M.Sc. Thesis
Master of Science in Sustainable Energy

DIPARTIMENTO DI
INGEGNERIA INDUSTRIALE

**DTU Electrical Engineering**
Department of Electrical Engineering

# Analytics of flexible electric consumption

## Forecasting the electrical load and the demand response availability

Fabio Moret (s142177)

Kongens Lyngby 2016

DTU

P. Alotto

Il co-relatore, Prof. Piergiorgio Alotto

**Ingegneria dell'Energia Elettrica**
**Dipartimento di Ingegneria Industriale**
**Università degli Studi di Padova**

Via Gradenigo, 6/a
35131 Padova, Italy
Tel (+39) 049 82 77 500
www.ienie.dii.unipd.it

**DTU Electrical Engineering**
**Department of Electrical Engineering**
**Center for Electric Power and Energy (CEE)**
**Technical University of Denmark**

Elektrovej
Building 325
2800 Kgs. Lyngby, Denmark
Tel (+45) 45 25 35 00
www.cee.elektro.dtu.dk

# Summary

The integration of renewable sources in the power system implies significant fluctuations of the generation. In view of larger shares of non-dispatchable power plants, new smart solutions have to be developed. The research community has recently focussed on the active participation of the consumption on real-time: the Demand Response. A real-time market that adjusts the electricity price each 5 minutes combined to automation systems on household users has been developed during EcoGrid EU project. Following up on the final recommendations of this project, this study aims to further develop the Short Term Load Forecasting model on the real database collected during EcoGrid EU. The statistical learning algorithms employed are all based on the Support Vector Machine, which is proved to be a very efficient machinery for large-scale datasets. All the methods implemented are found to be competitive with state-of-the-art load forecasting models. In particular, a reduced version of the Least Squares Support Vector Regression (IRR-LS-SVR) presents the best trade-off between accuracy and computational speed. Furthermore, the extraction of the non-linear dependency of the electrical load on temperature and price is carried out. The results demonstrate how the extraction of a general relationship, e.g. between consumption and price, is complicated by the presence of cross-dependencies and strong non-linearities in the process analysed. However, the forecasting routines provide a good level of accuracy and therefore they can also be used to predict the availability of the flexible assets at a specific time instant. Among all, the IRR-LS-SVR algorithm raises particular interest: its light and iterative structure makes it suitable for several further applications. For example, forecasting the dynamic of the responsive loads, and clustering all the users of the system to point out the representative subjects are two of the possible further employments of this method.

# Preface

This MSc thesis has been developed in fulfilment of the requirements for acquiring both a Master of Science in Sustainable Energy at the Technical University of Denmark (DTU) and a Master of Science in Electrical Energy Engineering at the University of Padua, in accordance with the double degree programme T.I.M.E. (Top Industrial Managers for Europe). Under the supervision of Pierre Pinson and the co-supervision of Piergiorgio Alotto, this work has been carried out between January and June 2016 at the Department of Electrical Engineering (DTU Elektro), in particular at the Center for Electric Power and Energy (CEE), at the Technical University of Denmark (DTU). The thesis was set to count 35 ECTS for the Master of Science in Sustainable Energy and 21 ECTS for the Master of Science in Electrical Energy Engineering.

Kongens Lyngby, July 26, 2016

Fabio Moret

Fabio Moret (s142177)

# Acknowledgements

This thesis could not have been done without the valuable guidance of my supervisor, Pierre Pinson, whom I would like to particularly thank for his enormous help both didactic and personal. I am also grateful to my co-supervisor, Piergiorgio Alotto, for his availability and help.

Many other colleagues and people at the Center for Electric Power and Energy (CEE) have to be thanked for the time they spent discussing some of the topics presented in this work and for the useful tips they provided. Special notice goes to Nicolò Mazzi for his patience and kindness.

A personal thank goes to Kenneth Bernard Karlsson, Head of Energy Systems Analysis Group, who granted me the use of the HPC cluster of DTU, in particular of the nodes of DTU Management. I would also like to thank specifically two of my colleagues at the Department of Management Engineering, Rasmus and Pablo, for their interesting and constructive comments on this work.

Finally, this work would not have been possible without the help of my family and friends: in particular, I would like to express my gratitude to Andreas, Elisa and Maria Teresa for their support and useful proofreading.

# Nomenclature

| Acronyms of the investigated mathematical methods | |
|---|---|
| SVM | Support Vector Machine |
| SVR | Support Vector Regression |
| LS-SVR | Least Squares Support Vector Regression |
| RR-LS-SVR | Recursive Reduced Least Squares Support Vector Regression |
| rRR-LS-SVR | Ranked Recursive Reduced Least Squares Support Vector Regression |
| IRR-LS-SVR | Improved Recursive Reduced Least Squares Support Vector Regression |
| WV | Weights Varying |
| GPK | Gaussian Process Kernel |

| Mathematical symbols | |
|---|---|
| $\alpha_i^{(*)}$ | Lagrange multiplier of the $i$-th constraint |
| $\hat{\boldsymbol{\alpha}}$ | Solution of the dual optimisation problem |
| $\boldsymbol{\alpha}_S$ | Coefficients of the reduced set $S$ of support vectors |
| $a$ | Intercept of the polynomial kernel function |
| $\boldsymbol{\beta}$ | Vector of hyperplane linear coefficients |
| $\hat{\boldsymbol{\beta}}$ | Vector of the optimal hyperplane coefficients |
| $\beta_0$ | Hyperplane intercept |
| $b$ | Linear coefficient of the polynomial kernel function |
| $\gamma$ | Center of the radial kernel function |
| $c$ | Degree of the polynomial kernel function |
| $C$ | Regularisation cost |
| $\varepsilon$ | Margin of the insensitive tube |
| $\hat{f}$ | Estimated output from a model |

| | |
|---|---|
| $\mathbb{H}$ | Hyperplane |
| $i$ | Observation index |
| $I$ | Observations set |
| $J$ | Error function |
| $K$ | Kernel matrix |
| $\mathbb{L}$ | Lagrangian operator |
| $M$ | Value of the margin |
| MAE | Mean Absolute Error |
| MAPE | Mean Absolute Percentage Error |
| $N$ | Number of observations |
| $p$ | Dimension of the feature space |
| $q$ | Support vector index |
| $Q$ | Gramian matrix |
| $r$ | Iteration index for the IRR-LS-SVR training procedure |
| RMSE | Root Mean Square Error |
| $S$ | Subset of support vectors |
| $t$ | Time stamp |
| $\boldsymbol{T}$ | Non-linear vector transformation |
| $\boldsymbol{w}$ | Hyperplane coefficients |
| $\boldsymbol{x}$ | General observation vector |
| $\hat{\boldsymbol{x}}$ | Test observation vector |
| $\boldsymbol{x}_i$ | Vector of the $i$-th observation |
| $\boldsymbol{y}$ | Vector of targets |
| $y_i$ | Target of the $i$-th observation |
| $y^{fore}$ | Output forecast |
| $y^{real}$ | Output measurement |
| $\boldsymbol{z}$ | Non-linear transformation of a general observation |
| $\mathbb{Z}$ | Feature space |
| $\vartheta$ | Ranking function |
| $\mu_i^{(*)}$ | Lagrange multiplier of the $i$-th constraint |
| $\lvert\xi\rvert_\varepsilon$ | Loss function, $\varepsilon$-insensitive tube |
| $\xi_i^{(*)}$ | Slack variable of the $i$-th constraint |
| $\mathbf{0}$ | Vector of zeros |
| $\mathbf{1}$ | Vector of ones |
| $\mathbb{1}$ | Identity matrix |

# Contents

CHAPTER 1

# Introduction

*"WITH GREAT POWER THERE MUST ALSO COME GREAT RESPONSIBILITY!"*

*Stan Lee*

The recent United Nations Conference on Climate Change (COP21), held in Paris on December 2015, confirmed the worldwide commitment towards solving the environmental issue. As stated in the agreement, the participating countries commonly acknowledged the need to limit the average global temperature increase to less than 2°C above pre-industrial levels, [1].

In order to achieve this target a massive employment of renewable sources has to be considered in the development of the future energy system: in particular, the power system will have to deal with non-dispatchable generators, e.g. wind farms and PV plants. This issue, even if negligible in cases of low shares of fluctuating generators, has become and will become of high priority in situations with considerable penetration of renewables in the power generation. For example, as reported by Energinet, the Danish Transmission System Operator (TSO), Denmark has generated 42% of its yearly electricity consumption out of wind power in 2015, [2]. However, the commitment is worldwide: as reported by IRENA (the International Renewable ENergy Agency), the growth of renewable energy capacity has increased exponentially in the last years, [3]. Figure 1.1 gives an idea of the global growth of renewable generation installation in the last few years, also achieved thanks to countries with historically low care about environmental issues. This is the case, for example, of China that has recently faced a steep increase of renewable generation: again the statistics of IRENA report a total installed capacity from renewable sources at 2015 of more than 500 GW only in the Asiatic Republic, [3].

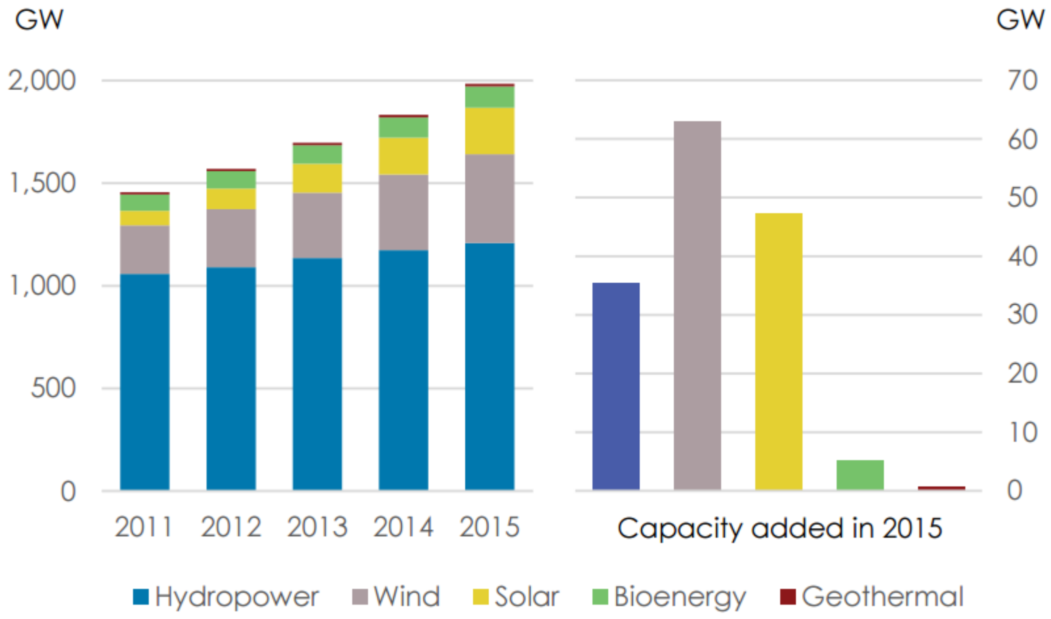

**Figure 1.1:** World renewable generation capacity growth [3]

Recalling the opening quote of this chapter in a more engineering sense, "with great power there must also come great responsibility": the increase of power produced by renewable generators will enhance their responsibility in the power system. They will have to be able to grant all the services (or at least most of them) to the users, now provided mainly by the conventional generators. The non-dispatchability of some of the renewable generators will make the issue of power balance even more critical: the reserve capacity will need to be higher and with faster ramps in order to achieve the same level of security of supply of today. Somebody could claim that all these changes will imply unsustainable costs, even more if considering that the oil price is currently decreasing: but what is the cost of climate change? What if this externality is added on top of the fossil fuel costs?

In order to try to achieve a future fossil-free power system, several solutions have been analysed in the last decades and most of them can be included in the concept of Smart Grid [4]. This term has recently been largely used and abused, however, from a general point of view, it refers to a new and intelligent management of the power system. The increasing stress on the power grids with large shares of renewable generators is leading to an urgent need to apply some smart and cost effective energy technologies. This wider framework involves different research topics: e.g. network interconnections, power reserves, storage systems and Demand Side Management (DSM).

## 1.1 The Demand Response concept

In the European energy efficiency directives, it is stated that not only the generation but also the demand side should access the electricity market, [5]. Moreover, the European Commission has assigned to the European TSOs (ENTSO-E) to commit on defining some network codes to regulate the system operation and market functions inherent to the demand-side participation in the power market balancing. In a recent study by Energinet.dk and the Danish Energy Association, it has been estimated that the total cost of a smart grid solution able to handle a 2025 Danish power system with 50% wind power will be 6.1 billion DKK smaller compared to a solution without a smart management system, [6]. In the last years, the belief that also the consumption side will play a key role in the development of an intelligent power grid has raised; Palensky and Dietrich define it as one of the possible means to stretch the limits of the power grids, [7]. Demand Side Management includes a wide range of different solutions aiming to improve the consumption of power in a flexible way. Palensky and Dietrich in their article divide DSM into four categories (Figure 1.2): Energy Efficiency, Time Of Use (TOU), Demand Response (DR) and Spinning Reserve (SR). On one hand, the first two solutions have already been widely implemented; in particular, Energy Efficiency improvements are meant to decrease the global consumption while the Time Of Use tariffs lead to load shifting and peak shaving effects. On the other hand, Demand Response and Spinning Reserve mechanisms are still open research topics. As displayed in Figure 1.2, their faster response timing could meet the growing need of balancing the fluctuating renewable generation. As shown in Figure 1.3 the effect of the energy efficiency compared to the one of Demand Response is completely different, but only the last one can have a dynamic behaviour. At the same time, this load shifting has to be properly handled through a robust forecasting system in order to avoid unpleasant rebound effects.

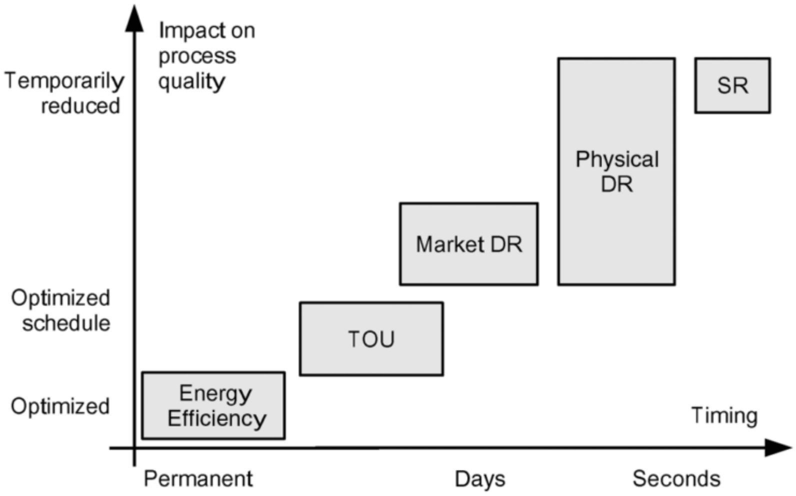

**Figure 1.2:** Categories of Demand Side Management, [7]

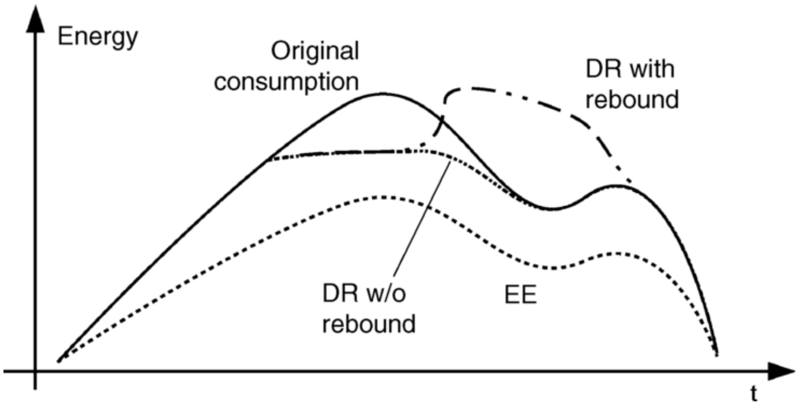

**Figure 1.3:** Impact of Energy Efficiency versus Demand Response, [7]

Several studies have been recently carried out aiming to assess the technical and economic feasibility of implementing a distributed load control system. In [8], O'Connel and others conducted a detailed review of benefits and challenges of Demand Response systems. In particular, the authors argued how the increase of flexibility in the power systems could reduce the costs of the integration of renewable generation. The growing reserve capacity needed to face the higher shares of non-dispatchable generators can be limited by load shifting and curtailment achieved through a Demand Response control system. Instantaneous consumption adjustments (mainly provided by energy services units) would grant the chance to ramp up or down the electrical demand, granting the same effect as a primary frequency control system. In the same way, units with slower dynamics could act as a secondary or tertiary control on the network. These features have widened the idea of Demand Response to the concept of Virtual Power Plant (VPP), as presented in [9]. The latter consists in an aggregation of different Distributed Energy Resources (DERs) and, extending the concept to Demand Response systems, of flexible assets. A smart management of this aggregated resources, from the generation as well as from the consumption side, will allow a further optimisation of the power system both at a local and at a broader level. Locally, the balance between demand and supply as well as solutions to potential grid congestions could be achieved by an internal compensation among the several DERs of the Virtual Power Plant. In a wider perspective, these VPPs could be considered as an interface layer to the grid: e.g. all the grid requests of power balancing could be managed by the Virtual Power Plants through different combinations of all their assets. Working at a more aggregated level allows this particular system to compensate fluctuations and stochastical behaviour of its component, theoretically achieving a higher degree of accuracy and reliability. In order to activate the flexibility of the electricity demand a price-driven approach has been considered so far [10, 11, 12, 13]. This control method is based on the notion of price-elasticity of the aggregated consumption, i.e. the relationship between load and price. In this way, once this function is estimated, a price signal can be optimally defined in order to achieve the desired load response.

The influence of Demand Response could impact the whole energy system in the next future. This should be taken into account while planning the future energy mix because, even though the development of Demand Response is still uncertain, it represents one of the possible solutions to the upcoming issues of the power system. Hence, this open research field is presently of large interest in the energy community.

## 1.2 Problem formulation

In the context presented above, this study aims to investigate the dependency of the electrical load on endogenous and exogenous quantities in order to get a better insight of the Demand Response process and to provide high-quality forecasts of the load, as input to a smart management system. The availability of one of the first datasets collecting measurements from a real implementation of a Demand Response system has given the author the great opportunity of testing all the models developed throughout this analysis on real test case data. This analysis could be of good use for the proper functioning of a load management system since the author expects to develop one or several models, with different levels of accuracy and computational burden, able not only to deliver accurate forecasts but also to represent the relationship between the available inputs and the analysed output. On one side, the ability to predict the future electrical load with a small error is fundamental for energy management systems, that need to know in advance the consumption as well as to assess the availability, in terms of power and time, of the responding load. On the other side, if a well-defined relationship between price and load could be extracted, it will be possible, in a price-based Demand Response system, to compute the optimal price signal that would lead to the desired variation of load in the desired time span.

The methodology followed to carry out this analysis is based on state-of-the-art statistical learning methods, specifically the Support Vector Machine (SVM) and some of its more recent reformulations. This approach allows the author to build a purely data-driven model of the electrical load, not only to provide forecasts but also to have a better insight of its dependency on all the input variables. The need for data mining raised, back in the history, as soon as there was an urge to reconstruct a mathematical relationship of an unknown process starting from its observations, and it is still one of the key points of engineering studies. The analysis of big data through statistical learning has widely spread in the last decades all over each sector. The growing availability of information and the need to extract useful knowledge out of it, have brought much interest in this field. Hence, one of the main focuses of this study will be on the analytical methods needed to be developed in order to grant the highest reliability in terms of forecasting. Given that the main concern of a System Operator in a power grid is the security of supply, detailed research has to be carried out on the stochastic methods employed to provide robust information in a limited time interval to all the agents involved in the Demand Response process.

The learning objectives that the author will try to fulfil with this work can be summarised as:

- understanding of the Support Vector Machine algorithm and its more advanced formulations
- learning of the statistical software R, implementing the investigated statistical methods and testing them on a real case dataset
- analysing the derived models in order to achieve accurate forecasts and to extract input/output relationships
- providing results comparable to state-of-the-art algorithms found in literature.

## 1.3 Structure of the thesis

After the introduction of the framework in which this study is inserted and of the objectives the author expects to achieve, the EcoGrid EU project is presented in Chapter 2. Particular attention is paid to the data gathered throughout the research project since the dataset resulted from its test case is the starting point for this whole analysis. Along with the real data of the implementation of a Demand Response system in Bornholm, an artificial dataset is also described as a reference and validating tool for the different algorithms developed.

