Faculdade de Engenharia da Universidade do Porto

Load forecast on a Micro Grid level through Machine Learning algorithms

Tiago Alexandre Castro Guimarães

Dissertation submitted in partial fulfilment of the requirements for the Degree of Master of Science in Electrical and Computers Engineering

Supervisor: Prof. Doctor Helder Filipe Duarte Leite
Co-supervisor: Eng. Luís Filipe Azevedo

March 2020

Resumo

As Micro Redes constituem um sector em crescimento da indústria energética, representando uma mudança de paradigma, desde as remotas centrais de geração até à produção mais localizada e distribuída. A capacidade de isolamento das principais redes elétricas e atuar de forma independente tornam as Micro Redes em sistemas resilientes, capazes de conduzir operações flexíveis em paralelo com a prestação de serviços que tornam a rede mais competitiva. Como tal, as Micro Redes fornecem energia limpa eficiente de baixo custo, aprimoram a coordenação dos ativos e melhoram a operação e estabilidade da rede regional de eletricidade, através da capacidade de resposta dinâmica aos recursos energéticos. Para isso, necessitam de uma coordenação de gestão inteligente que equilibre todas as tecnologias ao seu dispor. Daqui surge a necessidade de recorrer a modelos de previsão de carga e de produção robustos e de confiança, que interligam a alocação dos recursos da rede perante as necessidades emergentes.

Sendo assim, foi desenvolvida a metodologia HALOFMI, que tem como principal objetivo a criação de um modelo de previsão de carga para 24 horas. A metodologia desenvolvida é constituída, numa primeira fase, por uma abordagem híbrida de multinível para a criação e escolha de atributos, que alimenta uma rede neuronal (*Multi-Layer Perceptron*) sujeita a um ajuste de híper-parâmetros. Posto isto, numa segunda fase são testados dois modos de aplicação e gestão de dados para a Micro Rede.

A metodologia desenvolvida é aplicada em dois casos de estudo: o primeiro é composto por perfis de carga agregados correspondentes a dados de clientes em Baixa Tensão Normal e de Unidades de Produção e Autoconsumo (UPAC). Este caso de estudo apresenta-se como um perfil de carga elétrica regular e com contornos muito suaves. O segundo caso de estudo diz respeito a uma ilha turística e representa um perfil irregular de carga, com variações bruscas e difíceis de prever e apresenta um desafio maior em termos de previsão a 24-horas

A partir dos resultados obtidos, é avaliado o impacto da integração de uma seleção recursiva inteligente de atributos, seguido por uma viabilização do processo de redução da dimensão de dados para o operador da Micro Rede, e por fim uma comparação de estimadores usados no modelo de previsão, através de medidores de erros na performance do algoritmo.

Abstract

Micro Grids constitute a growing sector of the energetic industry, representing a paradigm shift from the central power generation plans to a more distributed generation. The capacity to work isolated from the main electric grid make the MG resilient system, capable of conducting flexible operations while providing services that make the network more competitive. Additionally, Micro Grids supply clean and efficient low-cost energy, enhance the flexible assets coordination and improve the operation and stability of the local electric grid, through the capability of providing a dynamic response to the energetic resources. For that, it is required an intelligent coordination which balances all the available technologies. With this, rises the need to integrate accurate and robust load and production forecasting models into the MG management platform, thus allowing a more precise coordination of the flexible resource according to the emerging demand needs.

For these reasons, the HALOFMI methodology was developed, which focus on the creation of a precise 24-hour load forecast model. This methodology includes firstly, a hybrid multi-level approach for the creation and selection of features. Then, these inputs are fed to a Neural Network (Multi-Layer Perceptron) with hyper-parameters tuning. In a second phase, two ways of data operation are compared and assessed, which results in the viability of the network operating with a reduced number of training days without compromising the model's performance. Such process is attained through a sliding window application.

Furthermore, the developed methodology is applied in two case studies, both with 15-minute time steps: the first one is composed by aggregated load profiles of Standard Low Voltage clients, including production and self-consumption units. This case study presents regular and very smooth load profile curves. The second case study concerns a touristic island and represents an irregular load curve with high granularity with abrupt variations.

From the attained results, it is evaluated the impact of integrating a recursive intelligent feature selection routine, followed by an assessment on the sliding window application and at last, a comparison on the errors coming from different estimators for the model, through several well-defined performance metrics.

Agradecimentos

Numa primeira instância queria agradecer ao Professor Helder Leite, não só pela oportunidade proporcionada de me envolver num ambiente profissional, mas também pelo acompanhamento regular, o cuidado e apoio que sempre manteve durante os meses de desenvolvimento da tese.

À equipa da Efacec que me acolheu, em especial aos engenheiros Alberto Bernardo, João Monteiro, Pedro Félix, José Ribeiro e Ismael Miranda. Queria deixar em particular, uma forte palavra de apreço pelo Luís Miguel Costa, pela maturidade e responsabilidade transmitida e pelo apoio incansável. Foi uma peça fundamental ao longo destes meses. Gostaria também de referenciar o Engenheiro Luís Filipe Azevedo, pelas sábias ideias transmitidas e pela disponibilidade demonstrada. Para todos vocês, desejo o melhor que a vida vos possa oferecer.

Queria agradecer também aos meus amigos que mais de perto me acompanharam e que comigo batalharam durante este percurso, nomeadamente o Carlos Rafael Fernandes, o João Valente Silva, o Helder Esteves, a Rita Pitrez, o João Carlos Pinho e a Maria Brogueira. A vocês vos agradeço, pelas memórias construídas e pelos bons momentos de amizade.

Gostaria de agradecer à minha família por todas as oportunidades que me proporcionaram, pelo investimento ao longo dos anos e por me darem asas para crescer.

Por último, quero deixar um agradecimento especial à Bruna, pela paciência e conforto. Por estar sempre do meu lado e me erguer nos momentos difíceis. Por me inspirar a querer ser melhor todos os dias.

“This is the moment I accept the most challenging times will always be behind me and in front of me.” - Kobe Bryant

Index

List of figures

List of tables

Acronyms and Symbols

List of acronyms (sorted by alphabetical order)

ACF	Auto correlation function
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CV	Cross Validation
ELM	Extreme Learning Machine
EVS	Explained Variance Score
FFANN	Feed-forward Artificial Neural Network
GBR	Gradient Boosting Regression
HVAC	Heating, ventilation and air-conditioning
KNN	k-Nearest Neighbors
LSTM	Long short-term memory
LTLF	Long term load forecast
MAE	Medium Absolute Error
MAPE	Medium Absolut Percentage Error
MG	Micro Grid
ML	Machine Learning
MLR	Multiple Linear Regression
MSE	Mean Square Error
MTLF	Medium term load forecast
PACF	Partial auto correlation function
PV	Photovoltaic
RBF	Radial Basis Function
RES	Renewable Energy Sources
RFR	Random Forest Regression
RMSLE	Root Mean Squared Logarithmic Error
RMSE	Root Mean Squared Error

RNN	Recurrent Neural Networks
SLV	Standard Low Voltage
SOM	Self-Organizing Map
STLF	Short term load forecast
SVM	Support Vector Machine
SVR	Support Vector Regression
VSTLF	Very short-term load forecasting
WNN	Wavelet Neural Network

Chapter 1

Introduction

This chapter provides a general overview of the topics that will be studied throughout this dissertation project[1], held in a business environment. Firstly, this work's motivation is presented, referring to the importance of the predicative models in a micro grid level as well as the adjacent benefits of machine learning algorithms applied to the load forecasting issue. Afterwards, the objectives that were thought adequate considering the context exposed in the previous section are proposed. Finally, the structure of the document will be detailed.

