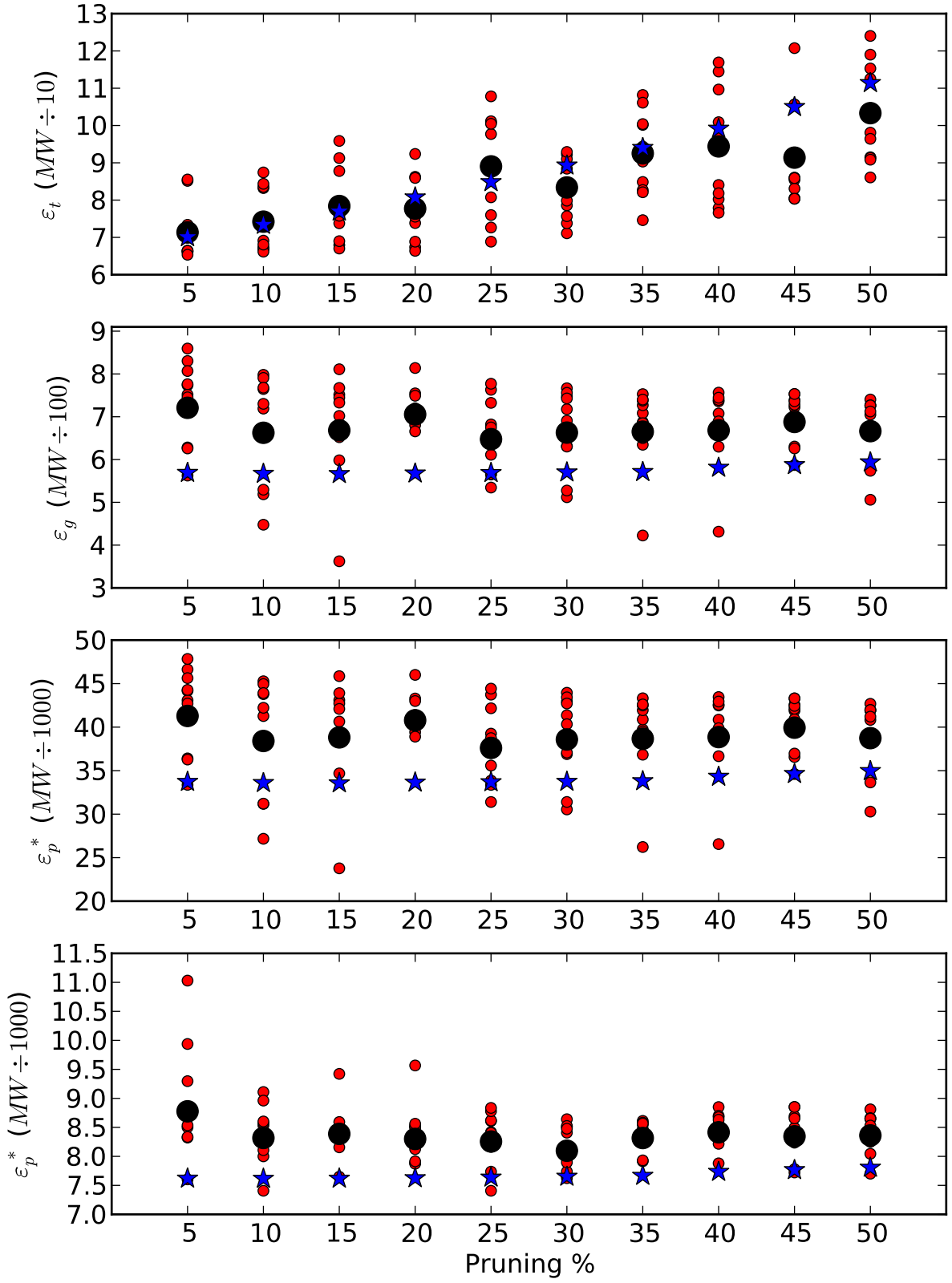

Figure 6.7: Pruning method based on $|\alpha|$, as explained on 3.1. On all plots, the larger circle marker represents the average obtained over the 11 modelling trials, the smaller circle markers represents all the modelling trials with random initialization of (σ, γ), and the star marker represents the modelling trial with $(\sigma, \gamma) = (h, 1.0)$ for each pruning percentage.

6.2.1 Comparing pruning methods

It was verified that in both pruning methods, the models loose the predictive performance over the years and more sharply after the first year. It was also noted that on the first year, there is a period where the results are worse. This period repeats every year denoting a time-of-year where results are systematically worse.