Chapter 3 covers all the main mathematical background a reader needs for understanding the Support Vector Machine algorithms employed in this study. This chapter could appear redundant for a reader who is already familiar with the SVM machinery, however, the author believes that a detailed description of the algorithms is essential in order to allow also non-expert readers to understand the methods employed.

In Chapter 4, the structure of the implementation of the different algorithms and of the simulations carried out in this study are presented together with the challenges faced by the author in the coding procedure. Furthermore, the results of the validation on the artificial dataset and of the testing on the real case data are described and discussed. Finally, Chapter 5 includes a recap of the whole study and further considerations on its results and limitations.

# CHAPTER 2

# The test case

In the past decades, Demand Response has been the center of several research studies since it is recognised as one of the key solutions for a smart integration of renewable generators. As explained in [14], so far almost all the implemented applications related to DR mechanisms belong to the so-called *explicit* Demand Response: i.e. solutions helping to maintain the balance in the electrical grid applied either directly by the Transmission System Operator (TSO) and the Distribution System Operator (DSO) (direct load control) or locally through Energy Management Systems (EMS) installed by the users. The *implicit* Demand Response, i.e. more distributed and automated systems of flexible demand as defined in [14], is still an open research topic. Up until a few years ago, this topic has been only studied theoretically; however, recently, some studies have been carried out in Europe and in the USA, through which the research community has started working on more applied and large-scale implementations of *implicit* Demand Response. In particular GridWise [15] and E-Price [16] are two of the best-known examples of real case research projects. This study builds on the work done during one of the presently most advanced project on applied Demand Response and recently winner of the EU sustainable energy award of 2016, [17]: EcoGrid EU.

## 2.1 The EcoGrid EU project

The EcoGrid EU project, [18], took place between March 2011 and August 2015 as an international cooperation between the academic and the business world, aiming to develop one of the first large-scale implementations of a Demand Response system. The high ambition was to integrate a new electricity market structure with load control systems installed in almost 2000 houses in Bornholm, a Danish island already home of advanced test facilities on smart management of the power system, i.e. PowerLab DK [19].

As displayed in Figure 2.1, the electrical generation of Bornholm includes a high share of generation from renewable sources, more than 50% according to [20]. The fluctuations of the RES generators were mainly solved by means of traditional power plants and of a transmission cable to Sweden of considerable capacity. The objectives of EcoGrid EU were to design and implement a large-scale Demand Response system in a power system with high penetration of renewable energy sources. By increasing the flexibility of the power system, it was expected to achieve a decrease of the peak demand, of the wind power curtailment and of the balancing price, as presented in [21], with the aim of integrating the RES generators even more.

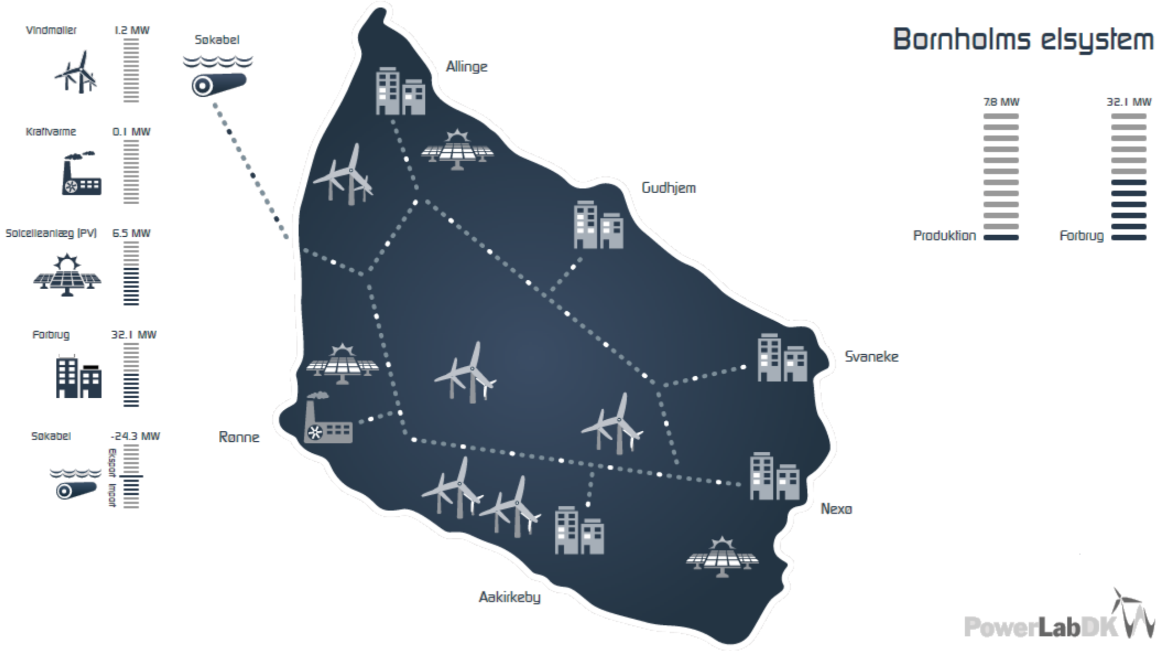

**Figure 2.1:** Overview of the electrical generation of Bornholm, [22]

### 2.1.1 The set-up of EcoGrid EU

The need of integrating a market structure with physical automation systems and, at the same time, granting comfort and security of supply to the customers, implied a great coordination among all the parties involved. Starting from the TSO, its responsibility to ensure the power balance in the electrical grid had to match with its aim to increase the flexible assets coming from the consumption side. At the same time ØSTKRAFT, the local DSO, was involved in the management of both the needs of the customers and of its potential benefits in employing zonal real-time prices to solve congestions of the power system. An optimally defined difference of price between two areas could, in fact, allow preventing potential issues of power flow in the grid, e.g. by reducing the burden on overloaded lines. The basis of the Demand Response system functioning was a real-time price, defined each 5 minutes, able to activate the potential flexibility of each house via its price-elasticity. This relationship between price and demand implies that an increase in price causes, at least theoretically, a decrease of the consumption and vice versa. In order to implement this price-based method, a real-time market has been developed, in which the flexible consumption adjustment was close to the operational time of TSO measures, as displayed in Figure 2.3. A significant effort was also put in the development of a full system of Information and Communications Technology (ICT), that was essential in order to standardise and secure the communication processes between the real-time market platform and the distributed energy sources.

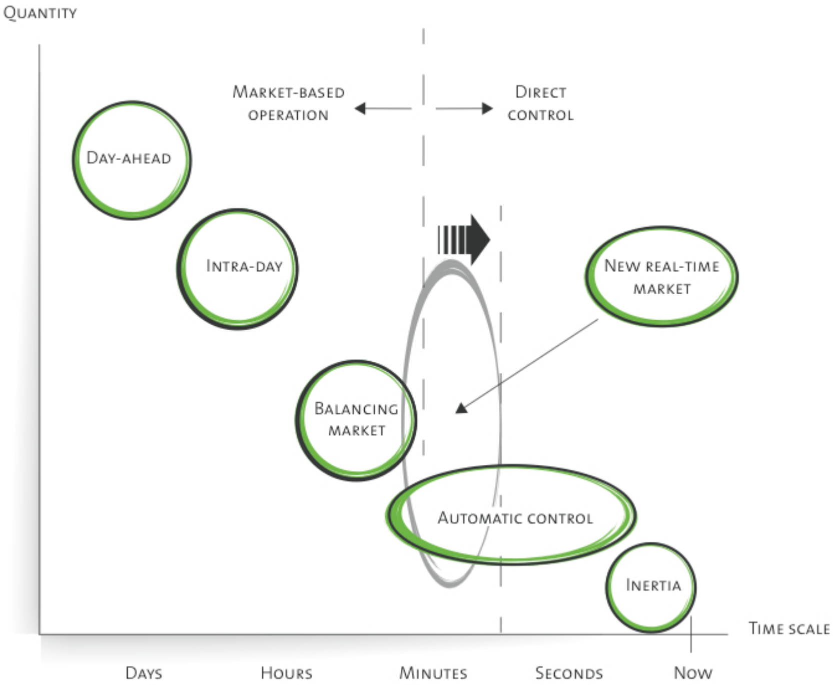

**Figure 2.2:** The implemented real-time market in the context of the power system balance solutions, [21]

What made this project unique was the ability to test the whole system on a large number of real households in Bornholm. A great participation of customers (almost 2000) made it possible to test not only the single control systems but also the communication interface and the real-time market platform. For this reason, it is possible to state that this dataset represents one of the most comprehensive and advanced large-scale data collection of a flexible demand demonstration. The installed load control systems addressed mainly the electric heating and heat-pumps of the houses involved. All participants were equipped with smart meters with time sampling of 5 minutes, but different load control systems have been implemented in order to compare how different technologies could respond to price signals: in particular, all the users could be aggregated into 5 groups, as described below.

**Group 1** In order to have a reference consumption, around 350 houses were equipped with smart meters, but no control systems were implemented. This reference group is fundamental to derive and validate results from all the other groups.

**Group 2** In around 500 houses a manual control was employed: the appliances were not provided with automation systems but the users had access to market information through a website or a mobile app and could adjust their consumption consequently. What is more, a system of feedback and reward points was developed in order to keep the consumers involved in the project by giving information on the results of their changes in the consumption pattern. This group of households was used to determine the awareness and behaviour of consumers towards the implemented system and an estimate of the potential willingness to adapt their consumption was expected to be observed.

**Group 3-4** These two aggregated groups were equipped with a semi-automated Home Energy Management System (HEMS): this solution required relatively cheap and simple instrumentation with a single indoor temperature measurement. Around 270 houses were equipped with automation on air/water or geothermal heat pumps (Group 3) while the control system of around 380 houses acted on electric heating appliances (Group 4).

**Group 5** A fully automated control system with multiple heating zones (i.e. multiple temperature sensors) was installed in almost 450 houses equipped with electric heating. This solution represents a more costly but more accurate control of the user loads.

Some “prosumers”, i.e. producers-consumers usually with local generation from PV or wind turbines, also took part in the project but, given their lower number compared to the households with only heating control, their influence on the results is negligible. Some commercial and industry users were also involved in the project (in total 18) with building energy management systems; however, for the sake of this study, only domestic consumers are taken into account, given the higher, hence more representative, number of users involved.

### 2.1.2 Findings and recommendations of EcoGrid EU

EcoGrid EU proved that it is possible to activate the flexibility of electricity consumption through a price signal with significant results: as reported in its final report, it has been found a reduced cost in the total balancing power of 5.4%, a decrease of 1.2% in the peak load and 80% reduction in wind power curtailment, [21]. At the same time, the project raised attention to some key features, essential for optimising the efficiency of the whole Demand Response system. The most relevant to this study is the importance of accurate forecast models for the electrical load, from which it is possible to predict the available flexibility in the next future. Quoting from the final report of EcoGrid EU [21], "further research, development and utilisation of the EcoGrid EU short-term forecasting model for Demand Response will be needed". Minimising the error on the predicted load will imply a better approximation of the process, hence, a minimisation of the uncertainty on the Demand Response to a defined price signal. Looking at the problem from another perspective, an accurate model of short-term forecasting is a fundamental tool for the clearing of the real-time balancing market: in fact, in this new market design, the balancing demand is a function itself of the price, increasing its elasticity. This relationship between demand and price is extracted from the short-term load forecasting model: in this context, this analysis aims to develop state-of-the-art algorithms to improve the quality of these models.

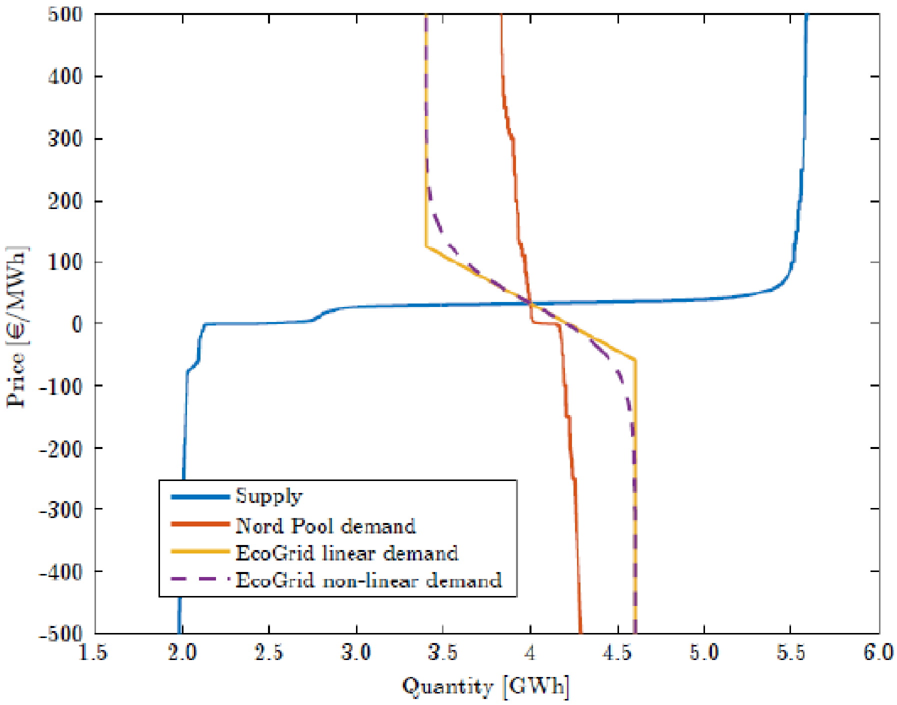

**Figure 2.3:** Elasticity of the balancing demand as in EcoGrid EU report [23]

## 2.2 The dataset

One of the biggest outcomes of EcoGrid EU project is the collection of an enormous amount of data, including not only the metered consumption of all the participating houses but also weather and market information. The uniqueness of this data set allowed the author to apply state-of-the-art algorithms of machine learning on this real and wide test case. Dealing with big datasets, however, implies a great attention at the preprocessing level: out-of-scale or absent measurements are two examples of the problems data analysts have to face at the first step of their analysis. Luckily, this dataset had already been processed and somehow “cleaned” so that the author only had to handle absent measurements. Given the high resolution and the size of the data, skipping the time stamps when one of the inputs was not measured appeared a reasonable choice.

### 2.2.1 Real data from EcoGrid EU

The data collected during the EcoGrid Eu project and used in this study include measurements of electrical loads, prices and external weather parameters recorded each 5 minutes between 00:00 of the 22nd September 2014 and 00:00 of the 8th February 2015. The exogenous variables play a fundamental role in the modelling process of the electric load: in particular, the dependencies on time, temperature and price are predominant. At the same time, all the other measured quantities, e.g. wind speed or air humidity, can always provide additional information making the model even more accurate. It is fundamental, however, to find a trade-off between number of inputs and accuracy since the computational burden of the statistical learning algorithms grows (more or less significantly depending on the algorithm employed) with the dimension of the input space. The input entered to the different algorithms investigated in this study are composed of all the features displayed in Table 2.1 plus some other built-up features, such as:

- hour and weekday indicators to track the repetitive daily pattern of the electrical load and its different behaviour during weekends and weekdays; a sinusoidal transformation is used to ensure circularity to both features

- load and real-time price values in the past hour, i.e. the last 12 measured values, are used in order to include the dependency of the output on its previous values, as in auto-regression models, and from the previous behaviour of the real-time price.

For the sake of this study, some of the available features, i.e. solar irradiation, precipitation and pressure, were not used as inputs for the models developed in this study. The reason for excluding these measurements is that they were sampled with an hourly resolution: consequently, the dynamic of these inputs was perceived by the algorithms as more than 10 times slower than all the other features and, hence, considered meaningless. Some expedients could have been taken into consideration to solve this problem of sampling time, for example by adding a controlled noise to avoid the stationarity of these inputs in the short term.

| ID | Description | Unit |
|---|---|---|
| ts | Unix time stamp | s |
| load_1 | Aggregated load for group 1 | kW |
| load_2 | Aggregated load for group 2 | kW |
| load_3 | Aggregated load for group 3 | kW |
| load_4 | Aggregated load for group 4 | kW |
| load_5 | Aggregated load for group 5 | kW |
| Total load | Total aggregated load | kW |
| RTP | Realtime price adjusted in 5 minute intervals | DKK/MWh |
| HA | Hour Ahead price | DKK/MWh |
| DA | Day Ahead price | DKK/MWh |
| 2MRH | 2 meter relative air humidity | % |
| 2MT | 2 meter temperature | °K |
| 10MWSPEED | Wind Speed | m/s |

**Table 2.1:** Specifics of the dataset from EcoGrid EU

### 2.2.2 Artificial data

When working with large data sets and statistical learning algorithms, the computational time could grow exponentially; therefore it is a good habit to build a simple and small artificial data set, the so-called toy model, through which testing and validating the methods developed.

One of the main focuses of this study is the dependency of the electrical load on two features: real-time price and temperature. For this reason, the artificial data set built by the author aims to provide a rough and simple approximation of this relationship between input and output. Figure 2.4 provides a representation of the two-variable system employed, where the dependency of the load on price has been modelled as a negative arc-tangent while the one on temperature as a combination of hyperbolic and polynomial trends. The relationship with the price describes an approximation of the demand elasticity: in fact, a decrease of the price will lead to an increase of consumption and vice versa. The negative arc-tangent behaviour has been chosen to consider the expected saturation of the demand flexibility for high variations of the real-time price. As for the temperature, it has been tried to implement the strong influence of electric heating on the total electrical load of Bornholm: a hyperbolic behaviour was assigned to decreasing temperatures, assuming that people would be wiling to pay a higher price to warm up their houses in case of very cold days. At the same time, the growth of the load for high temperatures, caused by electric cooling, has been modelled as a slowly increasing polynomial given the low need of cooling in the island of Bornholm.

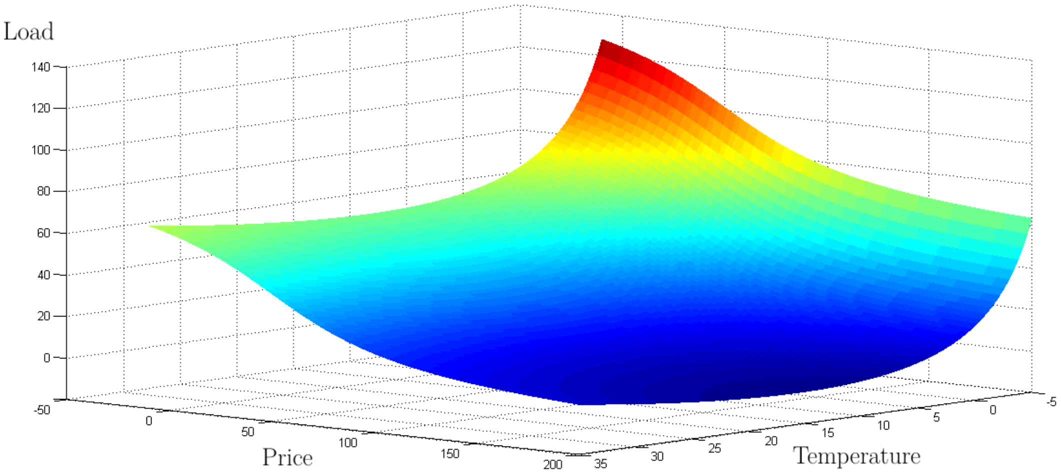

**Figure 2.4:** Toy model dataset representation

The major advantage of an artificial dataset is that the user has total control of the input/output relationship the algorithm is trying to estimate. Hence, it is possible to better understand if the method is working properly and, furthermore, it gives the user the chance to test whether the non-linear dependencies are correctly detected or not. In fact, when using historical data coming from real applications, it is possible to test and validate the accuracy of a forecasting algorithm; however, the same cannot be done with non-linear dependencies, as they are, obviously, not explicitly known.