1.1 - Context and Motivation

The traditional energy production and distribution system is fundamentally dependent on vast generation power plants, located far from the end-point consumer, which causes losses during transport and less efficient outputs. Traditional electrical grids also rely on natural resources which are finite and pollutant. With the ever-evolving electric grid and the fluctuating electric demand and generation, arises the necessity for a decentralized power generation structure.

The concept of Micro Grid (MG), a subsystem integrated within the utility grid, is introduced to find a solution for the aforementioned issues. A MG operated in a controlled, coordinated way, either while connected to the main power network and/or islanded. These electrical distribution systems include distributed energy resources, controllable loads, power conversion circuits, storage devices and distributed generators. Taking into consideration that a large-scale energy storage system is hard to attain, the need for an accurate electric forecast increases even more.

Especially at a MG level, there is the opportunity for not only the integration of renewable sources like Photovoltaic (P)V and wind but also other resources that contribute to an optimized exploitation of the already existent energy sources, such as the energy storage systems. However, the proper control and management of such grids is highly dependent of accurate load and production forecasting models, in order to define adequate scaling for its flexible

[1] Final semester project leading to the Master's degree program in Electrical Engineering and Computers from the Faculty of Engineering of the University of Porto (FEUP)

assets. This is particularly crucial for systems with a high level of integrated RES in the MG, instead of conventional ones (like Diesel generators), or systems that although more flexible, such as the storage energy systems, have a limited capacity.

For these reasons, electricity load forecasting is crucial, in the sense of being able to determine the stability of the generation system and maintenance of scalable loads. Contingency planning, load shedding, management and commercialization strategies are all influenced by load forecasts. Moreover, having an inaccurate sense of the future load leads to avoidable expenses: predicting a higher load results in the units that were committed for dispatch being unutilized. Predicting a lower load leads to the MG management platform not committing the necessary generation units and therefore having to resort to peak power plants. In this sense, the MG management platform should aim for a deep integration of RES whilst minimizing operational costs.

In this work, it will be addressed the forecasts of electricity demand in a MG level. According to [1], the high accuracy levels of the electrical forecasting process are explained due to the presence of several factors: having a well-defined idea of the contributing factors for the estimation (electricity demand can be driven largely by season and weather conditions), having a good historical data and the skills, tools and studies to develop good prediction models on electricity demand and its key features.

With the rise of the 4th Industrial Revolution comes the premise of a wireless world and an upcoming path towards smart cities, which implies mass distribution of smart meters, access to high-quality reliable data and an enhanced platform which uses this data to manage resources and assets effectively.

On top of that, the battery powered electric vehicles market is skyrocketing and evolving into the next standard asset in the automobile industry, which increases drastically the electric consumption demand. It is estimated a very impactful increase by 2030 [2], which will alter the electric demand through all the sectors of clients.

In this paradigm, accuracy forecasting is vital to respond to the need of coupling the generation and demand amount, avoiding excess in production or scarce of possibilities to the consumer. Also the sizing for transmission lines to operate at maximum power can be planned beforehand [3] and improvements in the generation schedule can be held.

The issue of load forecasting one's electricity demand is still a difficult and a complex problem to handle due to its unpredictability and nonlinear characteristics with different levels of seasonality. It is also known the influence within the load demand scheme of exogenous variables such as temperature, wind, humidity which makes the modeling process hard to attain[4].

Nonetheless, it is important to note that this topic of discussion is a well-established field of study for the data science and overall analytics community, with already vast and immensely successful case studies.

1.2 - Objectives

As previously stated, the need for accurate load forecasting is ever more evident to the energy panel scene, as electricity is the most crucial aspect of the industrial and domestic platforms. The next issue to address would be the longevity or spatial time of the forecasting. To properly schedule and program the generation, a grid operator must predict the electricity load in, at least, a day ahead. On the other hand, grid architectures and designers must have

a time horizon of years, in order to make sure that the infrastructure is sufficient. On a smart grid level, operators should have a time precision of hours or even minutes for their predictions [3].

This dissertation intends to study and later develop forecasting methodologies based on machine learning algorithms applied to the context of a MG, for a 24-hour time horizon.

1.3 - Publications

During the course of this dissertation it was submitted and accepted a conference abstract titled: 'A Hybrid Approach to Load Forecast at a Micro Grid level through Machine Learning algorithms' for the 3rd International Conference on Smart Energy Systems and Technologies (SEST), held in Istanbul, Turkey, from 7 to 9 September, 2020.

1.4 - Structure

This dissertation is organized throughout six chapters. In this first chapter, it was intended to contextualize the reader for the subject to be discussed, its importance in the modern world and the relevance of carrying out this study for the overall energy structure panel.

In the second chapter a bibliographic analysis is carried out, with a broad overview of the different emerging and consolidated technologies, which led to the methodologies that became a support for the development of this dissertation.

The third chapter covers the data analysis, load profile characterization and the illustrative electric demand variability inherent to a MG type community. Daily, weekly and seasonal patterns are also explored.

Fourth chapter aims to provide more detail into the methodology adopted for this dissertation. It includes insights on the proposed models and overall strategy and thought process for a quality forecast targeted at 24 hours ahead.

Fifth chapter goes through the results analysis, finding the best parameters and further proceeding to evaluate them. An in-depth review in order to validate the designed sliding window method is also carried out.

Finally, the main conclusions about the proposed methods and algorithms are presented in chapter six, together with some proposals for future work, and alternative routes that could have been followed.

Chapter 2

A conceptual review on the state-of-the-art literature for electricity load forecasting

Through this chapter is presented the literature review that served as a foundation for the development of this dissertation.

Electric load forecasting plays a crucial role in the planning, operation and optimization of a MG. At this moment in time there are not only extensive but also diversified methods presented in scientific publications, journal articles or conferences about this subject. With this in mind, it is pertinent to have a structured view of the models being studied in terms of its reliability on explanatory variables, which means classifying from univariate or multivariate models.

Univariate techniques are the ones who only depend on historical load data and do not rely on external variables, like weather information, econometrics or detailed human behavior, to attain satisfactory results. A univariate time series refers to a series that consists of single observations recorded over regular time intervals.

The most representative techniques of this type are the ARIMA and the exponential smoothing methods. In terms of practicality, these models have the advantage of being able to perform in situations where external variables are hardly accessible, due to poor quality of data, high access costs associated and even a lack of expertise dealing with this domain of knowledge. They rely only on historical load data and can perform reliably under those circumstances. Overall, it is still advised to pursue academic research on univariate techniques.