Comparing figures 6.6 and 6.7, we can conclude that for pruning using the $|\alpha|$ method the results are slightly better than the ones when pruning using $I(Pb(x_i))$ is applied. By looking specially to the last plot of both figures, we can conclude that ε_p^* results of the $|\alpha|$ method are smaller than ε_p^* results of the $I(Pb(x_i))$, mainly when we analyse the average ε_p^* and $(h, 1.0)$ on each percentage. Because of this we will continue the work in this dissertation using only the $|\alpha|$ method.

6.2.2 Choosing the best pruning percentages

On the first twelve months of simulation, the behaviour of the $|\alpha|$ method is very uniform among the models with regard to the predictive performance. From the set of results of experiment E4, two models were selected, one from the local time perspective, the other from the global simulation. The choice may be identified in figure 6.7, on the two bottom plots:

1. When we analyse ε_p^* using data from all the test set (second plot from bottom), a modelling trial with 25% of pruning presents the smallest ε_p^*.

2. When we analyse ε_p^* using data that includes only the first three months of the test set (bottom plot), a modelling trial with 15% of pruning presents the smallest ε_p^*.

This way the modelling results are analysed both from a localized and global perspective, regarding time.

The models obtained in these two modelling trials will be used on experiments $E6$ and $E7$.

6.3 On-line adaptation of the models

Table 6.8 shows ε_p^* results for the selected LS-SVM models from experiment $E4$, as well as for the RBF ANN model currently in use at REN. For each LS-SVM model two experiments were executed, $E6$ and $E7$ as described in 5.4. In the experiments, all models, including RBF ANNs were tested on parameter adjustment periodicities $p = 7$, $p = 14$ and $p = 30$ days. The acronym

	25%		15%		
p	*prev*	$(h, 1.0)$	*prev*	$(h, 1.0)$	RBF
7	9234.9	9352.5	9730.0	9402.3	11628.7
14	9455.2	9560.3	9858.2	9493.8	11783.9
30	9650.0	9766.7	9965.3	9579.8	11872.5
never	33853.4		23771.4		26139.5

Table 6.8: Results for ε_p^*, in MW, for each (σ, γ) combination tested, with three different parameter adjustment periodicities (p), for the best LS-SVM models and RBF ANN model. The acronym *prev.* represents the experiment $E7$ were (σ, γ) are initialized with values from the previous parameter adjustment, and $(h, 1.0)$ represents experiment E6 where $(\sigma, \gamma) = (h, 1.0)$. The row denoted by *never*, presents ε_p^* values considering the models without on-line adaptation.

prev. denotes results from models initialized with values from the previous parameter adjustment and the row denoted by *never* shows the results of the models without on-line adaptation. All the results are now obtained considering the complete test set, up to 2011.

Figures 6.8 and 6.9 show results obtained by the LS-SVM models and the RBF ANN model without on-line adaptation. On figure 6.8 each line shows the evolution of ε_p^* computed on a window of the past three months using a step of one week. On figure 6.9, each curve represents the evolution of ε_p over the prediction horizon.

Figures 6.10 and 6.11 show similar results but considering the models with on-line adaptation. On figure 6.10 each line represents the ε_p^* value for each model considering, varying the parameter (σ, γ) initialization. On figure 6.11, each curve represent the evolution of the ε_p for the models and variations just mentioned.

In figures 6.10 and 6.11 the periodicities p=7, 14 and 30, are denoted by solid, dashed and dot-dashed line types. The colours cyan and magenta are for the 25% pruning percentage simulation, red and blue for the 15%, and black for the RBF ANN. Magenta and blue denote simulations where the LS-SVMs were initialized using $(h, 1.0)$.

Figure 6.12 shows the evolution of the MAPE over the prediction horizon for a LS-SVM model trial obtained with a pruning percentage of 25% and the best RBF ANN model, all with the parameter adjustment periodicity $p = 7$. The MAPE values obtained with the model in use actually at REN, presented in 2.2, are similar to those presented in 6.12, although the data set has a much smaller size than the one used in this work. Also, the model at REN had been trained with data just before the period in analysis. The results from Gontar, Sideratos, and Hatziargyriou [14] from both architectures, also presented on 2.2, have worse MAPE values for 24 and 48 hours prediction horizon.