# CHAPTER 3
# Support Vector Machines

*"First of all, SVM was developed over 30 years. The first publication we did jointly with Alexey Chervonenkis in 1963 and it was about optimal separating hyperplanes. It is actually linear SVM. In 1992, jointly with Bernhard Boser and Isabelle Guyon, we introduced the kernel trick and in 1995, jointly with Corinna Cortes, we introduced slack variables and it became SVM. Why is it so popular? I believe there are several reasons. First of all, it is effective. It gives very stable and good results. From a theoretical point of view, it is very clear, very simple, and allows many different generalizations, called kernel machines. It also introduced pattern recognition to optimization scientists and this includes new researchers in the field. There exist several very good libraries for SVM algorithms. All together these make SVM popular."*

*Vladimir Vapnik*

The history of machine learning takes its roots at the end of the 19th century with the concept of linear regression. The goal of this first approach was the same of the more advanced algorithms developed in the last decades: fitting the data collected from a process to an analytical formulation between input and output. The problem has remained the same, but since the field of statistical learning is in constant expansion and development, the solutions proposed have changed over time, achieving not only more accuracy but also more complexity. The most basic approaches entail, among all, linear regression, expert models and persistence. Even if very simple and naive, these methods act as first benchmarks for more advanced and computationally heavier algorithms. In particular, the persistence method uses the measured output at time $t$ as forecast for time $t+1$; in case of processes with low-frequency changes, this approach can be very efficient and effective. As for the expert models, they can be based on heuristics and/or knowledge of the data analyst about the quantity to forecast. When dealing with white-box models, i.e. when it is possible to properly describe the observed phenomena through an explicit mathematical description, the expert models can be quite effective, depending on the noise of the system measurements. Linear regression, instead, has been widely employed as first and basic attempt to estimate a process output as linear combinations of the input features. Assuming a linear relationship can be quite inaccurate in most of the real processes; however, the simplicity of this method justifies its use as benchmark and as basis for a lot of more advanced algorithms.

A straightforward extension of linear regression implies non-linear transformations of the input variables to be used as features for a general linear regression algorithm. In this way, it is possible to fit the data to a non-linear curve with an easy algorithm. The degree of non-linearity has to be chosen by the data analyst and, in case of no or limited knowledge on the features dependency, it may lead to a high computational burden. An even further development of the linear regression is the so-called Auto-Regressive Moving Average (ARMA) method. The difference towards the linear approaches so far presented is essentially that also the output measurements of a pre-defined past window (Moving Average) are used as input features. In this way, the method is said to auto-regress since it employs the last trend of the output itself to predict its future values.

In the last decades, a new black-box approach has arisen in the data analysis, gaining more and more success: the Artificial Neural Networks. By simulating the brain neurones, these methods have been proved to be very effective in several applications. However, their strong dependency on the network structure, in terms of number of layers and neurones and in terms of type of transfer function of each neurone, leads to a heavy tuning process. State-of-the-art research has developed possible solutions to this issue: recently, the Deep Learning machine has acquired great notoriety in the data mining field. In fact, exploiting one single layer with a high number of neurones and a mechanism of weights randomization, this method achieves accurate results avoiding the complexity of the tuning process of the classic Artificial Neural Networks. Even if the whole data analysis is based on data-driven approaches, the user might prefer having more control over the entire process rather than using a complete black-box algorithm. For this reason, when possible a white-box approach might be preferred: e.g. Kalman filters are employed when the process can be adequately described by explicit mathematical equations. However, in real applications, it is very unlikely to have sufficient knowledge of the system behaviour in terms of analytical description.

For the sake of this study, a more “grey-box” approach has been preferred, granting the user the possibility to somehow control the algorithm. Hence, the Support Vector Machine has been chosen as basis algorithm given its strong mathematical foundations and its higher level of user control. The theory of SVM is briefly presented in Section 3.1, both in terms of classification and regression, while Section 3.2 focusses on the advantages of this method compared to the other statistical learning algorithms. Finally, a deeper insight on the state-of-art of the SVM algorithms is reported in Section 3.3.

## 3.1 Mathematical background

As stated by Vladimir Vapnik, co-inventor of the method, in an interview for the Association for Computational Learning [24], Support Vector Machine has been proved to be a powerful statistical learning algorithm for many different problems. Developed at the end of 20th century as an evolution of optimal separating hyperplanes, this method proposed a new approach to machine learning, including not only an elegant theoretical background but also a practical suitability for real-world problems. In the last decades, Statistical Learning Theory (STL) has been widely employed as a mean for extrapolating unknown dependencies on discrete datasets. The general aim is to minimise the error on the expected output estimated from stochastic input variables (the so-called empirical risk functional) and, at the same time, avoiding the overfitting on the data used for the training. In order to do so, the VC-dimensions[1] of the model, i.e. the complexity of the method itself, are also minimised. This double-target procedure is often addressed as Structural Risk Minimisation (SRM), and Support Vector Machine method has turned out to be one of its best expression, [25].

In order to give the reader the clearest theoretical framework possible, the Support Vector Machine theory is derived starting from the notion of Maximal Margin Classifier, gradually approaching the concept of support vectors and soft margin, and ending with the kernel trick. The classification problem is analysed at first since the SVM was first developed for such applications, but later it is extended to the regression formulation. In order to present a comprehensive explanation of the theory, the whole mathematical description provided below is the combination of different sources, including statistical learning books as [26] and [27], Vapnik own publication [28], a specific tutorial on Support Vector Regression (SVR) by Smola [29] and lectures of Prof. Abu-Mostafa from Caltech University [30].

### 3.1.1 Maximal Margin Classifier

An intuitive and simplified solution of the classification problem, but fundamental as introduction for more advanced approaches, is the concept of separating hyperplanes developed by Pearson at the beginning of the 20th century, [31]. Assuming that a linear decision boundary exists, i.e. the data are linearly separable, it is possible to define a $(p-1)$-dimensional plane that correctly classifies the $p$-dimensional observations. This hyperplane ($\mathbb{H}$) can be generally defined as:

$$\mathbb{H} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = \boldsymbol{\beta^T x} + \beta_0 = 0 \tag{3.1}$$

Once the hyperplane is defined, a test observation ($\hat{\boldsymbol{x}}$) can be classified in relation to which side of $\mathbb{H}$ it belongs: this can be easily checked by computing the sign of $\mathbb{H}(\hat{\boldsymbol{x}}) = \boldsymbol{\beta^T}\hat{\boldsymbol{x}} + \beta_0$. What is more, the distance from $\mathbb{H}$, calculated as

$$\frac{\boldsymbol{\beta^T}}{||\boldsymbol{\beta}||}\left(\hat{\boldsymbol{x}} - \boldsymbol{x}\right) = \frac{1}{||\boldsymbol{\beta}||}\left(\boldsymbol{\beta^T}\hat{\boldsymbol{x}} + \beta_0\right) \tag{3.2}$$

can be seen as a measure of the confidence interval of the probability of each measurement to be correctly classified. The farther from the dividing hyperplane, the more the assigned label is likely to be correct.

[1] VC dimension (or Vapnik–Chervonenkis dimension) describes the capacity of a classification method in terms of cardinality of the biggest set of points shattered by the algorithm.

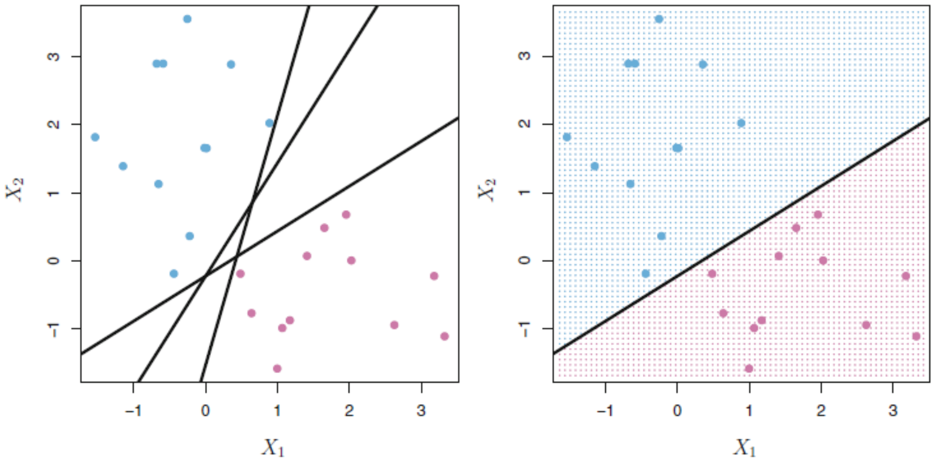

**Figure 3.1:** 2-dimensional representation of some of the possible linear hyperplanes (left) and of the maximal margin classifier (right), [26]

In order to find which, among all the infinite hyperplanes, can separate (always assuming that this is possible) all the observations $i \in I = \{1, \dots, N\}$ in the best possible way, an optimisation problem has to be written as:

$$\begin{aligned} \max \quad & M \\ \text{s.t.} \quad & y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) \geq M \quad \forall i \in I \\ & \sum_{j=1}^{p} \beta_j^2 = 1 \end{aligned} \tag{3.3}$$

where $y_i \in \{-1; 1\}$ $\forall i \in I = \{1, \dots, N\}$ is the label of each observation. As displayed in Figure 3.1, the Optimal Margin Classifier appears to be the one that maximises the margin, i.e. the one with the largest minimal (orthogonal) distance between the hyperplane and the set $I$ of the $N$ observations. Only for the sake of a simpler visualisation Equation (3.3) can be reformulated in a convex quadratic problem assuming that $||\boldsymbol{\beta}|| = 1/M$ resulting in:

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 \\ \text{s.t.} \quad & y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) \geq 1 \quad \forall i \in I \end{aligned} \tag{3.4}$$

In order to further understand this formulation, the Lagrange method is applied to this constrained optimisation model, as displayed below.

$$\mathbb{L} = \frac{1}{2}||\boldsymbol{\beta}||^2 - \sum_{i=1}^{N} \alpha_i[y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) - 1] \tag{3.5}$$

$$\partial_{\beta_0}\mathbb{L} = \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.6}$$

$$\partial_{\boldsymbol{\beta}}\mathbb{L} = \boldsymbol{\beta} - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \tag{3.7}$$

Substituting Equation (3.6) and Equation (3.7) in Equation (3.5) the dual of the optimisation problem is found. This formulation is, normally, the one used for computations since it consists of a convex quadratic problem with linear equality constraints.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T} \boldsymbol{x_j} \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad && \forall i \in I \\ & \alpha_i \geq 0 && \forall i \in I \end{aligned} \tag{3.8}$$

In this way, it is possible to express the quadratic problem in a more compact way:

$$\begin{aligned} \min \quad & \frac{1}{2} \boldsymbol{\alpha^T} Q \boldsymbol{\alpha} - \boldsymbol{1^T \alpha} \\ \text{s.t.} \quad & \boldsymbol{y^t \alpha = 0} \\ & \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{3.9}$$

Given that each entry of the $Q$ matrix is an inner product, $Q_{i,j} =< y_i x_i, y_j x_j >$, $Q$ is a Gramian matrix and hence symmetric and positive semi-definite[2]. By considering all the formulations above, it is possible to derive some interesting considerations. Firstly, the hyperplane $\mathbb{H}$, to be optimal, has to lie exactly in between at least two observations of different classes: their distance to the hyperplane, equal for the considered points, is the optimal margin. Intuitively, it is possible to understand that incrementally moving all observations but those defining the margin will not affect the definition of the hyperplane. Mathematically this can be translated by saying that the marginals of the constraint of the problem Equation (3.4) are zero for all the observations not lying on the margin. This property can also be verified by looking at one of the Karush-Kuhn-Tucker condition:

$$\alpha_i [y_i (\boldsymbol{x_i^T \beta} + \beta_0) - 1] = 0 \quad \forall i \in I \tag{3.10}$$

It is clear that the Lagrange multipliers ($\alpha_i$), i.e. the marginals of the already mentioned inequality constraints, have to be null for all the observations that do not lie on the margin (keeping in mind that the margin M corresponds to the constant 1 after the assumption of $\|\beta\| = 1/M$). Therefore, the only points that impact the solution are those with the shortest distance to the hyperplane: an incremental movement of one of them will imply a new optimal margin classifier. These observations, indeed, hold the hyperplane and, for this reason, they are called support points.

[2] These two properties ensure that the convex quadratic problem has a solution $\bar{x}$ if and only if there exists a $\bar{y}$ such that $(\bar{x}, \bar{y})$ is a Karush-Kuhn-Tucker pair.

### 3.1.2 Support Vector Classifier

One of the two strong assumptions made so far is that the data have to be perfectly separable, i.e. a well-defined hyperplane exists classifying the training observations with no error. However, in real applications this is not likely the case: data are always affected by noise or by stochastic behaviours that lead to fuzzier boundaries between different classes. In order to expand the method developed in the previous section, the concept of margin is relaxed to the so-called soft margin. In this way, the algorithm allows some observations to lie inside or even on the wrong side of the margin. The slack variables $\xi$ are introduced to model this soft margin in the optimisation problem. The basic idea is that $\xi_i$ represent the percentage[3] of the margin violated by each observation, i.e. $y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) \geq M(1 - \xi_i)$, so that:

- $\xi_i > 1$ for observations on the wrong side of the hyperplane: e.g. observations 11 and 12 of Figure 3.2
- $0 < \xi_i < 1$ for points on the correct side of the hyperplane but on the wrong side of the margin: observations 1 and 8
- $\xi_i = 0$ for observation on the correct side of the hyperplane or on the margin: i.e. all the remaining observations

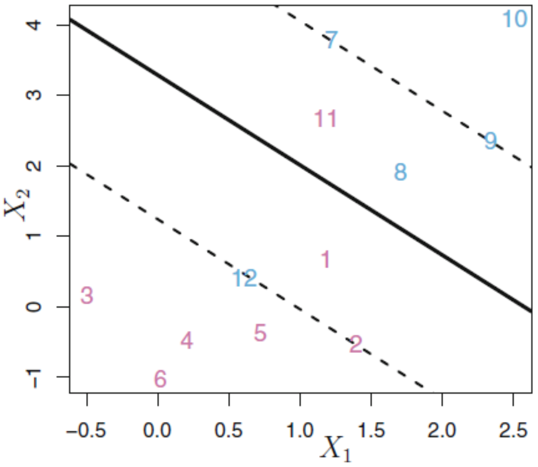

**Figure 3.2:** 2-dimensional representation of a soft margin classifier, [26]

With the same assumption over $\|\boldsymbol{\beta}\|$ made in the previous formulations, the problem can be written as

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i \quad \forall i \in I \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad\;\; \forall i \in I \end{aligned} \tag{3.11}$$

A new parameter, $C$, is included in this formulation in order to limit the observations allowed not respecting the margin classification. In the model just displayed, in fact, different values of $C$ will imply distinct equilibria between the norm of the coefficients vector $\boldsymbol{\beta}$ and the number of violation of the margin. Therefore, the “cost” $C$ works as a regularisation parameter used for tuning the model: the larger $C$, the more narrow the margin will be since there will be less violations allowed, this implies a higher level of data fitting and consequently a lower bias but a bigger variance on the testing error. Conversely, the right opposite happens for small values of $C$.

[3] Using the actual distance from the margin will lead to a non-convex problem, [27]

Applying the Lagrange method, the dual formulation can be derived:

$$\mathbb{L} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) - (1-\xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i \tag{3.12}$$

$$\partial_{\xi_i}\mathbb{L} = C - \mu_i - \alpha_i = 0 \tag{3.13}$$

While writing the Lagrange function and setting its partial derivatives to zero, one can notice that the partial derivatives over $\boldsymbol{\beta}$ and $\beta_0$ are equal to those of the previous model, Equation (3.6) and Equation (3.7). What is interesting is that the formulation of the dual problem

$$\begin{aligned} \max \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i^T}\boldsymbol{x_j} & \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 & \forall i \in I \\ & 0 \leq \alpha_i \leq C & \forall i \in I \end{aligned} \tag{3.14}$$

results quite similar to Equation (3.8): the variables $\mu_i$, in fact, are simplified and the new constraint, Equation (3.13), simply adds an upper bound to the variables $\alpha_i$. The resulting formulation is still a quadratic convex problem, hence all the considerations made in the previous section are still valid. Moreover, looking at some of the Karush-Kuhn-Tucker conditions will give a better insight of the method.

$$\alpha_i[y_i(\boldsymbol{x_i^T}\boldsymbol{\beta} + \beta_0) - (1-\xi_i)] = 0 \tag{3.15}$$

$$\mu_i\xi_i = 0 \tag{3.16}$$

When $\xi_i = 0$, meaning that the margin is not violated, the situation is the same as the one previously discussed, i.e. the coefficients $\alpha_i$ are not zero only for the observations that lie on the separating hyperplane (recalling the definition of support points). Conversely, in case of violations of the margin, i.e. $\xi_i > 0$, the Lagrangian multipliers $\mu_i$ are forced to be zero, hence $\alpha_i = C$, as imposed by Equation (3.13). The definition of Support Vectors comes straightforward from these considerations as all the observations $i \in S$ such that $\alpha_i \neq 0$, i.e. all the points that violate or lie on the margin. Once the optimisation is solved, the solution $\hat{\boldsymbol{\alpha}}$ is a vector containing all the weights necessary to define the separating hyperplane. Recalling Equation (3.7), in fact, the coefficients vector $\hat{\boldsymbol{\beta}}$ is computed as:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N}\hat{\alpha}_i y_i \boldsymbol{x_i} = \sum_{i\in S}\hat{\alpha}_i y_i \boldsymbol{x_i} \tag{3.17}$$

It is possible to notice, how the classification boundary is defined only through a linear combination of the support vectors. Computational wise, this represents a big advantage since the number of support vectors can be somehow controlled. The tuning parameter $C$ plays a key role in this process: in fact, decreasing the degree of data fitting (leading to smoother estimating functions) imply a smaller amount of observations needed to support the separating hyperplane. The trade-off between flatness of the fitting curve and accuracy will be discussed further in this analysis.

### 3.1.3 Kernel trick

The second strong assumption to overcome is the requested linearity of the hyperplane: in real applications, it is unlikely that a system could be accurately estimated by a linear dependency on different inputs. In order to deal with non-linearities, a first straightforward approach is to enlarge the input space considering as features also non-linear functions of the inputs themselves, through a general transformation $\boldsymbol{T} = (t_1, \ \dots\ , t_m)$ such that:

$$\boldsymbol{T} : \boldsymbol{x} \mapsto \boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = (t_1(\boldsymbol{x}), \ \dots\ , t_m(\boldsymbol{x})) \tag{3.18}$$

In this way, the regression is done linearly on non-linear features; the non-negligible drawback is that the computational burden of this process increase exponentially and still only a finite number of non-linear components can be exploited.

Including these new features in the Support Vector Machine formulation, will lead to the same model structure with $\boldsymbol{z}$ as input vector rather than $\boldsymbol{x}$. Therefore the problem can be written as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{z_i^T} \boldsymbol{z_j} & \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 & \forall i \in I \\ & 0 \leq \alpha_i \leq C & \forall i \in I \end{aligned} \tag{3.19}$$

It is possible to notice that solving this quadratic problem in the case of a high dimensional vector $\boldsymbol{z}$ can easily become computationally very heavy. However, the intuition at the base of the SVM method is that the transformation $\boldsymbol{T}$ is not needed in an explicit form as long as the inner product $\boldsymbol{z_i^T} \boldsymbol{z_j}$ $\forall i, j$ is known.

Rewriting Equation (3.17), one could argue that the coefficients of the separating hyperplane are function of the vector $\boldsymbol{z}_i$ and not of inner products between two observations:

$$\hat{\boldsymbol{\beta}} = \sum_{i \in S} \hat{\alpha}_i y_i \boldsymbol{z_i} \tag{3.20}$$

However, the equation of the boundary is not essential in an explicit form for the solution of the classification problem. What is fundamental is the equation of the estimated function which depends, again, only on the inner product $\boldsymbol{z_i^T} \boldsymbol{z_j}$ $\forall i, j$:

$$\hat{f}(\boldsymbol{z}) = \hat{\boldsymbol{\beta}}^T \boldsymbol{z} + \hat{\beta}_0 = \sum_{i \in S} \hat{\alpha}_i y_i \boldsymbol{z}_i^T \boldsymbol{z} + +\hat{\beta}_0 \tag{3.21}$$

As for $\hat{\beta}_0$, it can be calculated only in terms of the inner product of $\boldsymbol{z}$ by solving $y_i \hat{f}(\boldsymbol{z}) = 1$ for all observations that lie on the margin.

For this reason, the concept of Kernel functions has been included in the formulation of the SVM method, defined as:

$$k(\boldsymbol{x}, \boldsymbol{x'}) =< \boldsymbol{T}(\boldsymbol{x}), \boldsymbol{T}(\boldsymbol{x'}) >= \boldsymbol{z}^T \boldsymbol{z'} \tag{3.22}$$

The main achievement of the kernel trick is that it enables the representation of the inner product between high dimensional (even infinite) vectors as a function of a dot product of smaller (always finite) ones.