Multivariate models, however, have the accuracy advantage, especially for a short-term level. This is explained mainly due to the possible inclusion of exogenous variables, such as temperature or other driving factors, who could positively impact the electric load forecast panel. The most explored multivariate techniques include multiple linear regression, artificial neural networks and support vector machines. In a long term forecast, the biggest standout point of these models are the ability to envision potential scenarios analysis, which become crucial in terms of planning and operational strategies regarding the power system [5].

A reasonable approach would be to combine the forecast power of each different technique to obtain more robust and reliable results, in a sort of multi-level approach, which is in fact nowadays being considered to be the best practice for load forecasting.

In this chapter, the forecasting methodologies were aggregated in primordial, statistical, artificial intelligence (AI) and finally a multi-level approach forming the ‘new’ hybrid methods.

Furthermore, an overview about time horizon classification and performance measurements is also presented.

2.1 - Primordial forecasting and the "similar day" method

More than a century ago, when electricity was just given its first steps, there was already a sense of energy forecasting, which was straightforward to perform. For light bulbs, for example, the procedure was just to count the amount installed and it was possibly to have an accurate estimation of electrical load throughout the day, or especially during night times. This is still a recurrent and easy proceeding for public street lighting load estimate[6].

Moving forward in time and with the growing applicability of electricity for domestic and industrial usage, forecasting started to gain depth and to deserve a more expert care. It was noticed a strong dependence of electricity demand with the weather conditions, especially due to the high penetration of HVAC systems which present irregular demand. Before the computer era, engineers approached the issue with the similar day look-up method, forecasting the future load through charts, tables and historical data from past days that present similar characteristics to the day of the forecast. These characteristics were mostly weather conditions, similar day of the week, seasonality and holidays. This method can be done through a regression procedure, if one takes more than one similar day and analyzing the relevant coefficients of previous years for those days [7]. These procedures were inherited in modern times by many control operative centers leading to more complex and precise forecasting mechanisms.

2.2 - Statistical approach

In the last decades of the 20th century computer applications exploded, and with-it, technology advancements due to the availability of diverse software packages, analytical tools and easy access to enormous amount of historical data. For that reason, a more robust and reliable way of forecasting demand emerged.

Statistical methods are based on time series analysis and its related concepts. Auto-regressive integrated moving average (ARIMA), exponential smoothing, hierarchical forecasting or regression analysis can be used to model the relationship between load profiles and influencing factors as inputs. This process is done mainly in a linear way, which probably sacrifices accuracy [8].

A more extensive review of different statistical methods is presented:

i. Multiple Linear Regression (MLR)

For simpler cases with one predictor variable, a simple linear relationship can be established between the target y and the feature x. However, when in the presence of more than one predictive feature, this model is called Multiple Linear Regression (MLR).

MLR often suffers from the wrong premise of not being suitable for modeling the nonlinear relationships between the load and the weather forecast. In reality, its linearity is referring to the equations used to solve the parameters and not to map the relationship

between the dependent and independent variables. Polynomial regression models belong to the MLR family and can establish nonlinear relationships between the electric consumption and weather variables through polynomials [5].

ii. Auto Regressive Integrated Moving Average (ARIMA)

For stochastic time series analysis usually persist four components to fit the models: seasonal, trend, cyclical and random. Seasonal is a pattern that appears in a regular interval where the frequency of occurrence is within a year or even shorter. A trend is a long term of a relatively smooth pattern that usually persists for more than one year. Cyclical is the repeated pattern that appears in a time series but beyond a frequency of one year. It is a wavelike pattern about a long-term trend that is apparent over several years. Cycles are rarely regular and appear in combination with other components. The random component of a time series appears after the previous patterns have been extracted out of the series. Therefore, when plotting the residual series, the scatter plot should be devoid of any pattern and would be indicating only a random pattern around a mean value. The methods to follow must be applied with caution in order to provide reliable and accurate time line classifications [9][10].

When removing trend and seasonality from a time series, it becomes non-stationary and therefore independent on the time which the observation is made. However, time series that present cyclical patterns are considered to be stationary, since there is no way of knowing when and where the spikes and variations of these cycles will be. A stationary time series presents no predicable behavior in the long-term expectancy and should look roughly horizontal with constant variance [1].

In order to eliminate or reduce trend and seasonality, converting non-stationary to stationary time series can be done through differencing. This process is achieved by computing the difference between consecutive observations. In other words, the system outlets the difference between the original series and consecutive lagged values of itself. The autocorrelation function (ACF) can be used for testing the stationarity. Overall, if there is a slow decay presented in the plot then the series is not stationary, while if it drops to zero quickly then it is considered stationary.

The most commonly used method for electric load forecasting, especially at a short-term level, is the Box-Jenkins Autoregressive Moving Average (ARMA) or the improved Autoregressive Integrated Moving Average (ARIMA). This latter method is a junction of the autoregressive and moving average polynomials, applied to non-stationary time series, due to the differencing parameter *d* in ARIMA(*p,d,q)*. The parameter *p* refers to the autoregressive end and the moving average order number corresponds to *q*. The full model can be written as in Equation 2.1:

$$y_t' = c + \emptyset_1 y_{t-1}' + \cdots + \emptyset_p y_{t-p}' + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \qquad (2.1)$$

Where y_t' is the differenced series, $\emptyset$ and θ are the parameters who result in different patterns and ε_t represents the white noise.

The partial autocorrelation function (PACF) can be used to figure out the maximum order value of the autoregressive parameter (*p*), while ACF is used to identify *q* and the non-stationarity, as mentioned before. The residuals of ACF and PACF can be applied for white noise testing [11][12][13].

Additionally, in order to evaluate the model and determine the most suited order for the ARIMA, it is usually used the Akaike's Information Criteria (AIC). A good model has generally low values of AIC. This evaluator is expressed in Equation 2.2:

$$AIC = 2(p + q + k + 1) - 2\log(\hat{L}) \tag{2.2}$$

Where $\hat{L}$ is the likelihood of the data, while k = 0 or 1.

As another note, while univariate modeling through ARIMA shows sufficient results, literature suggests that the inclusion of exogenous variables like temperature can impact positively the model's forecasting performance. For the case being, model nomenclature is given as ARIMAX [14].

Finally, cases where time series presents seasonal patterns such as daily, weekly, monthly or for the different seasons of the year, can be modeled by an extension of the ARIMA version, called SARIMA. This model utilizes additional seasonal terms and can be expressed as ARIMA(p,d,q)x(P,D,Q)$_s$, where P is the order value of seasonal AR model, D is the degree of seasonal differencing, Q is the seasonal degree concerning the MA model and s is the determined number of observations per year [15]. Lowercase notation is used for the non-seasonal parts, while uppercase one is used for the seasonal parts. Once again, inclusion of exogenous variables is possible and lead to the designated SARIMAX model.

For a matter of guidance and clearance, basic steps for modeling can include[16]:

1. Testing for stationarity of the time series;
2. Calculating correlation coefficient after complying with conditions;
3. Model identification;
4. Parameter estimation;
5. Model test;
6. Model optimization.