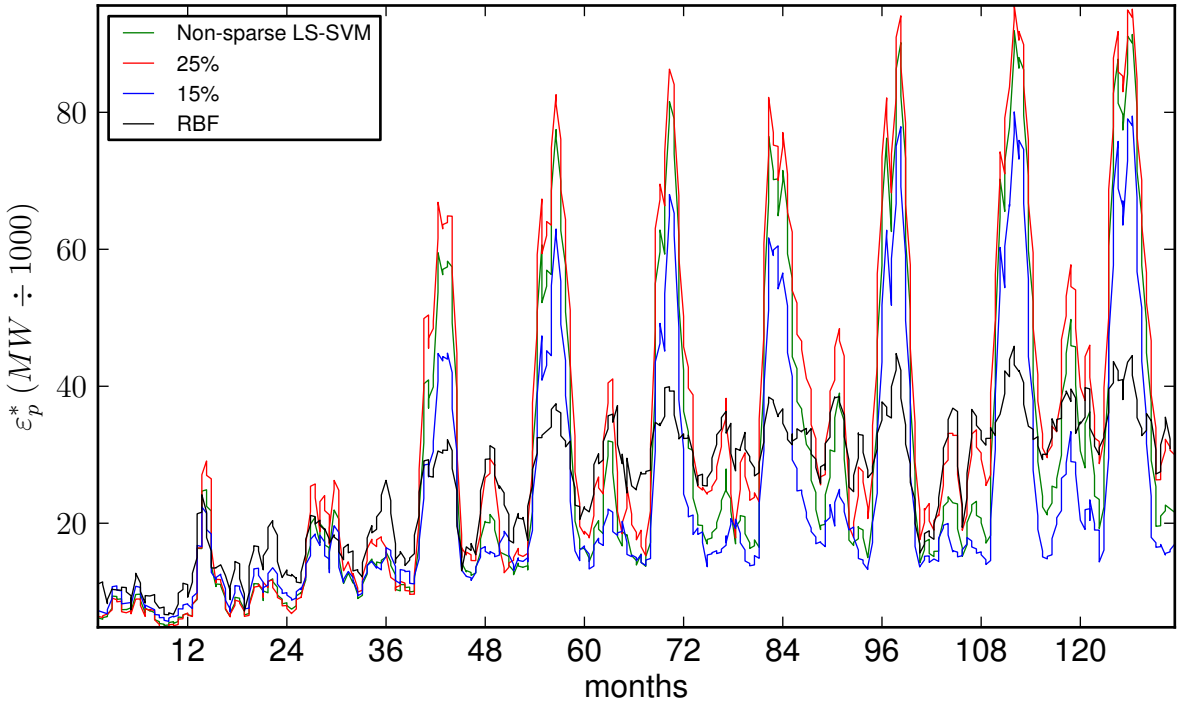

Figure 6.8: Comparison between ε_p^* values of the best LS-SVM models and RBF ANNs over all the test set. These results were obtained without the on-line adaptation of the models.

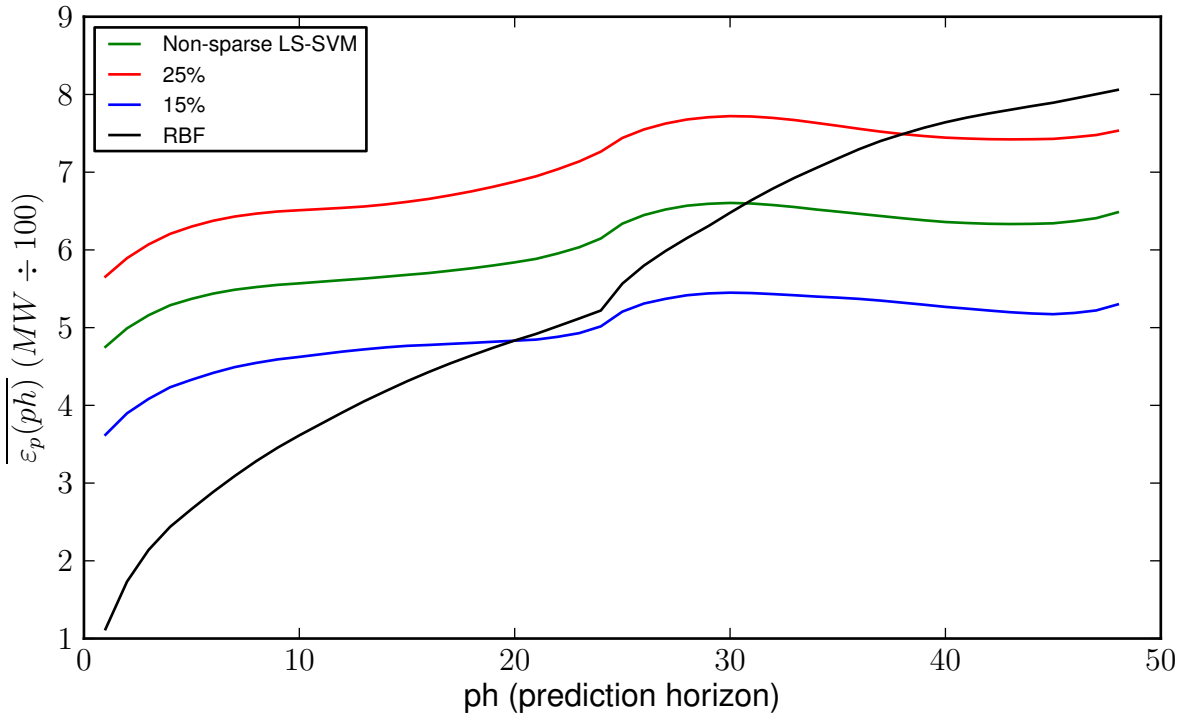

Figure 6.9: Average curves of evolution of ε_p, in MW, over the prediction horizon for all models. The curves were obtained without the on-line adaptation of the models.