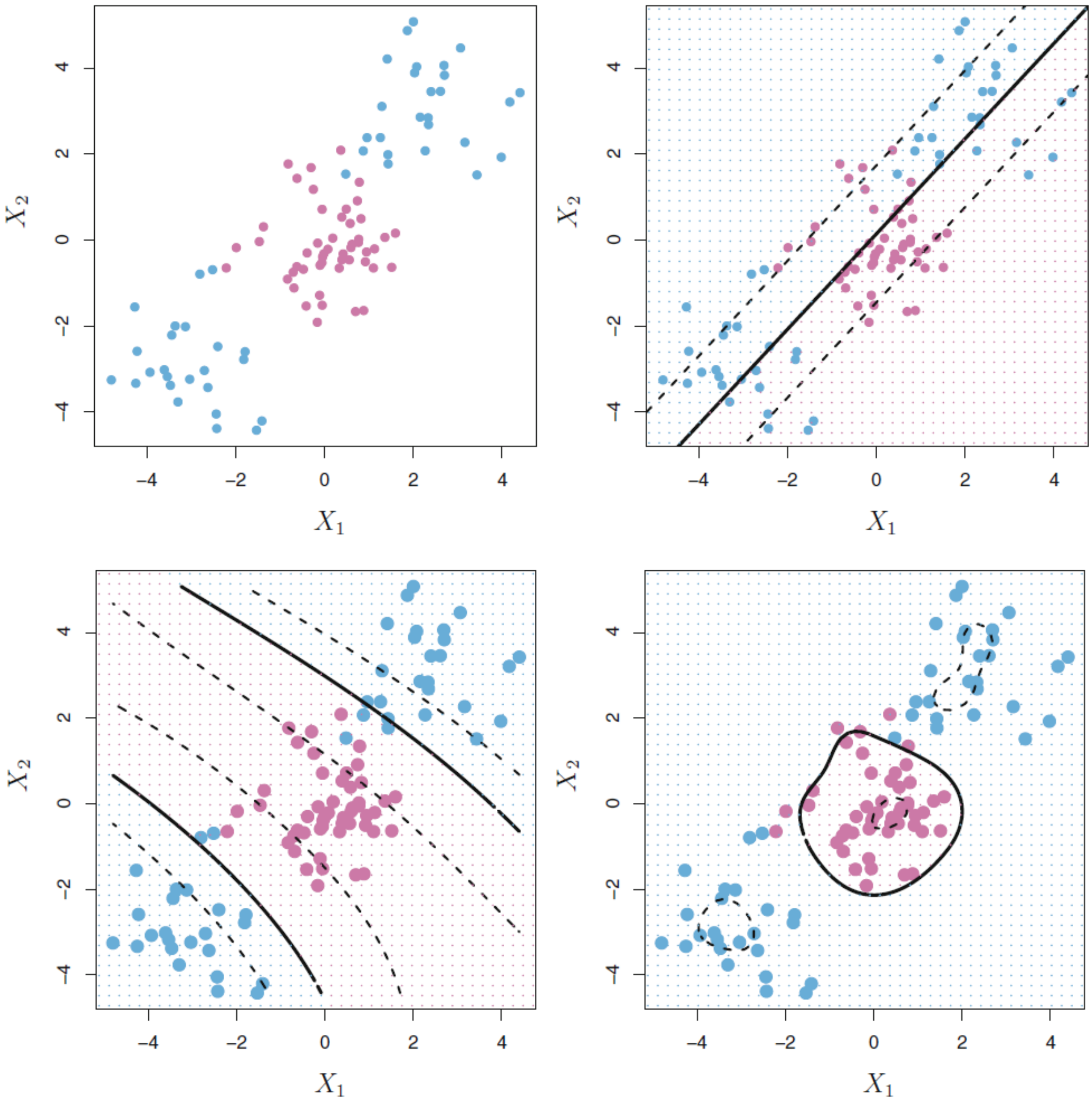

**Figure 3.3:** Decision boundary of a non-linear separable dataset (top-left) with three different kernel transformations: linear (top-right), polynimial of degree 3 (bottom-left) and radial (bottom-right), [26]

A clarifying example of this kernel trick can be easily seen considering two features ($\boldsymbol{X} = (x_1, x_2)$) and a quadratic polynomial kernel:

$$\begin{aligned} k(\boldsymbol{X}, \boldsymbol{X'}) &= (1+ < \boldsymbol{X}, \boldsymbol{X'} >)^2 = (1 + x_1x'_1 + x_2x'_2)^2 \\ &= 1 + 2x_1x'_1 + 2x_2x'_2 + (x_1x'_1)^2 + (x_2x'_2)^2 + 2x_1x'_1x_2x'_2 \end{aligned} \tag{3.23}$$

It is clear how this kernel function corresponds to the use of an inner product in a larger feature space $\boldsymbol{z} = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$. The great advantage of the kernel trick is that there is no need for knowledge of neither the transformation $\boldsymbol{T}$ nor the expanded space. A more general formulation of the polynomial kernel can be derived per mathematical induction, leading to:

$$k(\boldsymbol{X}, \boldsymbol{X'}) = (a < \boldsymbol{X}, \boldsymbol{X'} > +b)^c \quad \forall c \in \mathbb{R} \tag{3.24}$$

It is noteworthy to state that the inner product is limited on the dimension of the basic inputs ($\boldsymbol{X}$), hence it is possible to benefit from a great number of non-linear features without explicitly calculating them. As for the example of a polynomial kernel of order $c$, once the dot product $< \boldsymbol{X}, \boldsymbol{X'} >$ is calculated, the resulting number is raised to the power of $c$. Conversely, going through the explicit transformation, one should compute an inner product between two vectors that include each feature raised to the power of 1 to $c$ plus all the mixed terms.

The full potential of the kernel functions can be appreciated by analysing the radial basis kernel:

$$k(\boldsymbol{X}, \boldsymbol{X'}) = e^{(-\gamma \|\boldsymbol{X} - \boldsymbol{X'}\|^2)} \tag{3.25}$$

The exponential function, using Taylor expansion, can be written as a linear combination of infinite number of addends: in terms of its use as a kernel function it implies that the non-linear features used to define the separating hyperplanes are infinite. However, the computational time is the same as using a linear kernel. In Figure 3.3, it is possible to appreciate the power of this kernel trick in case of datasets with a high level of non-linearity. The only condition for this extremely powerful tool is that the expanded space $\mathbb{Z}$ where the inputs are mapped by the kernel function has to exist. At the same time, its definition cannot be provided explicitly unless the advantage of the kernel trick would vanish. In order to grant the existence of the space $\mathbb{Z}$ through a valid transformation $\boldsymbol{T}$, two approaches can be followed:

1. **Construction**: as for the polynomial kernel function, Equation (3.24), it was possible to carry out the calculations in a lower dimensional space and then, inductively, to extend the proof to a general polynomial degree

2. **Mercer Condition**: this theorem states that $k(\boldsymbol{X}, \boldsymbol{X'})$ is a valid kernel function if and only if $k(\boldsymbol{X}, \boldsymbol{X'}) = k(\boldsymbol{X'}, \boldsymbol{X})$, granting the symmetry of the inner product in the $\mathbb{Z}$ space, and that the kernel matrix ($K$),

$$K = \begin{bmatrix} k(\boldsymbol{x_1}, \boldsymbol{x'_1}) & k(\boldsymbol{x_1}, \boldsymbol{x'_2}) & \cdots & k(\boldsymbol{x_1}, \boldsymbol{x'_N}) \\ k(\boldsymbol{x_2}, \boldsymbol{x'_1}) & k(\boldsymbol{x_2}, \boldsymbol{x'_2}) & \cdots & k(\boldsymbol{x_2}, \boldsymbol{x'_N}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x_N}, \boldsymbol{x'_1}) & k(\boldsymbol{x_N}, \boldsymbol{x'_2}) & \cdots & k(\boldsymbol{x_N}, \boldsymbol{x'_N}) \end{bmatrix}$$

is positive semi-definite for all observations $i = 1, \, ... \, , N$

For the sake of this study, already proved kernel functions were considered: e.g. linear, polynomial and radial. Therefore, there is no need to mathematically verify that the transformation is valid. However, as for more advanced studies are concerned, the Support Vector Machines method allows the user to implement new kernel functions, as long as they respect the above mentioned properties.

### 3.1.4 Support Vector Regression

The same method, previously described, can also be applied to regression problems. The goal, in this formulation, is to define not an hyperplane but an hyper-tube that gives the best estimation of the input-output dependencies. Therefore, the Support Vector Regression (SVR) formulation becomes:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - (\boldsymbol{\beta^T z} + \beta_0) \leq \epsilon + \xi_i \qquad \forall i \in I \\ & y_i - (\boldsymbol{\beta^T z} + \beta_0) \geq -\epsilon - \xi_i^* \quad \forall i \in I \\ & \xi_i, \xi_i^* \geq 0 \qquad\qquad\qquad\quad\; \forall i \in I \end{aligned} \tag{3.26}$$

As for the SVM, also the SVR considers not only the accuracy but also the *flatness* (i.e. non-overfitting) of the solving function in the optimisation problem. The approach is to find an $\epsilon$-insensitive tube that limits all the data analysed (using the same kernel trick) but, at the same time, some observations are allowed violating this constraint. This expedient is adopted in order to avoid overfitting similarly to what described for the classification problem. As displayed in Figure 3.4, for the sake of simplicity the loss function associated to the variables $\xi_i^{(*)}$ is assumed to be:

$$|\xi|_\epsilon := \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ |\xi| - \epsilon & \text{otherwise} \end{cases} \tag{3.27}$$

However, the user is free to implement any other loss function as far as correctly modelled in the optimisation problem: this great customisation is one of the features of the SVM machinery that makes the algorithm extremely portable for many different applications.

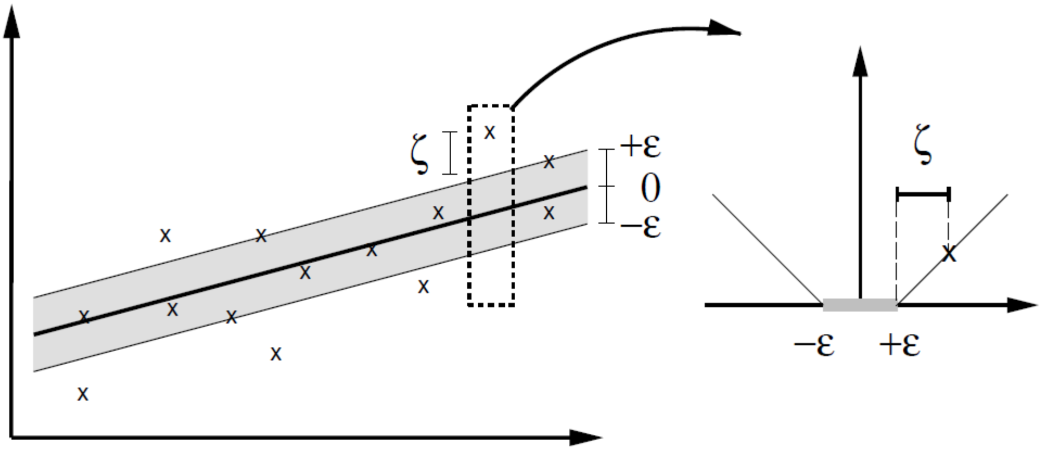

**Figure 3.4:** $\varepsilon$-insensitive tube representation, [29]

Following the same procedure as the previous sections, the Lagrange function associated to Equation (3.26) becomes:

$$\begin{aligned}\mathbb{L} =& \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\mu_i\xi_i + \mu_i^*\xi_i^*) - \sum_{i=1}^{N}\alpha_i[\epsilon + \xi_i - y_i + \\ &+ \boldsymbol{\beta^T z} + \beta_0] - \sum_{i=1}^{N}\alpha_i^*[\epsilon + \xi_i^* + y_i - \boldsymbol{\beta^T z} - \beta_0]\end{aligned} \tag{3.28}$$

$$\partial_{\beta_0}\mathbb{L} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0 \tag{3.29}$$

$$\partial_{\boldsymbol{\beta}}\mathbb{L} = \boldsymbol{\beta} - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\boldsymbol{z_i} = 0 \tag{3.30}$$

$$\partial_{\xi_i^{(*)}}\mathbb{L} = C - \mu_i^{(*)} - \alpha_i^{(*)} = 0 \tag{3.31}$$

Consequently the dual formulation of the optimisation problem appears slightly more complex than the one for the classification problem. However, it is still a convex quadratic problem and can be written as:

$$\begin{aligned}\max \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\boldsymbol{z_i^T z_j} - \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)y_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \quad \forall i \in I \\ & 0 \le \alpha_i^{(*)} \le C \qquad \forall i \in I\end{aligned} \tag{3.32}$$

The optimal solution $\hat{\alpha}_i^{(*)}$ of the model is used to compute the equation of the fitting hyperplane and of the estimated function as:

$$\hat{\boldsymbol{\beta}} = \sum_{i \in I^*} (\hat{\alpha}_i - \hat{\alpha}_i^*) \boldsymbol{z_i} \tag{3.33}$$

$$\hat{f}(\boldsymbol{z}) = \hat{\boldsymbol{\beta}}^T \boldsymbol{z} + \hat{\beta}_0 = \sum_{i \in I^*} (\hat{\alpha}_i - \hat{\alpha}_i^*) \boldsymbol{z}_i^T \boldsymbol{z} + \hat{\beta}_0 \tag{3.34}$$

The same considerations already made on the support vectors are still valid for the regression problem, since, even if the formulation is slightly different, the basis of the algorithm is the same as the SVM. The support vectors are identified by all the observations $\boldsymbol{x_i}$ that lie outside or on the edge of the $\epsilon$-tube. For these points, the corresponding kernel transformations $\boldsymbol{z_i}$ show a non-zero Lagrange multiplier. It is noteworthy to mention that the support vectors lie in the $\mathbb{Z}$ space, while in the input space only their pre-images under the transformation $\boldsymbol{T}$ can be displayed. Additionally, several loss functions can be employed in order to weight the observations that do not lie on the fitting hyperplane in a different way. In this case, the analytical formulation will get more complicated but the basic approach will still be the same.

## 3.2 Advantages and drawbacks of SVM

As already argued in the previous sections, the Support Vector Machine formulation benefits of some peculiarities that make it one of the most attractive methods in the current scenario of statistical learning. While digging deeper in its mathematical formulation, the reader could appreciate the robustness and the elegance of the kernel trick and of the algorithm in general: in fact, the method appears at the same time simple but solid, powerful but intuitive. Furthermore, its clear mathematical foundations increased its public acceptance as valuable method among the experts of the research field.

In particular, the idea of selecting some of the observations as representative for the whole dataset through an optimisation problem underlines how the Support Vector Machine includes a structure minimisation procedure, i.e. a regularisation process. The algorithm, through its cost parameter $C$, aims to avoid overfitting and to ensure a certain level of smoothness to the estimated function. At the same time, the kernel trick allows the method to operate in a higher dimensional hyperplane (even infinite), without the need to cut the non-linear approximation at some finite number of features. The latter is probably the greatest achievement of the algorithm since it can theoretically represent each non-linear behaviour as linear through a map that is not even necessary to be explicitly defined. Moreover, the SVM has been recently investigated in several studies and proved to be efficiently working on different datasets. Hence, it is possible to state that it could be ranked as one of the state-of-the-art algorithms for classification and time series analysis.

However, the Support Vector Machine comes with some drawbacks. Firstly, the training procedure becomes computationally heavy the more observations ($N$) are employed: Platt estimates the training time to be proportional to $O(N^3)$ for standard solvers of the quadratic optimisation problem, [32]. This burden can be reduced by means of more sophisticated and optimised solving algorithms: e.g. using the Sequential Minimal Optimization (SMO) method, the training time could be reduced to $O(N^2)$. Secondly, the use of non-linear kernel as maps from the inputs to the features space decreases the level of transparency of this model. The dependencies on the inputs, in fact, are not explicitly and analytically derivable; however, it is possible to extract them by use of some targeted simulations as presented further in this study.

At last, the whole algorithm is not only time but also memory consuming: the need for storing the support vectors (whose number could be non-negligible) makes the SVM, in its simple formulation at least, not very suitable for real-time applications. For these reasons, more advanced reformulations of the algorithms presented so far have been investigated, aiming to find a good trade-off between accuracy and online usability.

## 3.3 State-of-art of SVM methods

Thanks to its solid formulation, the Support Vector Machine has recently stepped up in the machine learning community as an excellent trade-off between solid mathematical basis and data-driven approach. For this reason, there have been recent studies and research on this topic aimed at modifying the method structure, turning it suitable for real-time and real cases application. In a scenario of a real-time electricity market, in which price signals are produced with small time intervals, e.g. each 5 minutes, it becomes fundamental employing a data analysis machinery that could grant not only good accuracy but also high computational speed. The problem of the computational speed could be solved by employing more powerful processors; however, it is important to remember that control systems for Demand Response are far from being a mature technology. Hence, limiting the components costs maintaining high performances is the main focus for the future economic feasibility of this new technology.

In this framework, several reformulations of the SVR algorithm have been proposed in literature: in particular, the first issue analysed has been the computational cost of the quadratic optimisation problem. When dealing with high dimensional data, as the case of Demand Response in which both electrical and weather measurements and forecasts are used as input, the computational burden can increase enormously. For the sake of this study, it has been chosen to focus the analysis on the Least Squares Support Vector Regression (LS-SVR) as a solution for avoiding the quadratic optimisation problem.

### 3.3.1 Least Squares Support Vector Regression

As aforementioned, the purpose of the Least Squares Support Vector Regression method is to get rid of the computationally heavy quadratic optimisation problem in favour of a lighter formulation. However, the price to pay is an increase on the memory needed to store the model results as explained further in this section. The LS-SVR method can be written as:

$$\begin{aligned} \min \quad & \frac{\|\boldsymbol{\beta}\|^2}{2} + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t.} \quad & y_i - (\boldsymbol{\beta^T z} + \beta_0) = \xi_i \qquad \forall i \in I \end{aligned} \tag{3.35}$$

The idea, on which this reformulation is based, is simply to apply the least squares approach to the SVR formulation: therefore, the method aims to minimise both the structure of the model (tracking the minimum coefficients) and the distance ($\xi_i$) between each observation and the hyperplane. In order to further simplify the optimisation problem, the two inequality constraints are transformed in a single equality constraint. In this way, the benefit of employing an $\epsilon$-insensitive tube, and consequently the possibility to define the hyperplane only on a subset of the observations, fails since each observation is considered as support vector.

However, at the same time, having only one constraint lead to simpler partial derivatives of the Lagrangian:

$$\mathbb{L} = \frac{\|\boldsymbol{\beta}\|^2}{2} + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(\xi_i - y_i + \boldsymbol{\beta^T z} + \beta_0) \tag{3.36}$$

$$\partial_{\beta_0}\mathbb{L} = \sum_{i=1}^{N}\alpha_i = 0 \tag{3.37}$$

$$\partial_{\boldsymbol{\beta}}\mathbb{L} = \boldsymbol{\beta} - \sum_{i=1}^{N}\alpha_i z_i = 0 \tag{3.38}$$

$$\partial_{\xi_i}\mathbb{L} = C\xi_i - \alpha_i = 0 \tag{3.39}$$

Substituting the partial derivatives of the Lagrangian in the equality constraint, one obtains:

$$\begin{cases} y_i - \sum_{j=1}^{N}\alpha_j \boldsymbol{z_j z_i} - \beta_0 = \alpha_i/C \\ \sum_{i=1}^{N}\alpha_i = 0 \end{cases} \tag{3.40}$$

By use of matrix notation, the LS-SVR can be summarised as:

$$\begin{bmatrix} 0 & \boldsymbol{1^T} \\ \boldsymbol{1} & K + \mathbb{1}/C \end{bmatrix} \begin{bmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix} \tag{3.41}$$

with $K$ the kernel matrix and $\mathbb{1}$ the identity matrix.

As it is possible to notice, the formulation is quite compact and, above all, directly solvable: hence, the quadratic optimisation problem has been completely overcome reducing the problem to the solution of a linear system. In terms of time, the training procedure becomes faster if compared to the SVR method; however, the testing and, in general, the prediction with new inputs get slower since all the observation are used as support vectors in the computation of the output forecast. This behaviour is not very suitable for real-time applications, where some batch training is performed on historical data and predictions are made online when new inputs/measurements are available. At this point, one could argue that the SVR still performs better in terms of forecasting time; however, a deeper insight on the LS-SVR method leaves room for improvement.