2.3 - Artificial Intelligence approach

The 21st century was the definite affirmative time for AI to establish itself as one of the most preeminent and promisor areas of research amongst technology fields. Its growing number of applicability descendants goes from Robotics, Autonomous Vehicles, image and voice recognition, medical diagnosis, assisted tasks and even outperforming humans at games like chess, strategy and overall cognitive thinking[17].

Machine Learning (ML) grows from within AI as a field of several different algorithms and mathematical problem-solving capability methods. As opposed to traditional programming relying on the 'if-then' logic and step-by-step manual decision schemes, these algorithms can learn by trial and error and actually optimize the desired output on its own. ML is about making

sense of the existing data, choosing features and then making predictions about new data instances for a specific task at hand. Procedurally, the training/learning phase is where the task model is created, and testing/validating is where the model performs for the purpose that has been designed to, assessing its capability of generalize. In a simple way, testing is how 'smart' the model became from the training he was subjected to. Additionally, there should be extra care for overfitting cases, where the model perfectly fits the training set and generalizes poorly for new datasets.

Training can be held as supervised learning if the data corresponds to description of situations and understands the desired behavior of the network for each particular situation. Unsupervised learning can also occur if the network is only exposed to unlabeled data but it does not receive information about its expected behavior or is influenced about that matter[18].

These models can fit in the following frameworks, as illustrated in Figure 2.1:

- **Classification:** This class is applicable in a supervised learning environment, where predictions are made based on a set of examples. In supervised learning there is an input variable that consists of labeled training data and a desired output variable. The algorithms are responsible for learning the function that returns the output, by training the labeled input data. The dataset is usually split into training and testing with the majority of the data serving the training purpose. Specifically, for classification, the data is being used to predict a categorical variable, meaning they will belong to a specific group with identical characteristics. The input consists of labeled data, from which training occurs. The model built from the training phase is then applied to unknown testing data with the purpose of identifying its belonging labels. The simplest case is the binary classification, where there are only two labels to decide from. However, there can be examples with a high number of labels called multi-classification. The sizing of data and the amount of labels weights in the decision of the algorithm to use. Support vector machines (SVM), Decision Trees or Random Forests can be used for classification.

- **Regression:** This group is similar to the classification one and so it works as a supervised learning the same way it was explained before. The difference is that regression is applicable to predicting continuous values through a best fit line, or overall a function that maps trends amongst points of data. Forecasting problems can mainly be seen as regression issues.

- **Clustering:** This class of algorithms aims to build coherent groups of data instances. This time, the machine is presented with totally unlabeled data with the purpose of discovering intrinsic patterns from the given data and segment it into perceived groups. It does not attribute labels or meanings to groups. There can be made some analysis by the user in order to decide the meaning and find patterns within each group. Since this class deals with unknown data, clustering is a sub-class of unsupervised learning. K-means, hierarchical or k-Nearest Neighbor can be used for clustering problems. For a next day prediction, clustering can be used to form subsets of the data for each 24 hours of the day, and then use this information as an input to the model to realize the forecast.

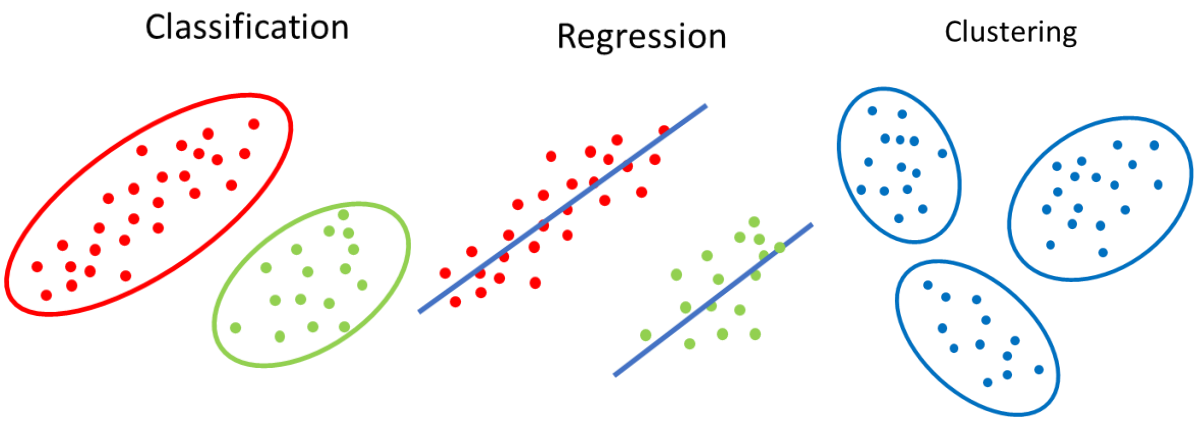

Figure 2.1 - Simple illustration of ML categories

As seen previously, ML models can be applied to data to find common patterns[19], that form models to make predictions on the selected data and it can be simply formulated as:

$$f(x_i) = y_i \tag{2.3}$$

Where x_i is a data point in regression problems, to predict some y_i. In classification problems, the output is a class.

With the premise of its high potential, researchers devoted innumerous studies to a special class of AI, the ML algorithms and particularly Artificial Neural Networks (ANN) with all of its derivatives. However, some researchers hesitate in declaring a superior overall capability of the ML methods, claiming low values for the forecasting accuracy and advocate practitioners to also rely on statistical traditional forecasting methods, reinforcing that new approaches should be properly tested through different datasets to avoid the risk of falling in generalized conclusions that lead into 'generating implausible solutions, leading to exaggerated claims[17].' On the contrary, [20] states that 'The analytical methods work well under normal daily circumstances, but they can't give contenting results while dealing with meteorological, sociological or economical changes, hence they are not updated depending on time.'

In other studies, more extensive reviews of ML algorithms are held. In [21], a feed-forward neural network with multiple layers is applied to forecast a 24-hour load demand. Different features were used that ultimately lowered the maximum forecasting error such as weather information, historical data and special days of the calendar. Compared to the conventional statistical approach, which relies on time series analysis to establish linear relationships with the past data, the AI solution showcased better results.

One of the most common ML algorithms, the Support Vector Machine (SVM) is applied as regression in forecasting financial time series [22]. This work develops a model capable of optimizing the kernel functions necessary for the support vector regression machine, whilst building clusters to partition the time series data into several contexts. Results showed good generalization performances and that the algorithm implemented reduces prediction time.

On another note, ML models can also fall in the category of 'black-box' models since they are being characterized on the functional relationships between system inputs and system outputs. In other words, one could say that these models are sacrificing interpretability for accuracy and performance on the process. For load forecasting purposes, ANN models do not

clearly offer many insights on the 'which' or the extent of the impact features have on the accuracy of the forecast.

The most commonly used 'black-box' model is the ANN. Its application does not require much in terms of background on statistics or data analysis experience and with the computing advancements on run time, its popularity is set to increase. Adding to this, there are modern software packages like MATLAB that allow forecasters to explore the ANN model structure and experiment the number of neurons, layers and functions with moderate simplicity[5].