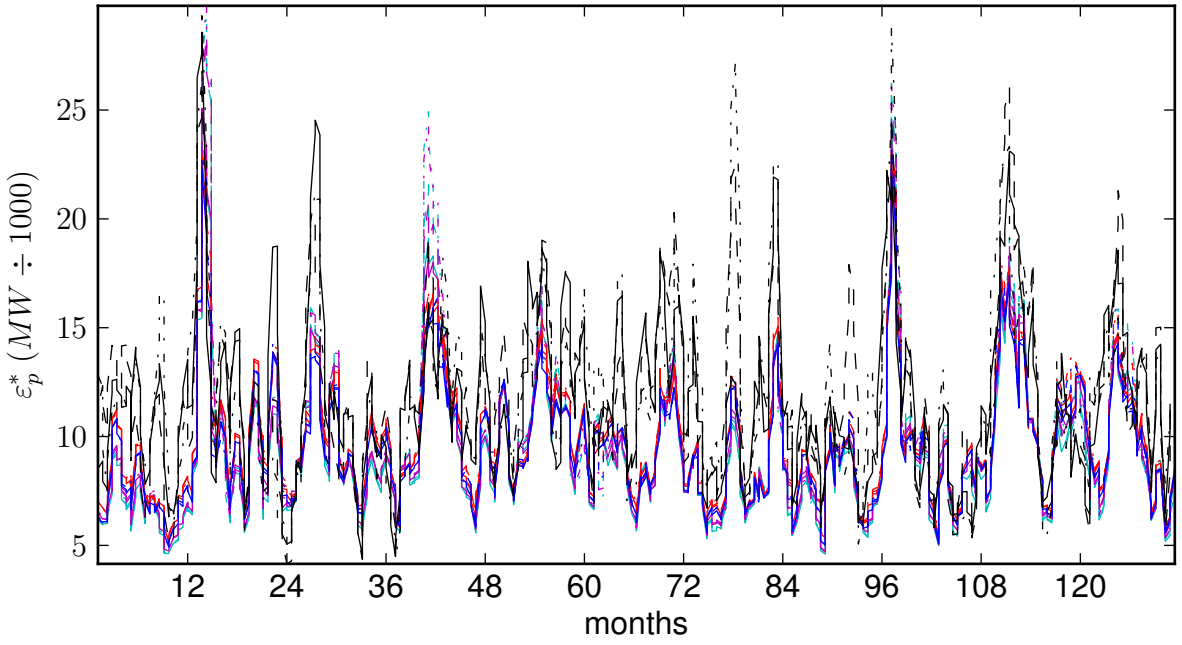

Figure 6.10: Comparison between ε_p^* values of the best LS-SVM models and RBF ANNs over all the test set. These results were obtained with the on-line adaptation of the models. Each model was tested with different parameter adjustment periodicities (p). For the LS-SVM models, each periodicity of each model was tested with different initial values of (σ, γ), as explained on 5.4. Please see the text for details.