### 3.3.2 Recursive Reduced Least Squares Support Vector Regression

While looking at the model coefficients, i.e. the weights of all the training data in the estimation of new outputs, it is possible to notice that a good amount of their magnitudes is low or close to zero. Hence, several methods have been found in literature aiming to increase the sparsity of the model, which, in this case, means decreasing the number of support vectors while maintaining a high accuracy. For the sake of this study, two sparsity improving algorithms have been investigated, both proposed by Zhao and others respectively in [33] and in [34].

On the first hand, the Recursive Reduced Least Squares Support Vector Regression (RR-LS-SVR) is investigated: this method considers only a randomly selected subset of the observations as support vectors. The recursivity of the method lies on the fact that all the targets ($\boldsymbol{y}$) are taken in consideration during the optimisation of the coefficients, which are calculated as:

$$(R + ZZ^T) \begin{bmatrix} \alpha_0 \\ \boldsymbol{\alpha_S} \end{bmatrix} = Z\boldsymbol{y} \tag{3.42}$$

$$\text{where} \quad R = \begin{bmatrix} 0 & \boldsymbol{0^T} \\ \boldsymbol{0} & K/C \end{bmatrix}, \quad Z = \begin{bmatrix} \boldsymbol{1^T} \\ \hat{K} \end{bmatrix} \quad \text{with } \hat{K}_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_j}), j \in S$$

In this way, the accuracy of the method depends on the number of support vector chosen by the user: as always a trade-off between accuracy and sparsity has to be found, by carrying out some simulations with different numbers of support vectors.

Along with the formulation found in literature, another approach has been tried in this study by selecting the support vectors based on the weights of the LS-SVR model: a ranked Recursive Reduced Least Squares Support Vector Regression (rRR-LS-SVR). The subset (S) of support vectors is then defined by evaluating the absolute values of the related coefficients, considering as support vectors only the ones contributing the most to the model; the computation of the coefficients is the same as in Equation (3.42). This approach is slower than the previous one because it requires the full training of the

LS-SVR algorithm; however, the author expects that randomly choosing the subset of support vectors could lead to redundancy and lower accuracy values. The advantage of this approach is that once the LS-SVR is trained, the selection of different numbers of support vectors is very fast, and several testing procedures can be carried out without exceeding on time.

On the other hand, in the second investigated algorithm, proposed by Zhao and others, i.e. the Improved Recursive Reduced Least Squares Support Vector Regression (IRR-LS-SVR), the support vector space is built iteratively, adding a new support vector ($\boldsymbol{q}$) based on a custom ranking function, $\vartheta$. Following the article mentioned, the ranking function used is the variation of the Lagrangian ($\mathbb{L}$):

$$\vartheta_q^{r+1} = \mathbb{L}^r - \mathbb{L}^{r+1} = \lambda \left( \begin{bmatrix} \mathbf{1} \\ \hat{K} \\ \hat{\boldsymbol{k}}_{\boldsymbol{q}} \end{bmatrix} \boldsymbol{y} \right)^T \begin{bmatrix} \boldsymbol{\delta} \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^T & -1 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \hat{K} \\ \hat{\boldsymbol{k}}_{\boldsymbol{q}} \end{bmatrix} \boldsymbol{y} \tag{3.43}$$

$$\text{with} \quad \begin{cases} \lambda = \left( \frac{k_{qq}}{C} + \hat{\boldsymbol{k}}_{\boldsymbol{q}}^T \hat{\boldsymbol{k}}_{\boldsymbol{q}} - [\ \boldsymbol{k}_{\boldsymbol{q}}^T \mathbf{1} \quad \hat{\boldsymbol{k}}_{\boldsymbol{q}}^T \hat{K}^T + \frac{\boldsymbol{k}_{\boldsymbol{q}}^T}{C}\ ] \boldsymbol{\delta} \right)^{-1} \\ \boldsymbol{\delta} = U^n \begin{bmatrix} \mathbf{1} \boldsymbol{k}_{\boldsymbol{q}} \\ \hat{K} \hat{\boldsymbol{k}}_{\boldsymbol{q}} + \frac{\boldsymbol{k}_{\boldsymbol{q}}}{C} \end{bmatrix} \\ k_{qq} = k(\boldsymbol{x}_{\boldsymbol{q}}, \boldsymbol{x}_{\boldsymbol{q}}) \\ \hat{\boldsymbol{k}}_{\boldsymbol{q}} = [\ k(\boldsymbol{x}_{\mathbf{1}}, \boldsymbol{x}_{\boldsymbol{q}}) \quad \cdots \quad k(\boldsymbol{x}_{\boldsymbol{N}}, \boldsymbol{x}_{\boldsymbol{q}})\ ]^T \\ \boldsymbol{k}_{\boldsymbol{q}} = [\ k(\boldsymbol{x}_{\mathbf{1}}, \boldsymbol{x}_{\boldsymbol{q}}) \quad \cdots \quad k(\boldsymbol{x}_{\boldsymbol{S}}, \boldsymbol{x}_{\boldsymbol{q}})\ ]^T \end{cases}$$

At each iteration ($r$), the matrix $U$ and the weights are updated as

$$U^{r+1} = \begin{bmatrix} U^r & \mathbf{0}^T \\ \mathbf{0} & 0 \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{\delta} \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^T & -1 \end{bmatrix} \tag{3.44}$$

$$\begin{bmatrix} \alpha_0^{r+1} \\ \boldsymbol{\alpha}_S^{r+1} \\ \alpha_q^{r+1} \end{bmatrix} = \begin{bmatrix} \alpha_0^r \\ \boldsymbol{\alpha}_S^r \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{\delta} \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^T & -1 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \hat{K} \\ \hat{\boldsymbol{k}}_{\boldsymbol{q}} \end{bmatrix} \boldsymbol{y} \tag{3.45}$$

in order to keep the method up-to-date during the whole process and to avoid the redundancy of the aforementioned random selection of support vectors. The tuning of the model could be time intensive given that the training time of this algorithm is higher than, for example, the normal LS-SVR. However, this procedure combines the sparsity that characterised the SVR method and the direct solvable optimisation problem of the LS-SVR. For this reason, theoretically it seems to be a competitive trade-off between accuracy and computational burden: the customisable sparsity of the method, in fact, makes it suitable for a recursive adaptation on the fly as described later in this study.

Some troubles have been faced in the implementation of this algorithm: in fact, setting the number of support vectors not always implied “convergence”. With a high (but still very reasonable) number of support vectors allowed, the method was found to crash because the matrix $U$ reached extremely large values. This was at first interpreted as a sign of divergence of the algorithm; however, a deeper insight on the method allowed the author to realise that it was just a numerical problem. The matrix $U$ was, in fact, exponentially growing in norm because the algorithm was trying to approximate too much the data: in particular, the gradient difference became small enough to numerically ill-condition the problem. This issue was easily solved by allowing the method to stop whenever the difference of Lagrangian gets below a defined threshold, i.e. when adding a new support vector brings negligible information to the model.

The advantage in term of computational burden is significant with both algorithms. As explained in [34], the SVR has a training time proportional to the cube of the number of observations ($O(N^3)$) while the RR-LS-SVR and IRR-LS-SVR bring their computational costs respectively down to $O(N^2)$ and $O(S\dot{N}^2)$, with S the number of support vectors. However, the computational cost in memory is decreased from $O(N^2)$ of the SVR and LS-SVR to $O(S\dot{N})$. This achievement is the most important considering that the training is usually performed offline (therefore the time needed is not fundamental) while the cost of memory is proportional to the time necessary to use the models for online applications.

### 3.3.3 Recursive Adaptation

A further solution for improving the accuracy of the online load forecasting without slowing down the whole procedure is to employ a recursive adaptation of the coefficients based on the forecasting error occurred at the previous time sample. The approach is the classic gradient descent method, aiming at minimising the squared error between the forecast and the actual load measurement. It has been chosen to investigate two slightly different adaptive recursions proposed by Sun and others in [35]: the basic Weight Varying method (WV) and the so-called Gaussian Process Kernel (GPK).

**Weights Varying** As for this method, the idea is to adapt on the fly the coefficients ($\boldsymbol{w} = [\ \beta_0 \quad \boldsymbol{\alpha}\ ]^T$) weighting the support vectors with a gradient descent method based on the squared error ($J^2$) between the measured output ($y^{real}$) and the forecast value ($y^{fore}$):

$$\operatorname{argmin}_{\boldsymbol{w}} J^2 = \operatorname{argmin}_{\boldsymbol{w}} (y^{real} - y^{fore})^2 = \operatorname{argmin}_{\boldsymbol{w}} [y^{real} - \beta_0 - \sum_{j \in S} \alpha_j k(\boldsymbol{x_j}, \boldsymbol{x})]^2 \tag{3.46}$$

The iteration of the gradient descend, at time step $t$, becomes:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - [\nabla^2(J^2)]^{-1}\nabla(J^2) \tag{3.47}$$

$$\text{with} \begin{cases} \nabla(J^2) = -2J \begin{bmatrix} 1 \\ k(\boldsymbol{x_j}, \boldsymbol{x}) \end{bmatrix} \\ \\ \nabla^2(J^2) = 2 \begin{bmatrix} 1 \\ k(\boldsymbol{x_j}, \boldsymbol{x}) \end{bmatrix} \begin{bmatrix} 1 & k^T(\boldsymbol{x_j}, \boldsymbol{x}) \end{bmatrix} \end{cases}$$

It is a good habit to avoid the inversion of the Hessian ($[\nabla^2(J^2)]^{-1}$) at each iteration in case of high dimensionality of the matrix, as in the case of the LS-SVR trained with a large number of observations. For this reason, it has been decided to use a fixed coefficient weighting the gradient even if it leads to a suboptimal solution: this figure has been tuned by hand around the order of magnitude of the usual values of the inverse of the Hessian.

As for all the other methods, the computation of the inverse of the Hessian can be carried out in reasonable time intervals; hence the author decided to stick to the proper mathematical formulation of the method trying to achieve the maximum accuracy possible.

**Gaussian Process Kernel** The idea at the basis of this formulation is to expand the kernel functions, each with a constant ($\boldsymbol{\alpha^c}$) and a linear ($\boldsymbol{\alpha^l}$) term, during the adaptive recursion. In this way, a more accurate and faster adaptation is hoped to be achieved. The formulation follows strictly the one of the previous approach, apart from the presence of new variables in the forecast value:

$$y^{fore} = \beta_0 + \sum_{j \in S} [\alpha_j k(\boldsymbol{x_j}, \boldsymbol{x}) + \alpha_j^c + \alpha_j^l \boldsymbol{x_j^T x}] \tag{3.48}$$

Hence, the gradient and the Hessian become respectively:

$$\nabla(J^2) \;=\; -2J \begin{bmatrix} 1 \\ k(\boldsymbol{x_j}, \boldsymbol{x}) \\ \mathbf{1} \\ \boldsymbol{x_j^T x} \end{bmatrix} \tag{3.49}$$

$$\nabla^2(J^2) \;=\; 2 \begin{bmatrix} 1 \\ k(\boldsymbol{x_j}, \boldsymbol{x}) \\ \mathbf{1} \\ \boldsymbol{x_j^T x} \end{bmatrix} \begin{bmatrix} 1 & k^T(\boldsymbol{x_j}, \boldsymbol{x}) & \mathbf{1}^T & \boldsymbol{x_j^T x} \end{bmatrix} \tag{3.50}$$

The same considerations presented for the Weights Varying approach are valid also for this method since the dimensions of the matrices involved are proportional to the ones of the WV algorithm. Therefore, the formal implementation of the gradient descent is employed for all methods but the LS-SVR, for which a fixed value is used to replace the inverse of the Hessian matrix.

# CHAPTER 4
# Short Term Load Forecasting

It is well known that one of the key targets of the whole power system is to grant the balance between generation and consumption. Much effort has been recently put into forecasting with high accuracy both the renewable power production and the electrical demand. In particular, with the aim of developing a Demand Response system, Short Term Load Forecasting (STLF) has become a fundamental research topic. In order to compensate power fluctuations also from the consumption side, it is necessary to be able to forecast the demand in a future time interval as well as to estimate its dependency on other parameters, e.g. on the real-time electricity price. In this way, assuming a real-time market or any other suitable structure, an optimal price signal can be computed and communicated to the responsive loads achieving a variation of the consumption as close as possible to the optimal one. This whole process is characterised by several stochastic variables; therefore robust and accurate forecasting algorithms have to be investigated aiming to minimise the expected risk of error. In this framework, several studies have already been carried out: [36], [37] and [38] are just some examples of the literature available on the topic.

In this chapter, at first, the structure of the simulations is presented together with the implementation of all the methods employed. Secondly, the results of the toy model are discussed as a mean of testing the different algorithms on a lighter and easier environment and, finally, the real dataset is investigated and the results are presented.

## 4.1 The simulation structure

All the simulations were run using the same structure: the author has tried to optimise not only the computations but also the readability and portability of the code. For these reasons, all the investigated methods were implemented as separated functions that could work with different datasets or different parameters. The skeleton of the simulation routine is:

1. Pre-processing of data
2. Training routines with the different methods
3. Testing and error computation
4. Non-linear dependency extraction

### 4.1.1 Pre-processing of data

The pre-processing of data appears to be one of the key processes in the whole data analysis: in fact, even the most advanced learning algorithm will produce meaningless results when wrong inputs are entered. Regarding the features selection, a Support Vector Machine based algorithm is not very resilient to meaningless features. The weighting coefficients are assigned to a full set of observed quantities and not to the specific features as it happens, for example, with a linear regression approach. For this reason, the author decided to exclude some of the features available in the dataset as already discussed in Section 2.2.1. Furthermore, the scaling of the data in input is fundamental for a proper functioning of almost every statistical learning: in fact, employing features with a large difference of numerical order will ill-condition the problem. Normalising the data with zero variance and a Gaussian distribution will instead equalise the potential contribution of each feature to the model.

### 4.1.2 Training and tuning

The second step in the simulation structure is the training of the models over the pre-processed inputs through one or several learning routines. The tuning of the algorithm parameters, i.e. the regularisation cost $C$ and the centers of the radial kernels $\gamma$, is carried out by way of a full grid search on all the possible combinations. For each of them, the model is trained and validated according to a ranking based on the forecast error. As a consequence, this part of the simulations is the most time consuming one but, at the same time, it represents the batch learning of real-time applications. Hence, at this stage, the training time has less importance rather than accuracy since it is typically carried out before the application itself starts working on online measurements. However, if considering a retraining of the model once in a while during the operation of this intelligent system, the training time gains some relevance again. For the sake of this study, the algorithms analysed are the ones whose mathematical formulation is explained in Chapter 3 and two benchmark models.

**Benchmark models** In order to assess the performances of each statistical method implemented, it is a good habit to build one or few benchmark models. These simplified approaches are mainly used to provide an idea of the order of magnitude of the forecast error and to allow users to check the relative improvement of more advanced methods. In this study, especially for the real dataset application, it has been chosen to employ two benchmark methods:

1. Persistence: often used as a benchmark in forecasting problems, this method assumes that the predicted output has the same value of the output at the previous time stamp. For problems with high-resolution samplings (like the case of the real dataset application where data are gathered each 5 minutes), this method could provide a meaningful benchmark since the output is not highly varying in a small time interval.

2. Auto-regression: slightly more elaborated, this algorithm carries out a linear regression on a user-defined number (12 in this study) of previous output values to compute the forecast. As the persistence, also this method has decent performances in case of slowly varying outputs; hence, it is suitable for the real dataset application of this study.

**Support Vector Regression** The Support Vector Regression (SVR) algorithm has been employed through the package *e1071* of the library *LIBSVM*, available from the repository of the statistical software R, [39]. The library offered both tuning and training functions to the user; however, the tuning process has been manually coded by the author mainly for two reasons:

1. The grid of possible combinations of the different parameters (i.e. the cost coefficient and the magnitude of the radial basis kernel) has been built as a logarithmic, rather than linear, scale between user-specified upper and lower bounds

2. The tuning code has been parallelised on several cores in order to further speed up the whole process. This achievement has been reached by means of another R package, *snowfall*, and was not possible with the *e1071* package for SVM.

**Least Squares Support Vector Regression** The implementation of the Least Squares Support Vector Regression (LS-SVR) algorithm has been entirely coded by the author: in fact, no libraries of R have been found that allow an effective application of this method to regression problems. The structure of the developed code includes three different functions: the first and basic one simply solve the LS-SVR problem for a defined set of input and tuning parameters, while the other two are used to run the k-fold cross validation and tuning procedures. It is noteworthy to mention that these functions are not highly optimised as the one provided by the official R libraries: hence, while carrying out the comparison of the computational time among the different methods, this has to be taken into account. On the contrary, the custom implementation allowed the author to parallelise the computation as much as possible: i.e. not only for the tuning, as aforementioned, but also for the validation process.

**Reduced Least Squares Support Vector Regression** As for the reduced versions of the LS-SVR algorithm, no help from R repositories has been employed; in the case of the rRR-LS-SVR, the tuning and validation routines have not been coded since the model parameters have been taken from the previously trained LS-SVR model. However, for the RR-LS-SVR and the IRR-LS-SVR algorithms, both a tuning and validation routine have been implemented. In particular for the IRR-LS-SVR method, the routine appears particularly slow: as examined later in this work, the author thinks that there could be room for a considerable improvement.

### 4.1.3 Testing procedure

During the testing routine, each method provides its forecasts which are compared with the respective real measurements. In order to calculate the output, the aforementioned package *e1071* provides a specific function (*predict*) which has been used for the SVR algorithm. As for all the others methods a similar function has been implemented by the author. As presented in the previous chapter, a further development of online algorithms consists in adapting the coefficients, weighting the support vectors, each time a new measurement is available. Hence, two functions, one for each method analysed (WV and GPK), have been implemented allowing the user to choose whether to invert the Hessian matrix or to use a fixed approximation of it.

## 4.2 Monitored parameters

Before analysing the results of the different simulations, it is important to define some reference parameters to compute in order to carry out meaningful discussions and comparisons. For the sake of this application, the key points are accuracy, time and memory employment. Hence, the following criteria would be considered for each of the implemented algorithms:

1. the Root Mean Square Error (RMSE) is computed as an indicator of accuracy. The figure obtained is neither an absolute value nor a percentage; in fact, it is calculated on standardised data (in terms of mean and standard deviation). Therefore the RMSE has to be intended as percentage of standard deviation and has been calculated as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i^{fore} - y_i^{real})^2}{N}}$$

2. the number of support vectors defines the memory needed for storing the model information and it also gives information about the time necessary to predict

the output when new observations are available. When it comes to coefficients adaptation, the number of support vectors also identifies the size of the matrix to invert and, consequentially, its computational burden.

3. the training time completes the information on the time burden, even though, among all the parameters, it brings the least meaningful information since the batch learning can be carried out in advance with no real time constraints. It is calculated for 100 tuning simulations with a 10-fold cross validation each. All the simulations have been parallelised on 20 cores at maximum[4].

What is important to notice is the splitting of the observations among training, validation and testing sets. Given the amplitude of this dataset, the author decided to chronologically divide the available data into two halves. The first half is used to tune the model on a random sub-selection of 5000 observations; on these data a 10-fold cross validation has been carried out by dividing the subset into 10 more sets and performing a leave-out-one validation. The algorithms are later tested on the remaining part of the first half of the observations: it is, however, worthy to mention that this testing is not completely meaningful since the observations used to build the different models are randomly collected from the same subset the models are tested on. Therefore, it is expected that the models would estimate with a good accuracy the first half of the data. The second half of the data is left for just a testing routine with and without an online adaptation of the coefficients of the models. This choice aims to test the robustness of the different methods as well as to simulate a real case problem: in fact, one would like to run this machine as soon as possible, that is without waiting to collect, for example, a whole year of data. It is also known that all the algorithms can be periodically updated, with a new training on the newly available data; however, for the author, this structure seemed to be the clearest and most consistent way to test the algorithms investigated.