ML models are by nature nonlinear and nonparametric as it is shown in [23], where is concluded that ANN are successful at dealing with the complex nonlinear relationships that exist between the load and the influencing features.

There is not yet an overall grading for classifying the best algorithms, as they can outperform each other depending on the context of the model, the different features and the weight of the variables on the output. The forecasting error is variant and highly dependent on the specificity of the issue and the extent of the dataset in hand. It is mainly the data and the objectives aimed that define the adequate technique and not the other way around[5].

Following the study of ANN other models appeared like the Support Vector Machines, k-Nearest Neighbor, decision trees, gradient booster or random forest. The next segments will give a better and deeper understanding of the mechanisms behind the different ML algorithms.

2.3.1 – Artificial Neural Networks

ANN as estimators, are the state of the art for electrical load forecasting. As mentioned before, distributed energy systems complexity due to the uncertainty and diversity of consumer's profile sets a good practice ground for the non-linear interpretability of ANN. This computational model has the ability to learn, adapt, generalize or organize/cluster data.

The models for Artificial Neural Networks (ANN) are based on the neural structure of the brain, which learns from experience. The neuron is the basic element in any neural network. Several neurons 'mounted' together form a layer, which then can be connected through links to the next layer. Neurons in different layers are interconnected, while there is no connection between neurons in the same layer[24]. For Feed-Forward Artificial Neural Networks (FFANN), connections are always unidirectional in a progressive way, from the input to the output. This means that information always moves forward, never going backwards. In the case being Recurrent Neural Networks (RNN), it is possible to acquire temporal dynamical behavior, with the nodes forming a directed cycle between them, for a temporal sequence. This happens due to the network allowing feedback exposure, with the neurons having connections that permits information to flow backwards. These networks use their memory to enhance processing capability and adapt to different length of inputs.

However, the complexity of the network is determined by the intermediate layer or hidden layer, varying from none (equivalent to linear regression), depicted in Figure 2.2, to more than one. For most of the problems, one hidden layer has shown to be sufficient, as many more than one can already be qualified as 'deep learning' territory. For the case being none, application of ANN is unnecessary, although still accomplishes the intended result, as the data can be linearly separable.

Different rules can be adopted to determine the number of neurons composing the hidden layer. Some researchers rely on trial and error, but most suggest that the optimal size for the hidden layer is between the size of the input and the size of the output layers. Another 'rule

of thumb' suggests having the number of neurons to approximately 2/3 the size of the input layer, plus the size of the output layer. Given these rules, one can set the initial searching limits for algorithms like Evolutionary Particle Swarm Optimization (EPSO), or construct the parameters grid so that methods like Grid Search or Random Search end up selecting the best possible cases of parameters, within the ones presented.

From **[1]** it was taken an overview of a two-layer network.

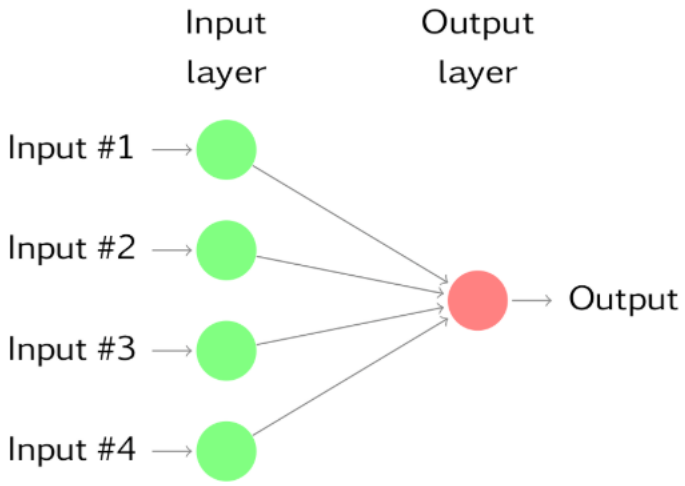

Figure 2.2 - Simple neural network equivalent to a linear regression

One important aspect to preserve in the neural networks is its ability to generalize. This means that the number of nodes should be kept as low as possible, to contradict the network tendency in becoming a memory bank, perfectly fitted to the training set. Although the performance will be optimal for the trained data set, it will fail to generalize on different new samples.

During training process, the weight values are consecutively updated to reach optimized parameters in order for the network to perform in a desired way, by repeated exposition to a set of data. This process is called backpropagation training, where the weights are updated through the propagation of the error from the output layer into the input ones. Additionally, the information flows unidirectional in a feedforward way, from the output to the input. An illustration of this process [25] is shown in Figure 2.3.

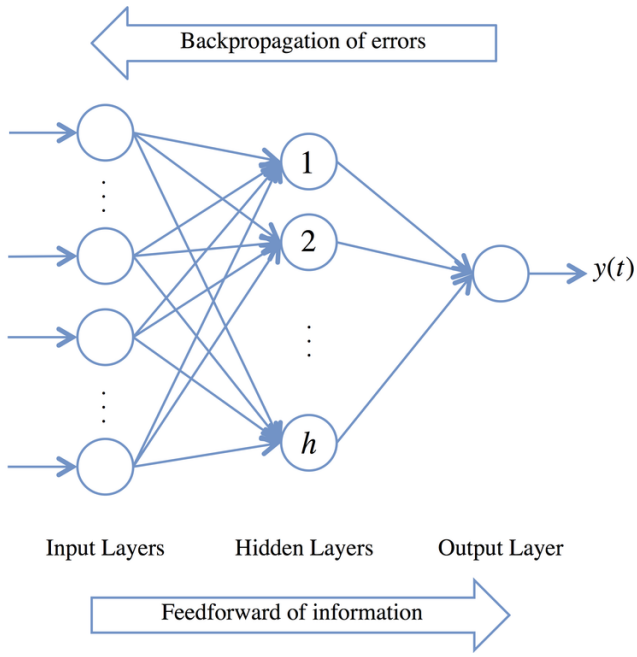

Figure 2.3 - Typical feedforward network with backpropagation training

The links that form the unidirectional connections are associated to synaptic weights. Any signal that is issued by a neuron is multiplied by the weight of the connection before entering the next neuron. The process is better understood in the following Equation 2.4, from [18].

$$O = f(\sum(w_{ij}x_j)) \tag{2.4}$$

Where O is the output of the neuron, w_{ij} is the weight associated to the connecting layers, x_j is the input to the neuron and f is the activation function. Consecutively changing the weight is what allows the network to learn and adapt.

Activation functions can be categorized as a threshold function, piecewise linear function or sigmoid function (including respective variants). The latter being the most recurrently used. These sigmoid functions [26], are expressed in Equation 2.5 and can have variations for serving different purposes:

$$S(t) = \frac{1}{1 + e^{-t}} \tag{2.5}$$

Fundamentally, the training process has the purpose of minimizing the quadratic error between the signal generated by the network and the intended target by a progressive correction of the synaptic weights' values. As already previously mentioned, known as a backpropagation algorithm presented in Equation 2.6, and the error is expressed as [26]:

$$E = \sum \frac{1}{2}(O_p - O_m)^2 \tag{2.6}$$

In which O_p is the prediction output while O_m being the measured desired output.