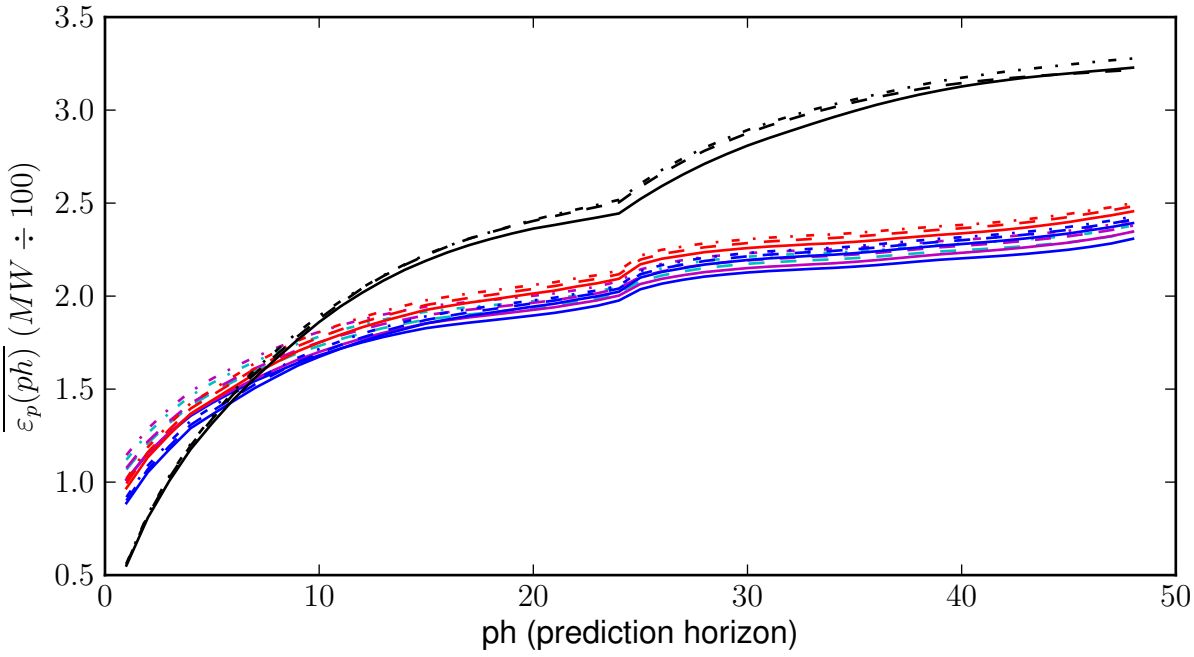

Figure 6.11: Average curves of evolution of ε_p, in MW, for all models over the prediction horizon. The curves were obtained with the on-line adaptation of the models. Each model was tested with different parameter adjustment periodicities (p). For the LS-SVM models, each periodicity of each model was tested with different initial values of (σ, γ), as explained on 5.4. Please see the text for details.