### 4.2.1 Non-linear dependencies extraction

The kernel trick that allows exploiting up to an infinite dimensional feature space, however, has some drawbacks; the process of extracting the output dependency on a specific feature becomes complicated and not straightforward. The main cause of this non-transparency comes from the definition of the coefficients weighting the new data input to the model. They are defined, in fact, through a weighted sum of a not explicitly known non-linear transformation of the support vectors. As deeper analysed in the previous section, this problem is overcome by the use of the kernel functions; however, this does not simplify the dependency extraction. In particular, the kernel functions are computed using both the support vectors and the new input to the model, therefore, they lose of generality when aiming to define a relationship only between one feature (or a subset of features) and the output. This extraction, however, is fundamental in

[4] It was possible to use all this computational power by means of the HPC cluster of DTU in particular of the nodes of DTU Management, by courtesy of Kenneth Bernard Karlsson - Head of Energy Systems Analysis Group

the case of a Demand Response application: in fact, one of the main keys for a good system control is to understand the load dependency on the real-time price. In this way, it would be possible to compute the optimal price signal, i.e. the one that theoretically causes the wanted load variation. In order to get rid of the dependency on all the other features, a sort of Monte Carlo approach has been exploited: a big number (100000) of simulations has been carried out trying to stochastically limit the dependency on all the non-investigated features. The procedure followed consists in:

1. a selection of 100000 random observations from the first half of the dataset as well as of random percentage variations (in the range $\pm 3\sigma$)
2. increase the investigated feature of each selected input by the correspondent relative variation
3. if the newly investigated feature value exceeds the maximum and minimum values of the original dataset the relative variation is randomly changed as long as the computed figure does not lie in between the feature boundaries
4. predict, by means of each statistical model, the output resulting from the selected inputs at their original values and with a variation of the investigated feature

In this way, the author expects to extract the sought relationships by means of the average output variation caused by a deviation of one of the input features from their original value. The author believes that this procedure, even if it is not the only possible way to extract the features dependency, is the one providing the best compromise between simplicity and accuracy. The same approach is, therefore, applied to both the toy model and the real dataset: the former as a way to validate the described procedure and the latter trying to extract the electrical load dependency on price and temperature.

## 4.3 Toy model

As already mentioned in Section 2.2.2, tuning and testing a complex statistical learning algorithm could be highly time-consuming, above all during the first stages of coding and implementation. Therefore, the toy model is employed: this custom dataset, in fact, helps data analysts to test their methods in a light computational environment and, at the same time, to have perfect control and knowledge of the input/output relation.

The approach that has been followed was to firstly set up a custom dataset; secondly to test all the developed algorithms with training, validation and testing sequences and eventually the extraction of the non-linear dependencies was carried out. The artificial dataset employed is the one described in Section 2.2.2; a white noise signal has been added to this approximation of the electrical load in order to simulate an even more realistic situation.

### 4.3.1 Forecasting results

In order to validate the implemented algorithms, all the methods have been tested on the toy model. The results achieved are displayed in Table 4.1 in terms of training time, number of support vectors and the RMSE both for the validation and testing routines.

As a benchmark for this dataset, the persistence method was employed; however, the results show a high testing RMSE (neither training and validation processes are needed for this approach). This error, of around 1.4 in terms of standard deviation, can be explained by the fact that the dataset has been built with random and sequentially uncorrelated samples of the inputs. Thus, the process is not smooth in time and cannot be defined quasi-stationary: for this reason, forecasting the following lead time with the last measurement brings a considerable error. However, the advantage of the toy model is that the error on the data is controlled by the user; therefore the standard deviation of the white noise added on top of the output, in this case 0.0885, can be used as a more meaningful benchmark for the model. The standard deviation of the noise equals the error that one will make in case of perfect knowledge of the system, allowing the author to use it as a lower bound for the RMSE.

The first algorithm tested is the basic formulation of the Support Vector Regression (SVR). The results achieved by this method in terms of testing RMSE reach a level of accuracy almost as good as the perfect foresight model. During the tuning procedure, the algorithm accomplishes an RMSE slightly lower than the standard deviation of the noise (0.0860). The overfitting can be excluded since the testing RMSE (0.0896) also displays a value very close to the benchmark one. The model employs 1326 out of the 5000 observations as support vectors, reducing the complexity of the system but achieving an almost perfect accuracy.

| **Method** | **Training Time [s]** | **Support Vectors** | **Validation RMSE** | **Testing RMSE** |
|---|---|---|---|---|
| SVR | 986 | 1326 | 0.0860 | 0.0896 |
| LS-SVR | 399 | 4500 | 0.0837 | 0.0905 |
| rRR-LS-SVR | 400 | 50 | - | 0.3236 |
| RR-LS-SVR | 54 | 50 | 0.1075 | 0.1196 |
| IRR-LS-SVR | 1671 | 38 | 0.0851 | 0.1006 |

**Table 4.1:** Results of the simulations on the toy model

Even if the training procedure is mainly performed offline, for problems with larger dimensional feature space and a higher number of training observations it is important, as already mentioned, to reduce the computational cost also of the tuning, training and validation routines. Therefore, the LS-SVR method is tested, achieving a more than 2 times faster training routine with a validation error even lower, 0.0837. This low value could be a first sign of overfitting, derived from the employment of all the 4500 observations (1/10 of the observations are used for the validation in the tuning process) as support vectors in the model. The testing RMSE, in fact, shows a slightly lower accuracy (0.0905). Even if the model still performs very well, the worsening of the forecasting error points out a lower robustness of the model towards future variation of the described process.

As mathematically described in the previous chapter, one of the main drawbacks of the LS-SVR is the employment of all observations as support vectors. In this way the risk of overfitting is higher but, above all, the model becomes computationally heavier in the estimation routine. In case of an online application where the operating speed (in terms of forecasting and of recursive coefficient adaptation) is one of the requirements, the LS-SVR algorithm, even if it grants a good accuracy, is not advised to be employed. For this reason, recent reformulations of the algorithm have been developed aiming to improve the sparsity of the model; for the sake of this study three of them have been implemented and tested. At first, the ranked version of the Recursive Reduced Least Squares Support Vector Regression, rRR-LS-SVR, was tested with the number of support vectors set to 50: it was expected to still get a good accuracy but the results show a testing RMSE of 0.3236.

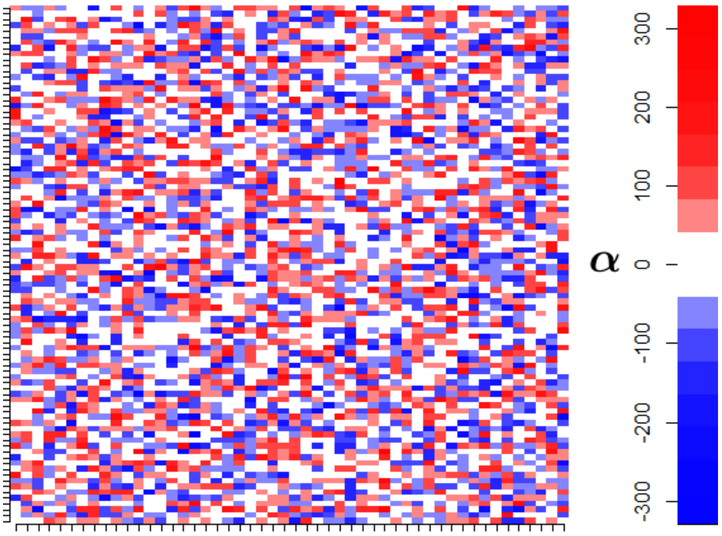

**Figure 4.1:** Graphical representation of the support vectors weights, $\boldsymbol{\alpha}$, of the LS-SVR algorithm for the toy model

However, by looking at Figure 4.1, where the weights of the support vectors, $\boldsymbol{\alpha}$, of the LS-SVR model are chromatically represented, one could easily notice that the white spots, corresponding to the coefficients with lower absolute value, are not predominant. This implies that the model needs many observations to describe the process and that the rRR-LS-SVR, using only the first 50 of those support vectors ranked according to the absolute value of their weights, cannot provide an accurate forecasting model. It is also noteworthy to mention that the rRR-LS-SVR method is a simple post-processing of the LS-SVR method; hence, no tuning nor validation is considered. For this reason, the validation RMSE is absent and the training time has been computed as the sum of the LS-SVR one and the time needed for the calculation of the new coefficients.

In order to improve even more the performance of the LS-SVR algorithm a random support vector selection has been implemented according to the already explained RR-LS-SVR algorithm. Given that no relationship with the previously trained LS-SVR model exists, a tuning and validating procedure were implemented and the number of support vectors was set to 50. The model provides accurate results; furthermore, a lighter optimisation problem and a lower dimensional space lead to a very fast training procedure. The RMSE is still close to the one with perfect foresight of the model both in the validation and testing routines, even if the model is an approximation of the basic SVM machinery. This algorithm combines speed, computational lightness and a good level of accuracy maintaining a satisfying robustness. The testing RMSE (0.1196), in fact, is only 11.25% higher if compared to the validation one (0.1075), while, for example, the LS-SVR method shows an increase in the RMSE of 8.12% but employing 90 times more support vectors.

As for the IRR-LS-SVR, the optimisation of the number of support vectors is even higher: the recursivity of the algorithm avoids the dependency on the random choice of support vectors, by adding to the model only the most meaningful observations. Hence, the method describes the process through 38 observations, achieving a forecast accuracy comparable to the SVR. The testing RMSE (0.1006) shows a slightly worsening from the validation one (0.0851); however, the model error is still close to the perfect foresight one (0.0860). One could argue that the training time of the IRR-LS-SVR algorithm is larger than all the other methods: on one hand, the training time, as already mentioned, is not a fundamental criterion since the training is not performed online. On the other hand, the reader has to be reminded that this method has been entirely coded by the author and, therefore, cannot compete in terms of time optimisation with the functions embedded in some R packages. Moreover, the statistical software R is not the optimal solver of recursive loops like the one needed for this algorithm: this could be one of the causes that slow down the whole tuning procedure. Nonetheless, providing a balanced trade-off between accuracy and online computational burden (low number of support vectors), this algorithm could be a competitive alternative to the SVR in case of a higher dimensional problem.

### 4.3.2 Non-linear dependencies

After having proven that the analysed methods have good performances in terms of forecasting, the toy model is employed to check whether the investigated algorithms are able to provide a description of the non-linear dependencies between the output and the input features. It is possible to reliably use the toy model in order to validate the methods since the user has total control of the input-output relationship.

Figure 4.2 displays the extracted relationships between output and the two inputs, simulating a simplification of the electrical load dependency on price and temperature. Each graph shows the behaviour of a process feature, estimated by means of a sort of Monte Carlo simulation: for this reason, both the mean value (solid line) and the standard deviation (coloured area) are displayed.

As it is possible to notice, the testing on the toy model validates the proper functioning of almost all the methods: in fact, the rRR-LS-SVR method, that showed the worst forecasting error, does not perfectly catch the non-linear trend. A growing error is faced at the two ends of the graph, not only for the rRR-LS-SVR method but also for the remaining algorithms. This could be explained by the fact that the ends of the graph describe a range of data that rarely appears on the dataset; hence, less knowledge on their behaviour is given and also needed.

The magnitude of the error (both in terms of mean and standard deviation) changes among the models in a consistent way with the forecasting results. The SVR and LS-SVR present a behaviour almost identical to the reference one on the non-linear reconstruction. However, also the RR-LS-SVR and the IRR-LS-SVR non-linear dependency extraction follows the reference one with good accuracy. Since both of them consist in an approximation of the LS-SVR, the author expected to see some signs of loss of information; however, both algorithms proved to be very competitive in terms of mean and of standard deviation of the estimated quantity.

It is possible to conclude that all algorithms can be considered validated and ready to be tested on a real and extremely more complicated dataset. In particular, the IRR-LS-SVR method proved to be competitive with both the SVR and the LS-SVR in terms of accuracy while granting faster performances for an online application given its low number of support vectors. Furthermore, it has been found a link between forecasting and dependency extraction performances: the lower the forecasting error, the more accurate the model (overfitting might always be taken into account) and, hence, the better representation of the features relationship with the output.

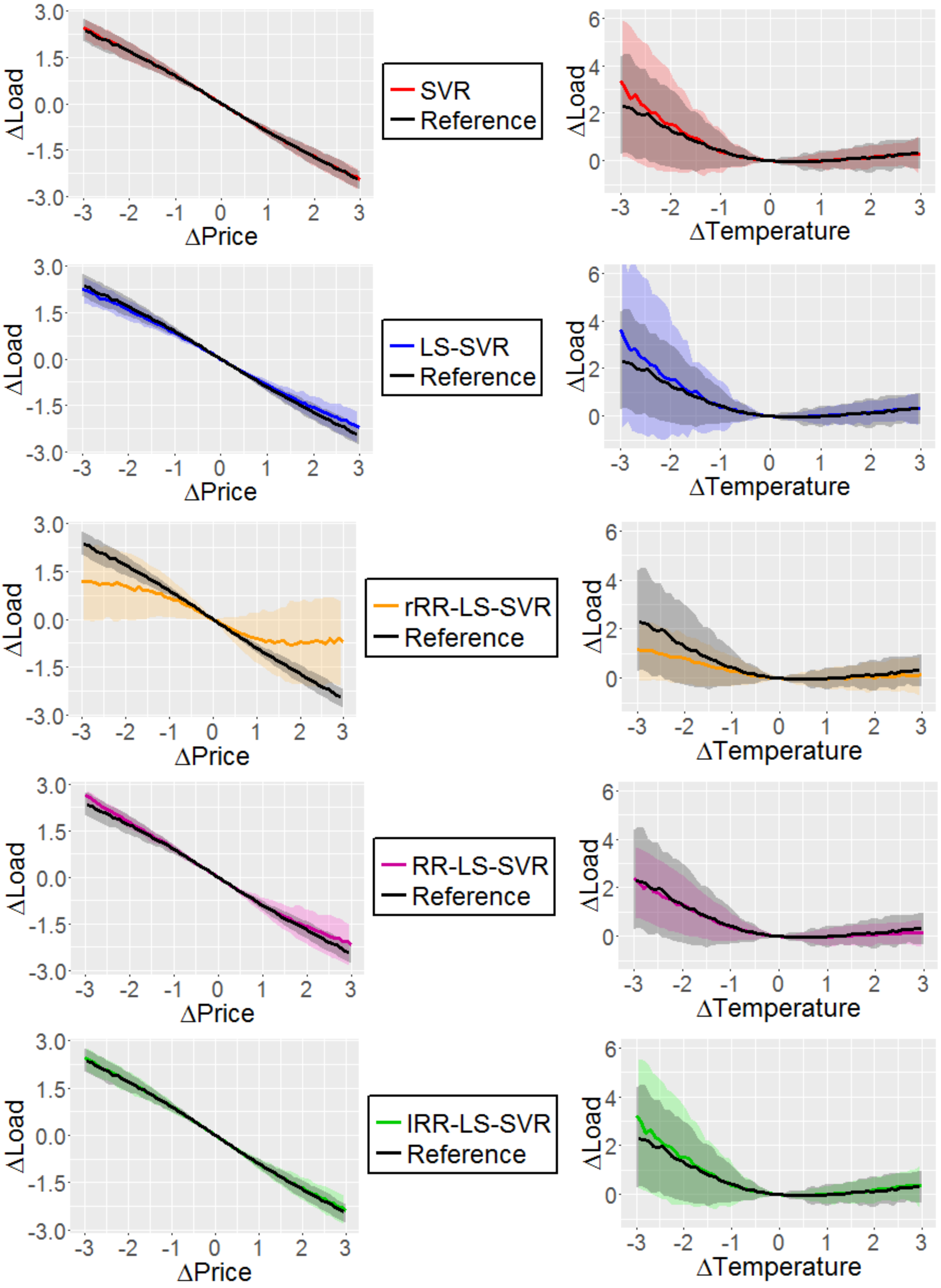

**Figure 4.2:** Price (left column) and temperature (right column) dependency extraction through SVR (red), LS-SVR (blue), rRR-LS-SVR (orange), RR-LS-SVR (purple) and IRR-LS-SVR (green) towards the real dependency (black) in terms of mean (solid line) and standard deviation (coloured area).

## 4.4 Real dataset

Once verified all the algorithms by means of the toy model, they are applied to the real dataset. As for the previous analysis on the artificial data, at first, the forecasting results are discussed and secondly the non-linear extraction is presented.

### 4.4.1 Forecasting results

In total, 7 algorithms have been implemented and tested on each of the load groups and on the total load: all the results are displayed in Table 4.2. For the two benchmark models, only the RMSE is provided since these two algorithms are meant as reference for the error evaluation. Starting from the benchmark models, the testing on the first set of observations shows an RMSE of around 0.1 of standard deviation, achieving already a good level of accuracy even if using a very simple and naive approach. It is important to remember that the forecasts are made with a lead time of 5 minutes ahead and, on this time span, the aggregated electrical load can be considered a quasi-stationary process. Figure 4.3 provides an example of the five load groups (excluding the total load) on a time window of an hour, where it is possible to notice how the process is quite stable and smooth without big drops or spikes. For this reason, the persistence method works properly and the auto-regression shows an even better RMSE on all the groups. The latter was expected to achieve more accurate results; however, in particular with auto-regressive algorithms, it is important to adjust the model online or to re-train it regularly since the relationship between the electrical load and its previous values changes in time. For this benchmark approach, it has been decided to employ only a simple version of the method, without adding a time dependency or any other exogenous variable; hence, the RMSE is not remarkably lower than the one of the persistence method.

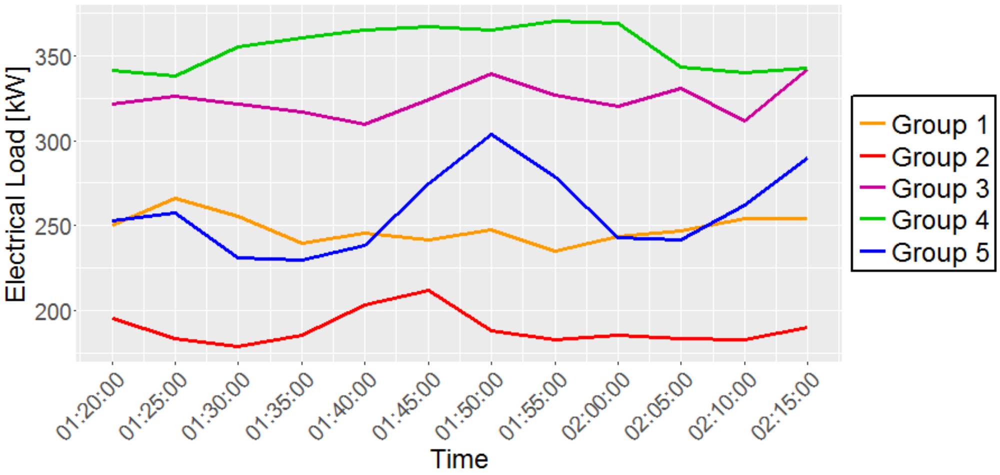

**Figure 4.3:** Trends of the 5 load groups of one hour (12 time stamps) of the 10th of December 2014

| Model | | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Total load |
|---|---|---|---|---|---|---|---|
| Persistence | Testing RMSE | 0.1148 | 0.1129 | 0.1331 | 0.1036 | 0.1517 | 0.0670 |
| Auto-Regression | Testing RMSE | 0.1058 | 0.1060 | 0.1108 | 0.0844 | 0.1183 | 0.0594 |
| SVR | Training Time [s] | 2410 | 2537 | 2312 | 2290 | 2345 | 2239 |
| | Support Vectors | 135 | 125 | 133 | 124 | 128 | 104 |
| | Testing RMSE | 0.0410 | 0.0483 | 0.0430 | 0.0419 | 0.0418 | 0.0485 |
| LS-SVR | Training Time [s] | 541 | 541 | 542 | 541 | 541 | 541 |
| | Support Vectors | 4500 | 4500 | 4500 | 4500 | 4500 | 4500 |
| | Testing RMSE | 0.0040 | 0.0057 | 0.0043 | 0.0039 | 0.0045 | 0.0035 |
| rRR-LS-SVR | Training Time [s] | 541 | 541 | 542 | 541 | 541 | 541 |
| | Support Vectors | 100 | 100 | 100 | 100 | 100 | 100 |
| | Testing RMSE | 0.0513 | 0.0611 | 0.0532 | 0.0565 | 0.0526 | 0.0452 |
| RR-LS-SVR | Training Time [s] | 66 | 67 | 66 | 67 | 66 | 68 |
| | Support Vectors | 100 | 100 | 100 | 100 | 100 | 100 |
| | Testing RMSE | 0.0513 | 0.0711 | 0.0505 | 0.0532 | 0.0484 | 0.0480 |
| IRR-LS-SVR | Training Time [s] | 18917 | 18492 | 19040 | 18701 | 18006 | 18052 |
| | Support Vectors | 61 | 63 | 55 | 51 | 52 | 55 |
| | Testing RMSE | 0.0288 | 0.0288 | 0.0313 | 0.0300 | 0.0314 | 0.0252 |

**Table 4.2:** Overview of the results from the training routine on the real dataset

The first algorithm applied to the real dataset is the SVR; employing around 125 support vectors out of 4500 selected observations, the method shows a good accuracy decreasing around 3 times the benchmark RMSE. What is also interesting to notice is the training time, even if not meaningful in a real application where the training is mainly done offline. If compared to the toy model results, the computational time is of the same order of magnitude, considering that the process analysed is extremely more complicated: using 2 or 32 input features does not significantly increase the computational time. This can be explained by analysing again the structure of the SVR algorithm (basis for all the formulations used in this study). No computation is strongly dependent on the number of input features, that always appear as dot products, which do not take a significantly higher time to be done over 32-dimensional vectors rather than over 2-dimensional ones. On the contrary, the computational cost depends on the number of observations used for the training routine. Hence, the fact that both the toy model and the real-case models use 4500 training observations explains the similar training time. This peculiarity of the SVR-based algorithms underlines one more time their suitability and efficiency on high dimensional datasets. It is possible to include more features, i.e. increasing the level of information brought to the model, and, at the same time, limit the training time.