Backpropagation (BP) networks are gradient descent algorithms, meaning the network weights are altered based on the negative of the gradient for the performance function [27]. *Steepest descent* is used for minimization problems, while *steepest ascent* is used for maximization ones.

It is common to designate as Multilayer Perceptron (MLP) a typical feed-forward ANN consisting of at least three layers of nodes, well visible in Figure 2.4: the input, hidden and output layers from which, starting from the hidden layer, each neuron uses a nonlinear activation function. MLP uses backpropagation for training purposes [28][29].

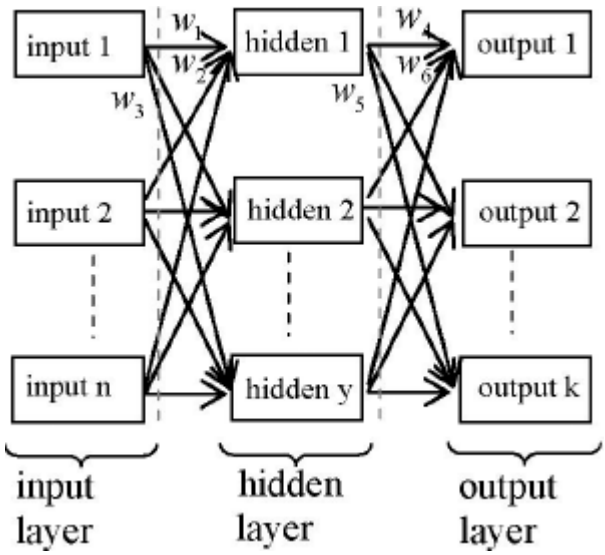

Figure 2.4 - Multilayer Perceptron Structure [18]

In [26], it is showcased an overall successful application of ANN, having it reach an impressive coefficient of determination (R^2) [30] of over 0.94 in forecasting the energy consumption for a period of 20 days, based only on five previous days. The method behind it proceeded to take back as an input the forecasted output, repeating the process for the following days.

Meanwhile, another work [31] studies to more detail the MG load forecasting issue, which presents particular difficulties dealing with the disaggregated load curves. For the purpose of minimizing forecast errors, only relevant inputs were selected, and it was noted that exists a relationship between said errors and the number of training patterns utilized. This brings special caution to the selection of training data to be used for the system.

Additionally, the following paper [32] developed a methodology to forecast electric consumption on a household, despite its unpredictable nature. An ANN architecture was developed and the Levenberg-Marquardt [33][34] algorithm, which is a combination of the steepest descent method with the Gauss-Newton quadratic convergence algorithm, was used to train the network. Results showed the possibility of forecasting electrical consumption with certainty.

Different typologies of ANN are presented next:

i. Long Short-Term Memory (LSTM) for Recurrent Neural Networks

Time series analysis for future load estimate presumes building a model with memory capability. This is where RNN fits in. This model was proposed to better deal with the model complex processes and in particular to learn temporal behaviors[35]. Typical feed-forward networks struggle with this matter. Essentially, RNN edge is the ability to transpose information from the output to the input, inciting information to flow from previous time-steps to predict future time-steps[36].

On the other hand, RNN faces a considerable gap on the issue of vanishing gradients. Long short-term memory RNN address this issue by integrating gating functions into the system dynamics. LSTM tries to solve the aforementioned vanishing gradient problem by not imposing any bias towards recent observations whilst maintaining constant error flowing back through time[37]. The process consists of maintaining a hidden vector, and a memory vector for the purpose of controlling state updates and outputs at each time step, respectively [38]. The hidden unit output fetches short-term information from previous time steps and restores it. Memory vector is responsible for restoring the long-term memory[39].

For these reasons, LSTM architecture was developed with the purpose of increasing accuracy prediction levels and dealing with deficient capability of classical RNN models[36] and to allow the network to learn dependencies in a longer-range matter[37]. Particularly in finding long term dependencies in the data, LSTM have proven to be very useful as they are the current state-of-the art for ANN time series forecasting.

ii. Auto Regression Neural Network (NNAR)

Based on the same purpose as auto regression for time series analysis, lagged values of a time series can be applied as inputs to a neural network. FFNNAR(p,k) is a feed-forward neural network auto regressive utilizing a multilayer scheme based on nonlinear autoregressive method for time series forecasting. Attribute p gives us the amount of input nodes, referring to the number of lags intended for the time series. The second attribute, k, indicates the amount of nodes belonging to the hidden layer, which uses a logistic function for activation[40]. This notation refers to non-seasonal data. For example, a FFNNAR (10,8) or simply NNAR (10,8) is a network with the last 10 observations used as inputs for forecasting the output, and with 8 neurons composing the hidden layer. For a matter of perspective, a model with just a value on the first term p is equivalent to the ARIMA(p,0,0), meaning it only has values on the autoregressive term and no values for the integration and the moving average.[1].

For seasonal data, it is of good practice to concatenate the last observed values from the same season as inputs. Broadly, a NNAR(p,P,k) has inputs (y_{t-1}, y_{t-2},..., y_{t-p}, y_{t-m}, y_{t-2m}, y_{t-Pm}) and a k number of hidden neurons. There is a high correlation with the ARIMA($p,0,0$)($P,0,0$)$_m$ model, except it has no restrictions on the parameters that ensure stationary[1].

iii. Convolutional Neural Networks (CNN)

For the purpose of processing time series with continual properties, LSTM or RNN architectures have shown to be effective. However, when dealing with feature

extraction, performance levels are low compared to CNN ones. When applied to short time series data, CNN based models can extract the features effectively whilst maintaining the time series characteristics between them [41]. CNN uses a variation of MLP through backpropagation algorithms, thus requiring minimal preprocessing [42]. These networks are successfully applied in image and video recognition, text categorizing and language processing, among others.

The foundation of CNN is based on the visual cortex of humans and animals. This part of the brain is responsible for processing any visual information. This analogy transposes to the network in the way that it can process images efficiently by separating the visual field into small regions, denominated receptive fields, which processes information in a local manner. Through the use of locality (only locally close neurons are connected), the number of connections between layers decreases which potentiates the use of a larger amount of layers, compared to fully connected networks [43].

Locality process is accomplished through convolution, which operates in two distinct functions in the following way:

$$s(i) = (x * w)(i) \tag{2.7}$$

Where the convolution operator is denoted as *, x is the input and w is the weighting function, which for CNN purposes can act as a *kernel* or a $filter$. Nevertheless, it can be trained in the same fashion as regular weights in ANN. Finally, s is the product of the operation, designated feature map. Feature map is created when the convolutional process runs through the entirety of the input vectors [44].

Two types of layers are built within CNN. Firstly, the convolutional layer, where locally connected neurons share the same weights, which leads to faster training. Convolution and weight sharing are implemented through a filter with predetermined specific *kernel* size responsible for determining the nodes that will share weights [45]. Typically following this layer is a *maxpooling* layer, responsible for computing the maximum value of a selected group belonging to the convolutional layer, ensuring invariance to shifts in the input data [45].