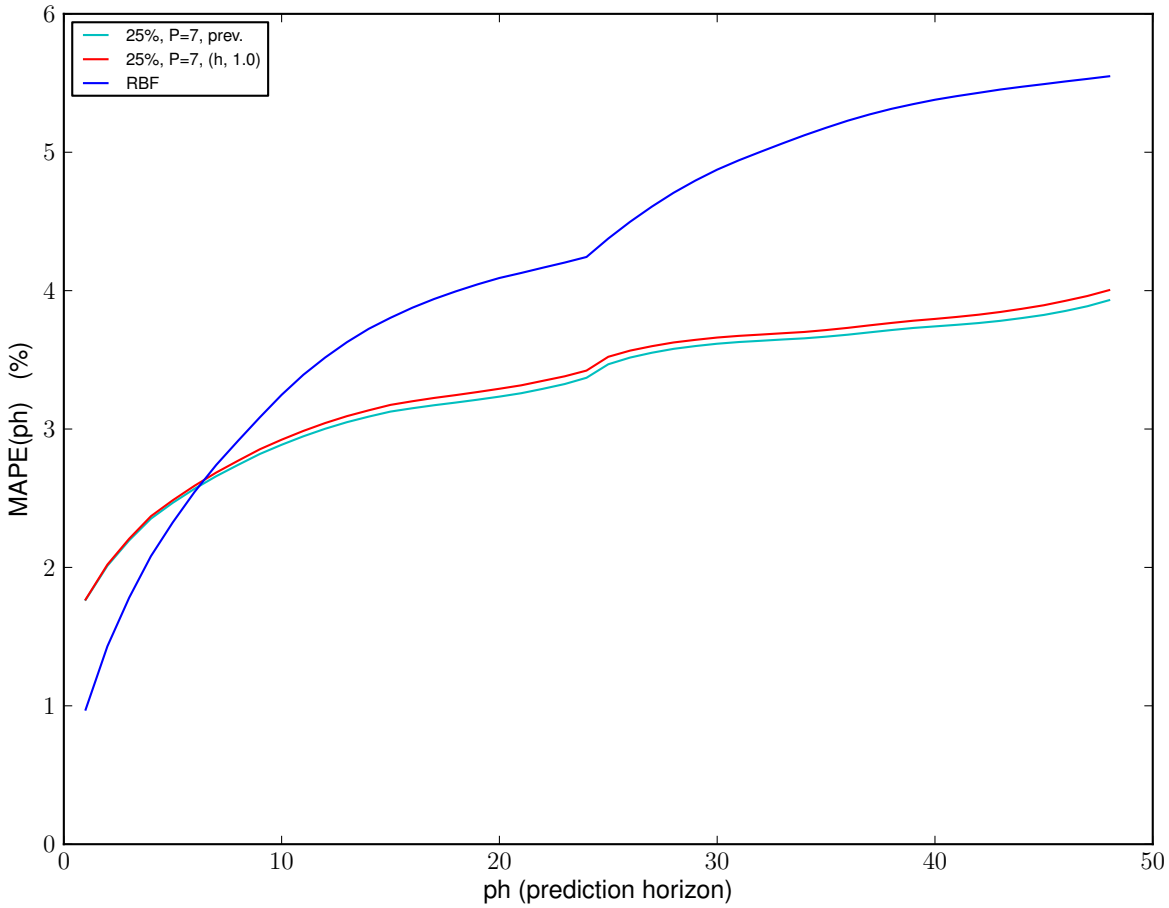

Figure 6.12: MAPE obtained using the test set for the LS-SVM model obtained using 25% pruning percentage with parameter adjustment periodicity $p = 7$ days, and the best RBF ANN model with $p = 7$, over the prediction horizon.

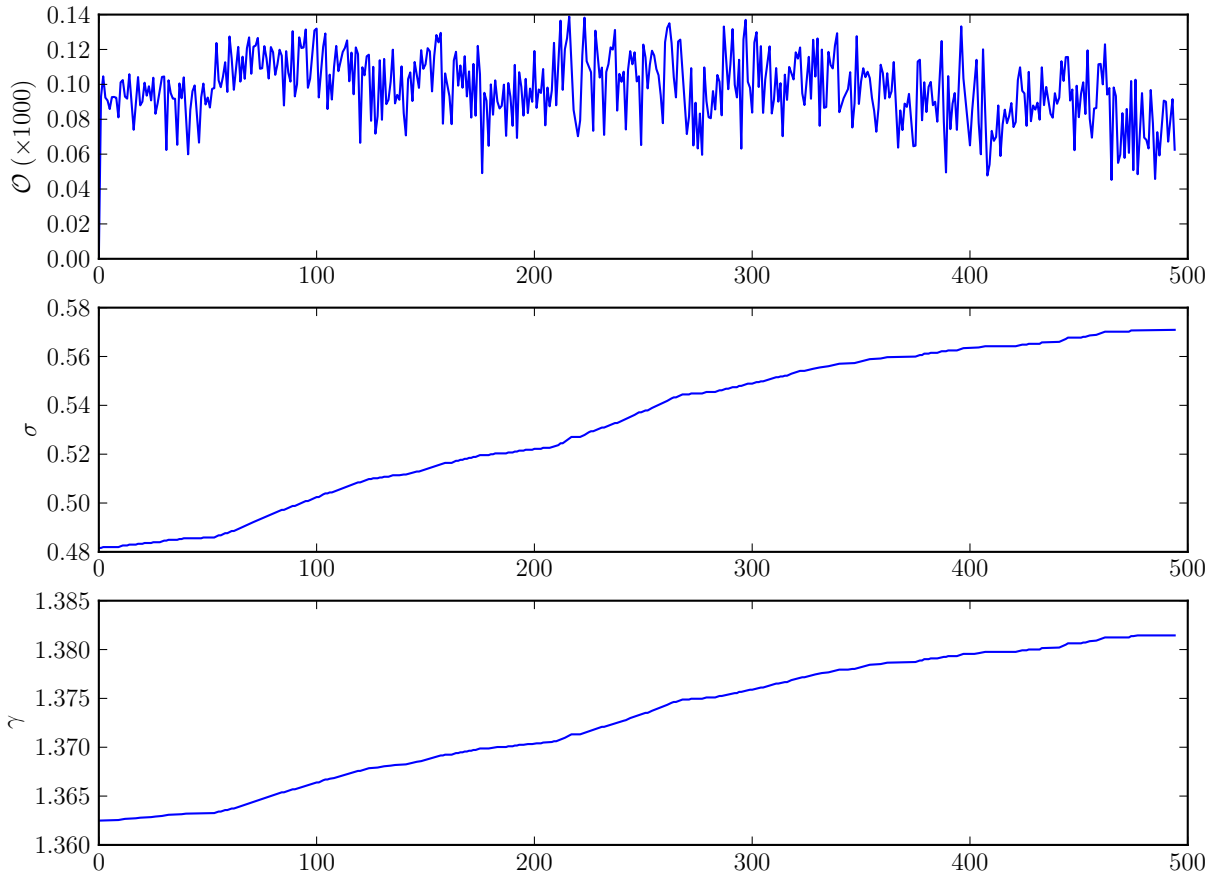

Figure 6.13: Results from model obtained from 25% pruning percentage with the parameter adjustment periodicity $p = 7$ and $(\sigma, \gamma) = (prev, prev)$. Top: Evolution of the objective function 21. Middle: Evolution of the σ. Bottom: Evolution of the γ. All over the test set in the parameters adjustment.