As for the LS-SVR algorithm, the same considerations done for the toy model still hold: the method shows a great level of accuracy (below 0.01 for all the load groups) and a faster training time compared to the SVR. However, these very low errors might be a sign overfitting: for this reason, a further testing on the second half of the dataset is carried out later in this study. Figure 4.4 displays the value of the coefficients $\boldsymbol{\alpha}$, weighting the 4500 support vectors: keeping in mind that the chromatic scale is logarithmic with the absolute values of the coefficients, it is possible to spot that there is room for sparsity improvements. Hence, in order to unburden the model, the rRR-LS-SVR and RR-LS-SVR methods have been tested, setting the number of support vectors to 100.

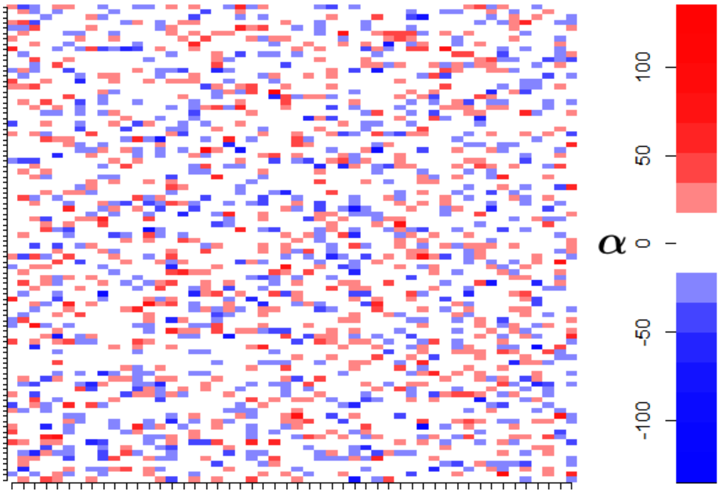

**Figure 4.4:** Graphical representation of the support vectors weights, $\boldsymbol{\alpha}$, of the LS-SVR algorithm for the real dataset

Both of them performed better on a large-scale dataset rather than on the toy model leading to an accuracy comparable to the SVR algorithm: Figure 4.4, in fact, displays more white spots, i.e. more negligible coefficients, than Figure 4.1. In particular, the RR-LS-SVR has slightly better results and a faster training routine confirming its expected good trade-off between speed and accuracy.

As previously discussed, the IRR-LS-SVR method showed a great potential when applied to the toy model: when tested on the real case database, the results are even better, underlining its efficiency on large-scale data analysis. In fact, a substantial reduction of the number of support vectors (around 60 employed for each group) does not affect accuracy: the testing RMSE present values even lower than the SVR ones. The level of accuracy of the LS-SVR is unreachable, the IRR-LS-SVR is indeed an approximation of it, but this method is more suitable for online applications. The custom choice of the number of support vectors makes this algorithm extremely portable: it allows, in fact, the users to build several models using the same method but with different levels of accuracy, which might be needed for different applications. The training time of the IRR-LS-SVR largely increases on the real data; however, the same considerations presented for the toy model on the time optimisation of this algorithm and on the solver employed are still valid.

All the 7 methods, as aforementioned, have also been tested on a different subset of data to check their robustness; furthermore, two procedures of online adjustment of their coefficients have been carried out. The results in Table 4.3 show the final RMSE for each algorithm and each load group with no adaptation of the coefficients, with the weights varying method (WV) and with the Gaussian Process Kernel one (GPK). All the algorithms, but the LS-SVR, employ the formal gradient descent approach to recursively adapt their coefficients; as for the LS-SVR, the gradient descent method is simplified by using a fixed coefficient, tuned with a try and error approach, rather than the inverse of the Hessian. When no adaptation is implemented, all the methods decrease drastically their accuracy: this is sign of both a small overfitting during the training procedure and of the non-stationarity of the input-output dependencies over time. However, when recursively adapting the coefficients, the forecasting error settles again to the order of magnitude of the training RMSE proving the essential and effective role of online adaptation of the models. Figure 4.5 shows the cumulative RMSE for the total load (the trends of the single load groups have a similar behaviour) during the testing routine. The RMSE spike that all graphs show in the first time stamps underlines the need for an adaptive algorithm: in fact, for non-stationary processes, a single batch training is not sufficient as the relationships among the different variables can change over time. When looking at the adaptive models, after the initial spike, the error is almost immediately brought to values around the training ones maintaining the accuracy of the forecasts. GPK algorithm is found to perform better, especially in the case of LS-SVR. For all the other algorithms, both models adapt in a fast way keeping the RMSE lower from the very beginning with no considerable differences in its final value.

| Model | | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Total load |
|---|---|---|---|---|---|---|---|
| SVR | No Adaptation | 0.1242 | 0.1370 | 0.1489 | 0.1275 | 0.1643 | 0.1354 |
| | Weights Varying | 0.0538 | 0.0492 | 0.0599 | 0.0569 | 0.0741 | 0.0417 |
| | Gaussian Process Kernel | 0.0566 | 0.0462 | 0.0622 | 0.0624 | 0.0713 | 0.0441 |
| LS-SVR | No Adaptation | 0.0513 | 0.0403 | 0.0634 | 0.0555 | 0.0717 | 0.0545 |
| | Weights Varying | 0.0350 | 0.0313 | 0.0456 | 0.0390 | 0.0442 | 0.0353 |
| | Gaussian Process Kernel | 0.0192 | 0.0199 | 0.0207 | 0.0187 | 0.0284 | 0.0173 |
| rRR-LS-SVR | No Adaptation | 0.1760 | 0.1571 | 0.2043 | 0.1659 | 0.2519 | 0.1574 |
| | Weights Varying | 0.0591 | 0.0615 | 0.0707 | 0.0709 | 0.0794 | 0.0594 |
| | Gaussian Process Kernel | 0.0574 | 0.0574 | 0.0675 | 0.0684 | 0.0764 | 0.0580 |
| RR-LS-SVR | No Adaptation | 0.2189 | 0.1651 | 0.2172 | 0.1658 | 0.2239 | 0.1807 |
| | Weights Varying | 0.0617 | 0.0785 | 0.0764 | 0.0854 | 0.0790 | 0.0560 |
| | Gaussian Process Kernel | 0.0592 | 0.0744 | 0.0717 | 0.0831 | 0.0764 | 0.0548 |
| IRR-LS-SVR | No Adaptation | 0.1485 | 0.1150 | 0.1435 | 0.1527 | 0.1702 | 0.1402 |
| | Weights Varying | 0.0440 | 0.0400 | 0.0575 | 0.0394 | 0.0624 | 0.0324 |
| | Gaussian Process Kernel | 0.0425 | 0.0372 | 0.0552 | 0.0385 | 0.0611 | 0.0313 |

**Table 4.3:** RMSE results from the testing routine with coefficients adaptation methods on the real dataset

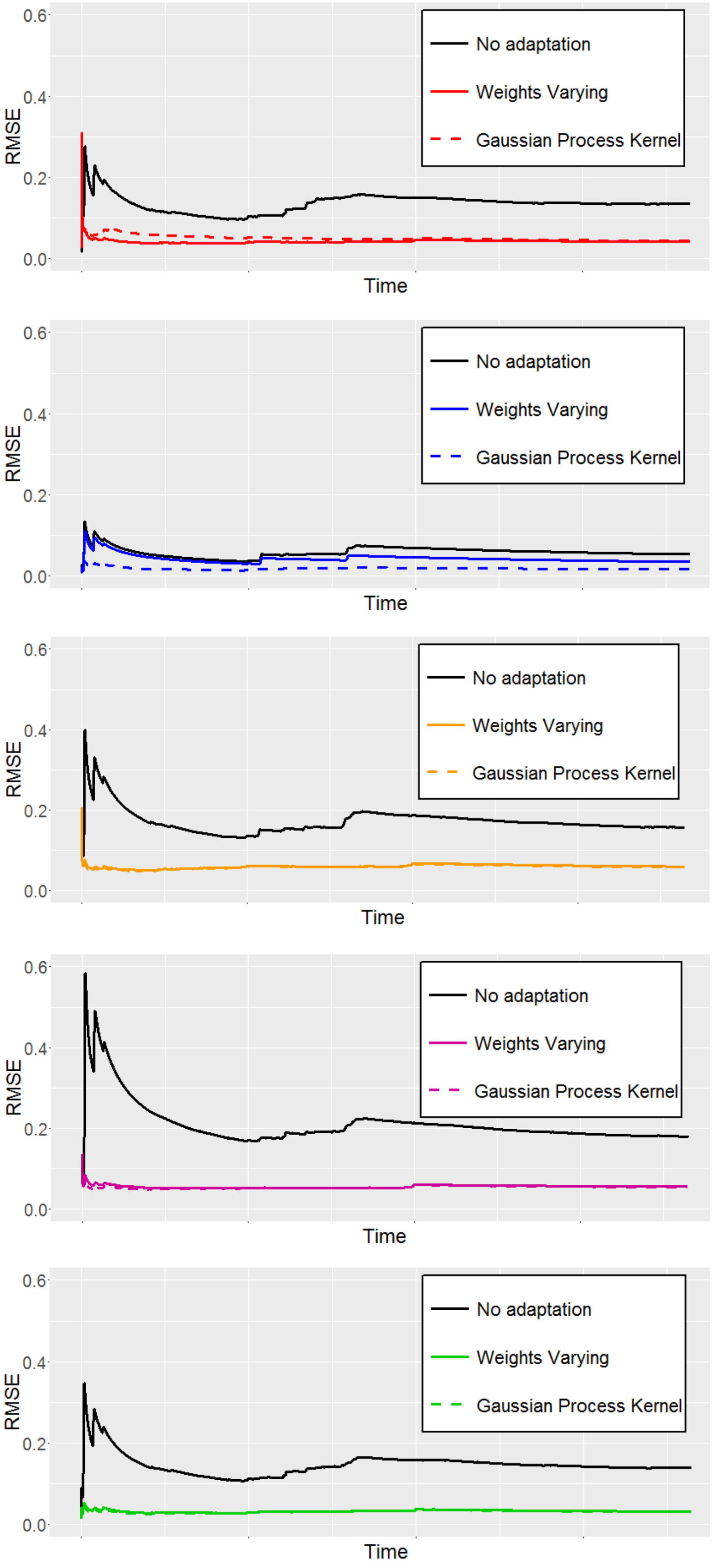

**Figure 4.5:** Cumulative RMSE of the total load forecasts with online recursive adaptation through WV (solid line) and GPK (dashed line) for SVR (red), LS-SVR (blue), RR-LS-SVR (orange), rRR-LS-SVR (purple) and IRR-LS-SVR (green) towards no adaptation testing (black)

Some interesting considerations can be extracted from the results of the coefficient adaptation of the IRR-LS-SVR method. The testing results with no adaptation still present an RMSE of the same order of the SVR one: hence, this approximated method proved to have the same robustness of a more “exact” algorithm. Furthermore, among all the models developed, the IRR-LS-SVR combined with the coefficients adaptation is the one with the lowest relative worsening of performance: even if the final RMSE is still higher than the LS-SVR one, its relative increase is considerably lower. Considering the total aggregated load with a GPK adaptation of the model coefficients as example, the LS-SVR achieves a testing RMSE around 4 times higher in the second half of the dataset, while the IRR-LS-SVR worsen its performance only of around 25%.

Another interesting feature of the IRR-LS-SVR algorithm is that its recursive structure could allow not only adapting its coefficients online but also adding new support vectors on the fly. The idea is simply to compute the ranking function (the difference of the Lagrangian of the objective function, $\vartheta$) for each new observation. This new input will be added to the model only if the carried information is considerable, e.g. if the ranking function value is higher than a tuned threshold. All these features make IRR-LS-SVR the most interesting algorithm among all the ones investigated in this study.

A state-of-the-art implementation of methods for Short Term Load Forecasting reports a Mean Absolute Percentage Error (MAPE) of around 0.1% for 5 minutes lead time forecasts and a Mean Absolute Error (MAE) of around 15 MW, [40]. In order to compare these results to the ones achieved in this study, the MAPE and MAE have been calculated for the total load group as follows:

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{n} |\frac{y_i^{fore} - y_i^{real}}{y_i^{real}}|\dot{1}00$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{n} |y_i^{fore} - y_i^{real}|$$

As displayed in Table 4.4, apart from the LS-SVR whose great accuracy has been already pointed out, all the methods show a MAPE around 10 times higher. However, it has to be considered that the MAE reported is expressed in kilowatts, and not in megawatts like in [40]; hence, the size of the investigated load is around 1000 times smaller. It is well known that the MAPE, being a percentage calculation, is most likely to decrease along with the size of the quantity predicted: therefore, it is possible to state that the algorithms employed in this study present performances comparable to the ones found in state-of-the-art literature.

| Model | Training dataset | | Testing dataset | |
|---|---|---|---|---|
| | MAPE [%] | MAE [kW] | MAPE [%] | MAE [kW] |
| SVR | 1.96 | 21.31 | 0.84 | 16.26 |
| LS-SVR | 0.05 | 0.57 | 0.27 | 5.41 |
| rRR-LS-SVR | 1.49 | 16.61 | 1.13 | 21.76 |
| RR-LS-SVR | 1.63 | 18.18 | 1.09 | 21.07 |
| IRR-LS-SVR | 0.83 | 9.16 | 0.55 | 10.82 |

**Table 4.4:** MAPE and MAE results on the total aggregated load for all methods

### 4.4.2 Non-linear dependencies

For the sake of this study, only the dependencies on price and temperature have been investigated, since all the load control systems installed in Bornholm houses were based on the real-time price and acted mainly on electric heating or heat pumps. The non-linear extraction of the load dependency on price and temperature for a large and complex dataset is obviously not as easy as for the toy model. Cross-dependencies with other features and non-stationary processes can mine an accurate approximation of the sought relationships.

The same approach used with the toy model has been applied to the real dataset, but the results obtained were not as accurate as expected. As for, the extracted dependencies on temperature, displayed in Figure 4.6, the SVR model is the one presenting the most consistent relationship detecting a load increase in case of lower temperatures. A possible load increase for high temperatures is not the case for the island of Bornholm where the need of electrical cooling is almost absent. The consumption, in fact, is even decreasing for temperatures higher than the average: this could be easily explained by considering that the average temperature in Bornholm recorded in this dataset is around 5°C. As for the extraction of the price dependency, which is the most useful in view of a price driven Demand Response architecture, the relationship extracted through the method employed in this study is neither clear nor consistent among the different models, as displayed in Figure 4.7. The extraction via the LS-SVR algorithm shows an almost null load change with any variation of the real-time price, while the SVR method has a consistent trend among all groups but without detecting different behaviours among the different load control systems. The IRR-LS-SVR model, instead, provides the expected trend for Group 1, where no variation was expected, for Group 4 and for the total load. As for Group 2, the manual controlled one, the extracted relationship could be explained by the willingness of the consumers to decrease their consumption in case of high prices. However, the author expected to isolate a more meaningful trend also for Group 3 and Group 5.

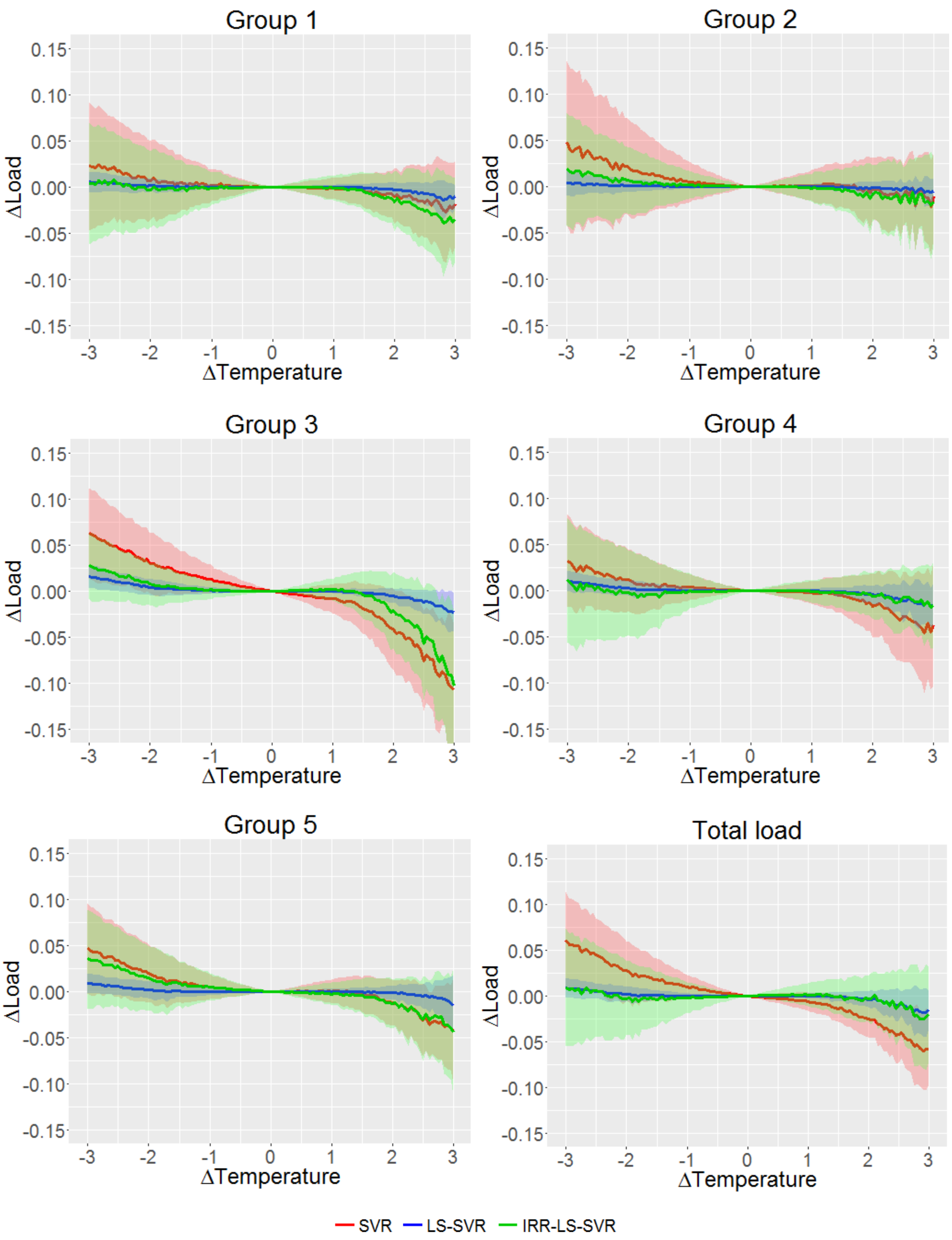

**Figure 4.6:** Temperature dependency extraction through SVR (red), LS-SVR (blue) and IRR-LS-SVR (green) for the 5 load groups and the total load