These three studies [45][46] and [47] apply CNN based models for time series analysis and forecasting.

iv. Radial Basis Function Neural Network (RBFNN)

In order to fight against slow learning speeds and high computational demands, the RBFNN is established. These networks use radial basis functions as activations functions. Study [48] claims RBFNN as the best partial approximation of continuous function, while the typical BP network as the network for overall approximation. For the process of BP training, there is an adjustment of every parameter of the network for every input/output. Additionally, BP network suffers not only from the local minimum issue, but also from the ambiguous rules applied to determining the number of nodes in the hidden layer, which makes it difficult to find the ideal network.

Structure wise, RBFNN is composed of three layers. The first layer corresponds to the input neurons, which the amount is dependent on the number of input data. This

primary layer is in charge of signal translation and transmission. The second layer is the hidden one, where the number of neurons depends on the complexity of the problem in hand. Hidden layer is responsible for regulating the parameters of the activation function, which is usually a Gaussian Function. Training occurs by constantly adjusting the weights. Nonlinear data is converted from the input to the hidden area. The third layer matches the dimension of the output data. Linear weights are adjusted in this layer, where the linear optimization strategy is usually adopted[49][50].

Moreover, in order to achieve better performances, the usual approach is to increase the dimension of the hidden space. It was taken from [51] an illustration of the neuronal structure of the radial basis function neural networks.

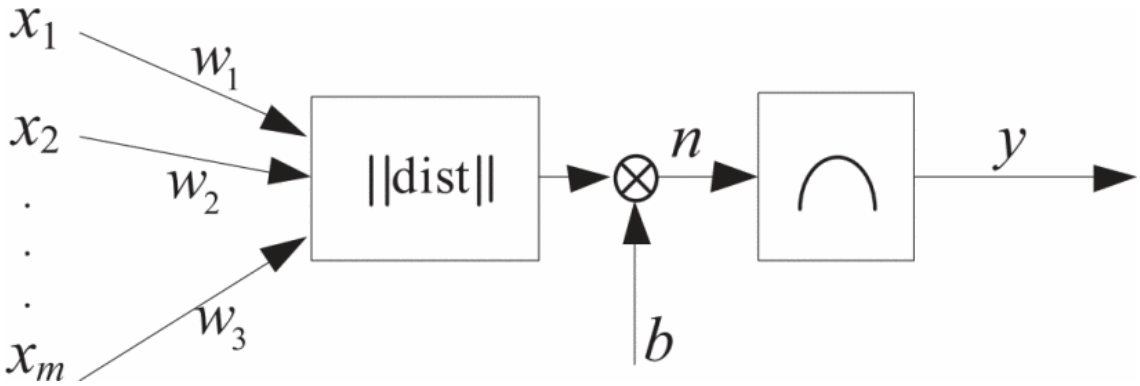

Figure 2.5 - Neuronal Structure of a radial basis function neural network

v. Wavelet Neural Network (WNN)

Neural network theory and wavelet analysis combination are the foundation for the wavelet neural network. They are both directly connected and simultaneously independent of each other [52]. There have been different approaches in applying wavelet theory for forecasting purposes. One uses the wavelet transform as a means to compose the load time series into its wide range of frequency components, by preprocessing. Another approach consists in building the WNN in a way that the activation functions of the hidden neurons for a feed-forward ANN, are, actually, wavelet functions [53].

According to [49] 'Wavelet neural networks combine the good time frequency localization properties of wavelet transform with neural networks self-learning function, so it has strong approximation ability and fault tolerance capability at the same time.' Two types of activation functions compose the hidden layer: the wavelet function and the scaling function.

In the WNN, the nonlinear sigmoid function is replaced with the nonlinear wavelet basis and so the load historical sequence is completed with its linear superposition [54][55][56]. The forecast values can be given as:

$$\hat{y}(t) = \sum_{k=1}^{L} w_k \ \varphi\left(\frac{t - b_k}{a_k}\right) \tag{2.8}$$

Where:

$\hat{y}(t)$ concerns the forecast value of given time series y(t), w_k are weight numbers, b_k the translation factors, a_k is the expansion or contraction factor and L is the wavelet basis number.

In a generalized manner, WNN goes through the following steps[48]:

- Wavelet decomposition;
- Predicting the intended forecast by the use of neural networks algorithm;
- Reconstruct the values.

vi. Extreme Learning Machine (ELM)

Extreme learning machines are efficient algorithms belonging to the FFNN family and are used for classification, regression and clustering. To enhance performance, ELM minimizes training error and learning speed, by randomly generating weights rather than adjusting network parameters in every iteration. The main concept of ELM is turning the learning problem into a least squares problem, which reduces iteration time. Other improvements of this method were proposed such as the evolutionary extreme machine learning (E-ELM), to remove redundancy between hidden nodes, or the pruned-ELM (P-ELM) aimed at pruning the network during training [57].

Nevertheless, these models are known to produce good generalization performance and learn considerably faster than backpropagation networks. Additional literature [58][59][60] suggests they can even outperform SVM for both classification and regression problems.

2.3.2 – Support Vector Machine (SVM)

SVMs are also very popular amongst forecasters and are considered a robust and reliable mechanism. SVM is a powerful supervised learning algorithm based on statistical learning theory that is usually used for classification, recognizing patterns, visual features, amongst others. When this method is applied to function estimation it becomes a regression problem, and so it is usually referred as a support vector regression (SVR) method [61].

This approach inherits statistical learning theory fundamentals and seems to offer excellent generalization solutions for real-life examples, both for regression and classification. The most usual formulation of this method is the ε-SVR, the ε-tube support vector regression, illustrated in Figure 2.6 [62], first introduced by *Vapnik* in 1995 [63]. For regression, the set of training data includes predictor variables and observed response values. The goal of this method is to find a function $f(x)$ that admits at most a ε deviation from the obtained targets y_i for all the training data while at the same time, maintaining it as flat as possible [64]. This technique relies on kernel functions to be able to define an hyper-plane to perform the linear separation. Polynomials and Gaussians are the most popular kernel functions.

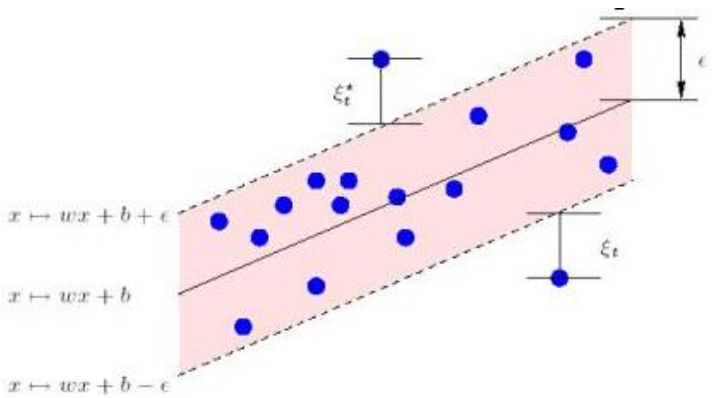

Figure 2.6 - Epsilon band for SVR with selected data points

Another method for the penalty/deviation function is to attribute one of two slack variable penalties, depending on whether they lie above, or below the tube.