Figure 6.13 shows results from a model trial obtained with 25% pruning percentage, with the parameter adjustment periodicity $p = 7$, and (σ, γ) initialized with corresponding values from the previous parameter adjustment. From top to bottom, the figures shows the evolution of the objective function (21), the evolution of σ and γ, over all the adjustments done during the simulation. The γ and σ values increase while the objective function varies near the 0.0001 value, which is the precision specified for algorithm 1 $(\theta = 0.0001)$.

6.4 Comparing on-line adapted LS-SVMs and the existing RBF model

By comparing the results obtained without on-line adaptation, presented on figures 6.8 and 6.9, with the results with on-line adaptation, presented on figures 6.10 and 6.11, we can see that the results obtained with on-line adaptation achieve a significant improvement. Because of this, it may be concluded that on-line adaptation is required for improved results.

By looking at figures 6.10 and 6.11, and to table 6.8, it may also be concluded that the lower the parameter adjustment periodicity is, the better the results for all models are.

For the on-line adaptation of LS-SVM models, the results from all the parameter adjustment periodicities with the distinct (σ, γ) initializations, are very similar. The model with 15% pruning percentage with $(\sigma, \gamma) = (h, 1.0)$ and the model with 25% pruning percentage with $(\sigma, \gamma) = (prev, prev)$ are comparable and in general have better results than the remaining cases.

By analysing figures 6.9 and 6.11, it may also be concluded that RBF ANNs models present significantly better results than LS-SVM at the beginning of the prediction horizons.

7 Conclusions and future work

For this master thesis dissertation, the LS-SVM algorithm was tested for multi-step ELD forecasting with the goal of overcoming the difficulties found in the use of RBF ANNs. To provide sparseness to the LS-SVM model, pruning methods, including an information-theoretic criterion based on multivariate kernel density estimation, were tested. All algorithms were implemented using the Python/C API programming language, including a direct translation of the MOGA considering a reference Matlab implementation.

The use of a second input variable, which is an encoded number with a different value for the day of the week and holidays, and larger regression windows with smaller regressor size were beneficial to have better forecasting results. It was shown that sparseness is required for better LS-SVM models. The pruning method based on $|\alpha|$ values presented better results when compared to the information-theoretic based method using SUS as the fitness indicator. The latter method needs more work, specially regarding the conditions on which it may benefit the models predictive performance.

For off-line models, on average, LS-SVMs present better results than RBF ANNs except when the prediction horizon is less than about 20 hours. For larger prediction horizons LS-SVMs achieved much better results. The same happens when we use on-line adapted models but, in this case RBF ANNs are better only with prediction horizons are smaller than 5 hours. This comes to confirm the main objective of this dissertation of obtaining better results than with RBF ANNs, which is a significant contribution for the multi-step ELD forecasting subject. With this, the goals of REN, or other companies who will choose to use this methodology, will be achieved more efficiently and accurately because LS-SVM models are less sensitive to adaptation over time and more accurate. The $(h, 1.0)$ values are good initial values for (σ, γ), avoiding the necessity of using random initializations for these parameters. The use of smaller parameter adjustment periodicity also improves the results.

For future work, it can be interesting to test if LS-SVM models can be improved either by increasing the size of the regressor window, by better initializations of (σ, γ), or by a different fitness indicator to choose points to prune based on the information-theoretic measure. The increasing of the size of the regression window may also improve the results achieved by the information-theoretic pruning method, because the larger the training set is, the more significant the selection of points is expected to be.

References

[1] S.M. Al-Alawi and S.M. Islam. Principles of electricity demand forecasting. i. methodologies. *Power Engineering Journal*, 10(3):139–143, 1996.

[2] Christopher M. Bishop. Neural networks: A pattern recognition perspective. Technical report, 1996.

[3] Q.S. Chen, X. Zhang, S.H. Xiong, and X.W. Chen. Short-term power load forecasting with least squares support vector machines and wavelet transform. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 3, pages 1425–1429. IEEE, 2008.

[4] Pedro M. Ferreira and António E. Ruano. Exploiting the separability of linear and non-linear parameters in radial basis function neural networks. In *IEEE Symposium 2000: Adaptive Systems for Signal Processing, Communications, and Control*, pages 321–326, Canada, 2000. doi: 10.1109/ASSPCC.2000.882493.

[5] Pedro M. Ferreira and António E. Ruano. Evolutionary multiobjective neural network models identification: Evolving task-optimised models. In António E. Ruano and Annamária R. Várkonyi-Kóczy, editors, *New Advances in Intelligent Signal Processing*, volume 372 of *Studies in Computational Intelligence*, pages 21–53. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-11738-1. doi: 10.1007/978-3-642-11739-8_2. URL `http://dx.doi.org/10.1007/978-3-642-11739-8_2`.