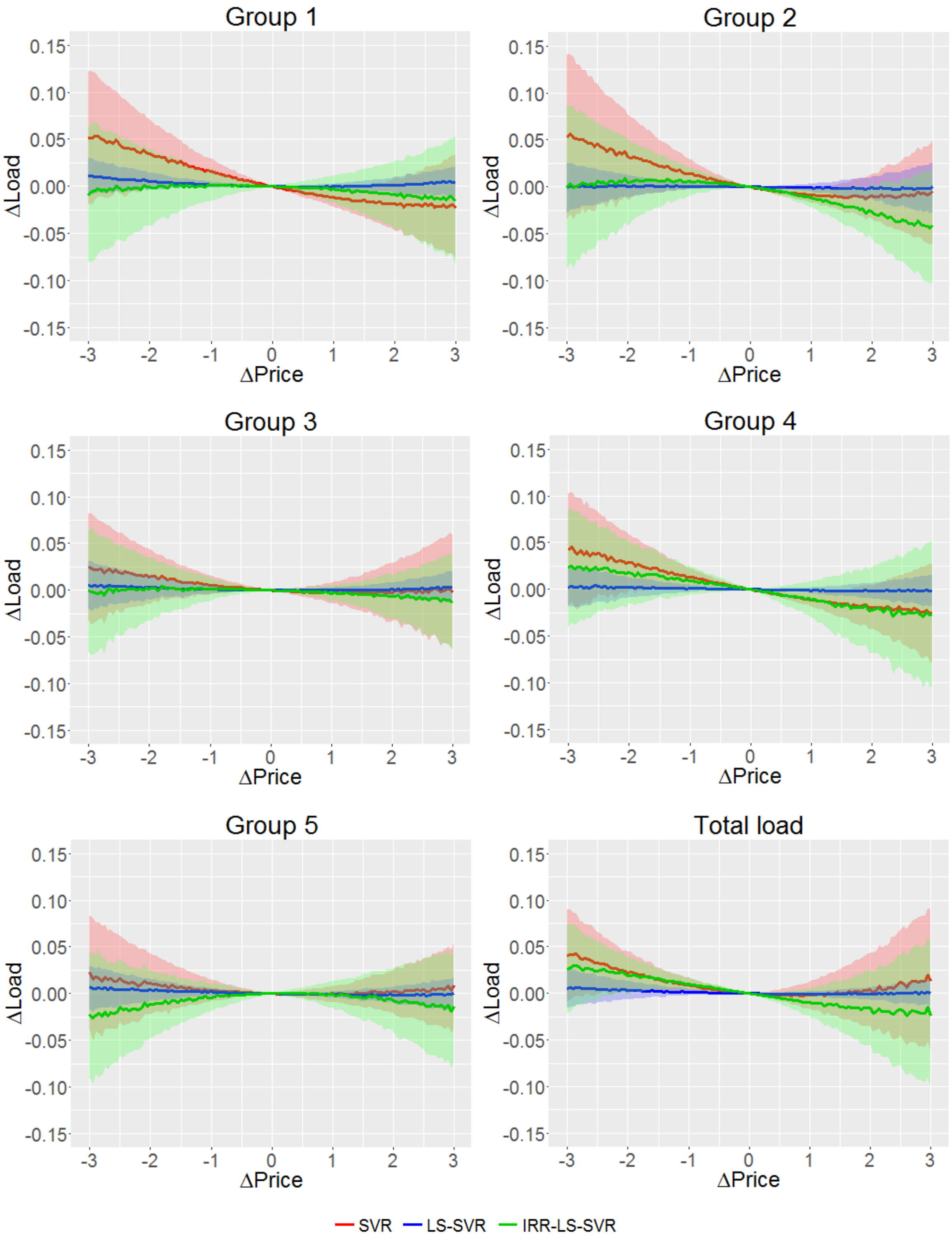

**Figure 4.7:** Price dependency extraction through SVR (red), LS-SVR (blue) and IRR-LS-SVR (green) for the 5 load groups and the total load

A possible explanation of this ineffective dependency extraction could derive, as already mentioned, from the strong non-linearity and non-additivity of the behaviour of the electrical load towards the real-time price. Another possible strategy to detect this relationship would be to decrease the number of input features employed for the training of the models. In this way, it would be possible to get rid of all the cross dependencies with other quantities, but this will lead to a lower accuracy of the forecasts. Further research on this issue has to be carried out in order to extract a more general relationship. However, what is needed to a Demand Response management system is to forecast the availability of the flexible assets at a specific time instant. Therefore, these algorithms could be used "locally" on time to predict the load variation after a change of the real-time price under the assumption that all the other input features will not change.

Another possible reason that could explain the extraction of an unclear relationship between electrical load and price is the lead time considered. Forecasting the load only 5 minutes ahead in time could not be enough to detect a response of the loads towards a variation of the real-time price. It has been estimated that, for example, the heat pumps of Group 3 reacted completely only after 20 minutes because of their inner operational cycles, [21]. Therefore, training the models on several lead times could allow investigating the load dependency on price dynamically in time and extracting, if possible, a more meaningful behaviour.

CHAPTER 5

# Discussions and conclusion

The goal of this study, as stated in the introduction, is to analyse data from a real implementation of a large-scale Demand Response system in order to develop a robust and accurate forecasting model, also able to provide knowledge about the dependency of the electrical load on the different inputs. The test case of Bornholm during EcoGrid EU project granted the author the chance to work with one of the biggest and most advanced database in the field of Demand Response. Almost 2000 household consumers were involved with 4 different load control systems acting on a real-time price market platform.

Following up on EcoGrid EU recommendations, a state-of-the-art forecasting model has been investigated: Support Vector Machine based algorithms have been chosen as statistical learning tools for this application. The main reason for this choice lies on the special feature of these methods to automatically minimise the complexity of the trained model by exploiting only a few observations (i.e. the support vectors) to fit large data in a robust manner. Moreover, recent research has led to lighter formulations of the SVM algorithm without significantly worsening its accuracy: hence, in view of an online application, some of these methods have been described, implemented and tested. All the algorithms investigated in this study have been coded by means of the statistical software R: only the basic formulation of the SVR was founded in the repository of R libraries while all the reformulations have been fully coded by the author. For this reason, a toy model has been developed in order to validate them. This artificial dataset has been built as a rough estimation of the electrical load and its non-linear dependency on temperature and price. The results of this first testing proved that the algorithms worked with a high level of accuracy both in terms of forecasting and extraction of the dependency on the single inputs. Once the algorithms were successfully validated, the real case dataset has been analysed and all the methods have been tested on the 5 different load groups and on the total system load. These groups are the result of the aggregation of the households according to the different load control system installed: i.e. no control, manual control, two different architecture on electric heating appliances and one control system on heat pumps. The simulations have been structured miming a real application case: a first tuning and validation routine is carried out on the first half of the dataset, while a pure testing is performed on the second part. Given that the process investigated is not stationary in time, an online adaptation of the models has been taken into account in order to recursively adapt the methods to the new data available.

The results show a great accuracy it terms of forecasting, if compared to the persistence and auto-regression benchmark models, in particular for the LS-SVR method. The forecasting errors have been successfully compared to a state-of-the-art model employed to predict the electrical load with 5 minutes lead time. In particular, the Improved Reduced Recursive Least Squares Support Vector Regression (IRR-LS-SVR) has been found to provide the best balance between computational burden and accuracy. Its recursive approach allows selecting the observation carrying the largest information as a new support vector and customising the sought accuracy and operational time, both increasing along with the number of support vectors allowed. It has to be pointed out that the training time computed for this method results around 8 times higher than the SVR; however, the batch training is done offline and with low frequency, while the online use of the method is faster if compared to all the other algorithms investigated in this study. Moreover, the implementation of this method was made by the author and no optimised library has been used; hence, there is room for a wide improvement on the training time.

The extraction of non-linear dependencies of the electrical load, in particular on the real-time price, was hard to reconstruct given the numerous cross-dependencies and the strong non-linearity of the inputs. Thus, the relationship between one feature and the output was difficult to isolate: if some meaningful results for the temperature were derived, the dependency on price was almost imperceptible. However, this does not mean neither that the models are wrong, nor that this dependency cannot be extracted *ad hoc* with a specific set of input features. It is important to consider that also the lead time of the forecast influences the ability to detect and extract a meaningful relationship between the load and the input features.

## 5.1 Future work

Due to the highly time-consuming implementations from scratch of the algorithms and of all the simulations run for this study, and considering the limited time, some of the initial ideas are left as future work to the readers or to the author himself. Starting from the most straightforward step, several models could be trained on different lead times: in this way, a dynamic response of the system could be predicted. Furthermore, assuming to have trained all these models and, therefore, be able to predict with good accuracy the flexibility of the responsive loads, the derivation of the optimal price signal to achieve a desired Demand Response could take into account also the dynamics of this flexible assets. In particular, one would like to find a balance between the speed of the response and the rebound effects, e.g. overshoots in the dynamic response.

Another interesting analysis for this dataset could be to perform some clustering algorithms in order to aggregate the single households consumptions according to specific criteria: e.g. based on their dynamic responses or on their actual participation to the flexibility of the system. Furthermore, a clustering procedure could also be carried out in order to identify a set of households that are representative of a bigger group. In this case, a lower number of houses will need to be smart metered, with consequent lower costs, but the total information could be preserved. A simple and naive approach to this last problem was tried during this study by applying the RR-LS-SVR (the fastest method investigated) to several random subsets of households. The error obtained was around 10 times higher than the one computed with the aggregated load. However, this approach is far from being optimal since it relies on several simulations with random clusters, i.e. no real clustering procedure was implemented. By further optimising the training time of the IRR-LS-SVR, it would be possible to exploit its iterative potential in order to select the most meaningful support vectors both in terms of observations and of subset of houses.

Addressing the scalability of this study to an ideal power system with load control automation implemented on a large scale, the enormous amount of data will not likely be processed in a core computational center. Hence, a distributed learning architecture will have to be developed with multiple stages learning algorithms in order to pass at each stage the most meaningful information. This could be carried out, for example, at some grid nodes where the power flowing to a group of users is known, hence, avoiding the smart metering of all houses.

## 5.2 Further considerations

The scalability of Demand Response to the whole power system implies not only technical evolution but also social and economic ones. From a technical point of view, on top of broader and more advanced intelligent management systems, other sources of flexibility besides electric heating or heat pumps could be involved. In particular, the so-called “prosumers” equipped with the proper Energy Management System (EMS) could be able to increase the flexible assets of a power system by considering their local generation to supply the needed power. If combined with some storage systems, or considering their thermal inertia as a storage system, each consumer could potentially act like a small lung for the system, shifting load in time or supplying to the grid both directly generated or stored power. Technical feasibility, however, is not enough: a widespread social interest on this issue should be deeper examined; quoting from a final project overview of EcoGrid EU [21], “a wider implementation of the EcoGrid EU concept depends on the degree of smart grid readiness among the customers”. During the recruitment of participants for EcoGrid EU, it was chosen to stress the environmental goal of the project, aiming to increase the integration of renewable generation, rather than an economic profit. A customer survey at the end of the project confirmed that a large part of the users involved committed to the project because of its environmental aspect. However, it has to be pointed out that the consumers were granted not to suffer any economic loss, somehow biassing their motivation, and that Bornholm population has always been aware of the environmental issues to achieve a fossil free society.

Even if Larsen in [11] extracted a cost reduction of €2/year (on data related to EcoGrid EU project) at the customer level, it is noteworthy to mention that neither a subsidy scheme nor a remuneration for grid services are taken into account. A large employment of DR systems could provide notable benefits to the power system, and users could perceive a consequent profit, e.g. through an incentive policy subsidising the responsive loads. In any case, the need for a regulation scheme is urgent, technically- and economically-wise. On one hand, in order to grant the stability of this complex and distributed system, studies like the one of Roozbehani on the dynamics of power grids subjected to a real-time market, [36], should be further developed. Furthermore, network standards and common communication protocols have to be defined and implemented at least at European level, encouraging the development of Demand Response systems without facing contingent barriers. On the other hand, the lack of a proper regulation and of a market structure suitable to integrate the Demand Response solutions are two of the challenges suggested in [8]: recently, two projects have been started on these research topics, i.e. EcoGrid EU 2.0 [41] and Flex4RES [42]. The former will investigate the best market structure to integrate the responsive loads, in particular whether, among all, a price-based Demand Response is the optimal strategy for every service that the flexible consumption could offer. The latter is focussed on investigating the proper regulation policies to implement, in particular in the Nordic countries, in order to achieve the ambitious sustainable targets set for 2050, taking into account also the demand flexibility.

# Bibliography

[1] United Nations. “Adoption of the Paris Agreement”. In: *Conference of the Parties on its twenty-first session* 21932.December (2015), page 32. URL: http://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf.

[2] Energinet. *Wind power penetration in Denmark*. 2016. URL: http://energinet.dk/EN/El/Nyheder/Sider/Dansk-vindstroem-slaar-igen-rekord-42-procent.aspx.

[3] IRENA. *RENEWABLE CAPACITY STATISTICS 2016*. Technical report. 2016. URL: www.irena.org/Publications.

[4] European Union. *Smart Grid*. 2016. URL: http://www.smartgrids.eu/ (visited on February 1, 2016).

[5] European Parliament. *DIRECTIVE 2012/27/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL*. 2012.

[6] Energinet and Dansk Energi. *Smart Grid in Denmark*. Technical report. 2013.

[7] Thomas G. Dietterich. “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees”. In: *Machine Learning* 40 (2000), pages 139–157. ISSN: 0885-6125. DOI: 10.1023/A:1007607513941.

[8] Niamh O’Connell et al. “Benefits and challenges of electrical demand response: A critical review”. In: *Renewable and Sustainable Energy Reviews* 39 (2014), pages 686–699. ISSN: 13640321. DOI: 10.1016/j.rser.2014.07.098.

[9] Saeed Rahmani Dabbagh and Mohammad Kazem Sheikh-El-Eslami. “Participation of demand response resources through virtual power plant: A decision framework under uncertainty”. In: *2015 IEEE 15th International Conference on Environment and Electrical Engineering, EEEIC 2015 - Conference Proceedings* (2015), pages 2045–2049. DOI: 10.1109/EEEIC.2015.7165490.

[10] Olivier Corradi et al. “Controlling electricity consumption by forecasting its response to varying prices”. In: *IEEE Transactions on Power Systems* 28.1 (2013), pages 421–429. ISSN: 08858950. DOI: 10.1109/TPWRS.2012.2197027.

[11] Emil M. Larsen. “Demand response in a market environment”. In: (2015).

[12] Carlos Gorria et al. “Forecasting flexibility in electricity demand with price/-consumption volume signals”. In: *Electric Power Systems Research* 95 (2013), pages 200–205. ISSN: 03787796. DOI: 10.1016/j.epsr.2012.09.011. URL: http://www.sciencedirect.com/science/article/pii/S0378779612002933.

[13] Gabriele Kotsis et al. “Demand Aggregator Flexibility Forecast: Price Incentives Sensitivity Assessment IEEE Procs. 2015 12h I (in press).” In: *12th International Conference on the European Energy Market (EEM)*. 2015. ISBN: 9781467366922.

[14] SEDC - Smart Energy Demand Coalition and Mapping. *Mapping Demand Response in Europe Today Mapping Demand Response in Europe Today*. Technical report. SEDC - Smart Energy Demand Coalition Mapping, 2015.

[15] Donald J. Hammerstrom et al. “Pacific Northwest GridWise(TM) Testbed Demonstration Projects Part I . Olympic Peninsula Project”. In: *Contract* (2007), page 157. ISSN: 00179078. DOI: `10.2172/926122`. URL: `http://www.gridwise.pnl.gov/docs/op{\_}project{\_}final{\_}report{\_}pnnl17167.pdf`.

[16] Eindhoven University of Technology. *E-Price*. URL: `http://www.e-price-project.eu/website/TLL/eprice.php?q=1` (visited on June 1, 2016).

[17] European Commission. *EU Sustainable Energy Awards*. URL: `http://eusew.eu/2016-awards-winners` (visited on June 1, 2016).

[18] European Union. *EcoGrid EU - A prototype for European Smart Grids*. 2015. URL: `http://www.eu-ecogrid.net/` (visited on June 20, 2002).

[19] *PowerLab DK*. URL: `http://www.powerlab.dk/` (visited on January 1, 2016).

[20] Per Lund. *EcoGrid EU – A Prototype for European Smart Grids*. Technical report. Energinet.dk, 2016. URL: `http://www.eu-ecogrid.net/documents-and-downloads`.

[21] Ove S. Grande. *EcoGrid EU : From Implementation to Demonstration*. Technical report October. SINTEF, 2011. URL: `http://www.eu-ecogrid.net/documents-and-downloads`.

[22] PowerLabDK. *Bornholms elsystem*. URL: `http://bornholm.powerlab.dk/` (visited on June 13, 2016).

[23] Pierre Pinson, Emil M. Larsen, and Guillaume Le Ray. “EcoGrid EU evaluation : The concept and market place”. In: 2015.September (2015), pages 1–14.

[24] Association for Computational Learning. *“Learning Has Just Started” – an interview with Prof. Vladimir Vapnik*. 2014. URL: `http://www.learningtheory.org/learning-has-just-started-an-interview-with-prof-vladimir-vapnik/` (visited on February 1, 2016).

[25] Xiangying Wang and Yixin Zhong. “Statistical learning theory and state of the art in SVM”. In: *The Second IEEE International Conference on Cognitive Informatics, 2003. Proceedings.* 2 (2003), pages 55–59. DOI: `10.1109/COGINF.2003.1225953`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1225953`.

[26] Daniela Witten et al. *An Introduction to Statistical Learning*. 2013. ISBN: 978146147 1370. URL: `http://www-bcf.usc.edu/{~}gareth/ISL/`.

[27] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning The Elements of Statistical LearningData Mining, Inference, and Prediction, Second Edition.* 2009, page 282. ISBN: 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7. URL: http://www.worldcat.org/oclc/405547558$\backslash$nHastie,Tibshiranietal-Theelementsofstatisticallearning.pdf$\backslash$nhttp://www.springer.com.libproxy1.nus.edu.sg/statistics/statistical+theory+and+methods/book/978-0-387-84857-0$\backslash$nhttp://statweb.stanford.edu/{~}tibs/E.

[28] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pages 273–297. ISSN: 0885-6125. DOI: 10.1007/BF00994018. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/BF00994018.

[29] Alex J. Smola and Bernhard Schölkopf. "A Tutorial on Support Vector Regression". In: *Statistics and Computing* 14.3 (2004), pages 199–222. ISSN: 09603174. DOI: 10.1023/B:STCO.0000035301.49549.88.

[30] Yaser Abu-Mostafa. *Machine Learning Course.* 2016. URL: http://work.caltech.edu/lectures.html (visited on February 1, 2016).

[31] Karl Pearson. "On lines and planes of closest fit to systems of points in space." In: (1901), pages 559–572.

[32] John C. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization.* Technical report. Redmond - USA: Microsoft, 2014.

[33] Yongping Zhao and Jianguo Sun. "Recursive reduced least squares support vector regression". In: 42 (2009). DOI: 10.1016/j.patcog.2008.09.028.

[34] Yong-ping Zhao et al. "An improved recursive reduced least squares support vector regression". In: *Neurocomputing* 87 (2012), pages 1–9. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2012.01.015. URL: http://dx.doi.org/10.1016/j.neucom.2012.01.015.

[35] Li Guo Sun et al. "A novel adaptive kernel method with kernel centers determined by a support vector regression approach". In: (2012).

[36] Mardavij Roozbehani, Munther A Dahleh, and Sanjoy K Mitter. "Volatility of Power Grids under Real-Time Pricing". In: *IEEE TRANSACTIONS ON POWER SYSTEMS* (2012).

[37] Tao Hong and Pu Wang. "Fuzzy interaction regression for short term load forecasting". In: *Fuzzy Optimization and Decision Making* 13.1 (2014), pages 91–103. ISSN: 15732908. DOI: 10.1007/s10700-013-9166-9.

[38] Jonathan R. M. Hosking et al. "Short-term forecasting of the daily load curve Smart Grid". In: April (2013). DOI: 10.1002/asmb.1987.

[39] Chih-chung Chang and Chih-jen Lin. *LIBSVM – A Library for Support Vector Machines.* URL: https://www.csie.ntu.edu.tw/{~}cjlin/libsvm/ (visited on June 1, 2016).

[40] Che Guan Che Guan et al. “Very short-term load forecasting: Multilevel wavelet neural networks with data pre-filtering”. In: *2009 IEEE Power & Energy Society General Meeting* 28.1 (2009), pages 30–41. ISSN: 1944-9925. DOI: 10.1109/PES.2009.5275296.

[41] European Union. *EcoGrid EU 2.0*. 2016. URL: http://www.eu-ecogrid.net/rss-feed/78-ecogrid-is-continuing (visited on June 20, 2002).

[42] DTU. *Flex4RES*. URL: http://www.sys.man.dtu.dk/Research/EER/Research-projects/Flex4RES (visited on June 1, 2016).

√17
e^{iπ} = −1
{2.7182818284}
ϑφεϱτυθιοπσδφγηξκ