[6] P.M. Ferreira and A.E. Ruano. Online sliding-window methods for process model adaptation. *Instrumentation and Measurement, IEEE Transactions on*, 58(9):3012–3020, sept. 2009. ISSN 0018-9456. doi: 10.1109/TIM.2009.2016818.

[7] P.M. Ferreira, E.A. Faria, and A.E. Ruano. Neural network models in greenhouse air temperature prediction. *Neurocomputing*, 43(1–4):51–75, 2002. ISSN 0925-2312. doi: 10.1016/S0925-2312(01)00620-8. URL `http://www.sciencedirect.com/science/article/pii/S0925231201006208`. Selected engineering applications of neural networks.

[8] P.M. Ferreira, A.E. Ruano, R. Pestana, and L.T. Kóczy. Evolving rbf predictive models to

forecast the portuguese electricity consumption. In *Intelligent Control Systems and Signal Processing*, volume 2, pages 414–419, 2009.

[9] P.M. Ferreira, A.E. Ruano, and R. Pestana. Improving the identification of rbf predictive models to forecast the portuguese electricity consumption. In *Control Methodologies and Technology for Energy Efficiency*, volume 1, pages 208–213, 2010.

[10] P.M. Ferreira, A.E. Ruano, and R. Pestana. Towards online operation of a rbf neural network model to forecast the portuguese electricity consumption. In *Intelligent Signal Processing (WISP), 2011 IEEE 7th International Symposium on*, pages 1–7. IEEE, 2011.

[11] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *ICGA*, pages 416–423, 1993.

[12] L. Ghods and M. Kalantar. Long-term peak demand forecasting by using radial basis function neural networks. *IRANIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING*, 6(3):175–182, 2010.

[13] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0801854148. URL `http://portal.acm.org/citation.cfm?id=248979`.

[14] Z. Gontar, G. Sideratos, and N. Hatziargyriou. Short-term load forecasting using radial basis function networks. *Methods and Applications of Artificial Intelligence*, pages 432–438, 2004.

[15] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.

[16] R.J. Hyndman and E. Sztendur. Modelling long-term peak half-hourly electricity demand for south australia. *Report for Electricity Supply Industry Planning Council (SA)*, 2007.

[17] Antonia J. Jones. New tools in non-linear modelling and prediction. *Comput. Manag. Sci*, pages 109–149, 2004.

[18] C. Kahraman and M. Yavuz. *Production Engineering and Management Under Fuzziness*. Springer-Verlag New York Inc, 2010.

[19] K. Levenberg. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *The Quarterly of Applied Mathematics*, (2):164–168, 1944.

[20] M.A. Mamun and K. Nagasaka. Artificial neural networks applied to long-term electricity demand forecasting. In *Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on*, pages 204–209. IEEE, 2004.

[21] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.

[22] John Moody and Christian J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[23] David W. Scott and Stephan R. Sain. Multidimensional density estimation. In E.J. Wegman C.R. Rao and J.L. Solka, editors, *Data Mining and Data Visualization*, volume 24 of *Handbook of Statistics*, pages 229–261. Elsevier, 2005. doi: 10.1016/S0169-7161(04)24009-3. URL `http://www.sciencedirect.com/science/article/pii/S0169716104240093`.

[24] DavidW. Scott. Multivariate density estimation and visualization. In James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori, editors, *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, pages 549–569. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-21550-6. doi: 10.1007/978-3-642-21551-3_19. URL `http://dx.doi.org/10.1007/978-3-642-21551-3_19`.

[25] Alexander Smola, Chris Burges, Harris Drucker, Steve Golowich, Leo Van Hemmen, Klaus-Robert M̈ller, Bernhard Schölkopf, and Vladimir Vapnik. Regression estimation with support vector learning machines, 1996.

[26] M. Smola, AJ Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. 1997.

[27] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[28] J.A.K. Suykens, L. Lukas, and J. Vandewalle. Sparse approximation using least squares support vector machines. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 2, pages 757–760. IEEE, 2000.

[29] J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[30] T. Van Gestel, J.A.K. Suykens, D.E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle. Financial time series prediction using least squares support vector machines within the evidence framework. *Neural Networks, IEEE Transactions on*, 12(4):809–821, 2001.

[31] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

[32] Q. Wu, W. Liu, and Y. Yang. Time series online prediction algorithm based on least squares support vector machine. *Journal of Central South University of Technology*, 14 (3):442–446, 2007.

[33] Jun Zhao, Quanli Liu, W. Pedrycz, and Dexiang Li. Effective noise estimation-based online prediction for byproduct gas system in steel industry. *Industrial Informatics, IEEE Transactions on*, 8(4):953–963, nov. 2012. ISSN 1551-3203. doi: 10.1109/TII.2012.2205932.