The use of SVR provides some benefits:

- Good results when in the presence of a clear margin of separation;
- Effective for high dimensional spaces and planes;
- Performs well for cases where the dimensionality number often exceeds the number of variables;
- Memory efficient due to the use of support vectors;

2.4 - Hybrid methods

Hybrid models usually employ a combination of linear and non-linear models for the time series linear and non-linear components, respectively. They can also have entirely an artificial intelligent approach, segmented into tasks and distinct phases. These models are based on a multi-level approach to the forecasting problem. First step can be a pre-processing of the features, in an input selection process. The second phase is usually the implementation of the trained model to cope with the nonlinear characteristics of the load. Papers like [65][66] highlight the combined strength of an hybrid optimization method. On the other hand, the designations and nomenclatures for classifying processes of forecast are becoming more ambiguous, due to their related concepts, applicability and nature. The boundary between statistical and artificial intelligence techniques is losing its purpose as a result of the collaborations between the two to combine the best inherent properties for the case study in hand.

In [67] is proposed a new electric load forecast strategy for Medium-Term Load Forecast, where a feature selection process prepares the inputs of an organized data model. Then, the hybrid training mechanism includes a Levenberg-Marquardt learning algorithm and a novel stochastic search technique, called IDE.

The authors in [68] developed a two-stage hybrid network for Short-Term Load Forecast, more specifically to train and predict 24-hour loads of next day. In the first stage, a self-organizing map (SOM) is implemented to cluster the input training data. These inputs were

separated into different subsets considering similar dynamical properties, through unsupervised learning. There was also a classification of the input data into regular days and anomalous days. Next phase included fitting the training data for each subset using several groups of 24 SVRs, by supervised learning. The number of SVRs for each subset is chosen due to the prediction being for a 24-hour load of the next day. Putting it in a simple manner, this method breaks the time series into different segments or subsets where data in the same segment can be modeled by the same SVR due to similar properties. These model presents advantages such as dealing with non-stationarity, differentiating regular days and anomalous days with different schemes, strong robustness and adaptability to changing power systems or markets.

Results of work [69] support the superiority of a wavelet based approach for STLF over non-wavelet methods. The wavelet decomposition process consists in decomposing the historical load through smoothening by deleting the high frequency components and fed as input to the proposed models for prediction. The forecasting process includes wavelet decomposition for preprocessing of the signal, smoothening the data for a smooth and fast training of the network and then forecasting using this pre-processed load as an input to the desired forecasting model.

A new approach to STLF based on ELM is proposed in [70], where several improvements were implemented to boost performance. These processes included a novel wavelet-based ensemble scheme to generate the ELM forecasters, a hybrid learning algorithm combining ELM with the Levenberg-Marquardt method to improve converging accuracy and a feature selection method based on the conditional mutual information to select a compact set of input variables. Lastly, least squares regression is applied to combine the individual forecasts. The methodology proposed is tested for both 1-hour and 24-hour ahead load forecast using data from two electric utilities. The article claims to alleviate difficulties such as random weights and biases for the neural networks, overtraining and wavelet parameter selection.

Another paper [71] studies STLF giving all the focus to fuzzy neural networks, where it presents not only a wavelet fuzzy neural network using the fuzzy wavelet features as the inputs to the fuzzy network, but also a fuzzy neural network using the *Choquet* integral. The model produces forecasts with a lead time of 1-hour. Finally, a more recent article [72] presented a hybrid AI (OP-ELM) and deep learning (LSTM) solution for a distribution load forecasting system, specifically two South African substations. Results were intriguing giving the fact the investigated models like the optimally pruned extreme learning machine (OP-ELM) and the Adaptive Neuro Fuzzy Inference System (ANFIS) achieved lower levels of error without the inclusion of the temperature as an input. The opposed happened with LSTM, where accuracy was at its highest. As a result, LSTM was the one with the overall highest accuracy forecast.

2.5 - Time horizon review and classification

It is important to note that this is not the norm adopted for this dissertation, which refers to a 24-hour forecast as a LTLF, and 3-hour forecast as STLF. The following classifications were made based on commonly used state-of-the-art descriptions.

Undoubtedly, it is of good practice to classify a time series forecasting based on the longitude of its horizon. For most literatures there is a segmentation in 4 parts, with no golden rules for the time partition, as it is mainly on the forecaster's hand to decide:

- **Very short-term load forecast (VSTLF)**, which includes a prediction for less than 24 hours ahead.
- **Short term load forecast (STLF)** including time horizon of a few days to one/two weeks. They are particularly relevant in the day-ahead planning of the power system, taking in consideration the significant changes in the load profile in a short period of time, on a MG level. Note that for the case of large distribution or transmission systems, the predictions are made by aggregation, reflecting the load profiles of several costumers. For this reason, instantaneous changes are hardly observed due to the amount and diversity of customer's consumptions[8]. However, that is not the case for MG level, where changes in the load profile can be detected and have an effect on the overall scheme. This for itself is a premise for the challenges that STLF offers.
- **Medium term load forecast (MTLF)** which makes predictions for a few weeks to a few months in advance.
- **Long term load forecast (LTLF)** referring to forecasting load profiles with 1 to 10 years ahead.

In addition, more exact classification only depends on the paper and what the author decides the cut-time horizons should be. This time horizon review and classification refers to the standard norm according to the state-of-art literature on the forecast research field.

It is also important to highlight the significance of STLF in the development of MG and smart grids, as it will be the focus of this dissertation. In these environments, it is particularly relevant to have a well-established power system structure, where it is mandatory to sustain enhanced flexibility and control.

2.5.1 - Short-Term Load Forecast review

Keeping up with the amount of STLF related papers published every year would be considered a challenge. Nonetheless, solid reviews play a fundamental role in setting a path to future developments. These works are essentially determinant when they illustrate real-life applications, extract concise conclusions on the advantages and disadvantages of particular methods towards the problem discussed and present a future development path in terms of further research needed.

A study carried out in this article [73], showed diverse newly designed ML algorithms that were used to train a RBF for load forecasting 24 hours ahead and showed promising results. In [74] is presented a generic approach to STLF based on SVR machines. For the purpose of improving effectiveness and reducing the operator's interaction in the model-building process, two optimizing solutions were adopted. The first one was the use of feature selection algorithms, for generation of model inputs. The second one was based on particle swarm optimization techniques for the optimization of the SVR's model hyper-parameters[2]. Results claimed improved accuracy when compared to other state-of-the-art algorithms and trained and tested on popular data sets.

[2] Hyper-parameters are parameters whose values are set before the training process begins and can impact the accuracy of the estimator. Its optimization is often a necessary and beneficial practice, included in the forecast methodology.

Another SVR based forecaster was used for the distributed system in [65], where data normalization was used as a pretreatment of the historical load data due to its unpredictable character and abrupt changes. Similar to the case previously presented, it is proposed a two-step hybrid optimization algorithm in the hopes of pursuing the best parameters. The first one is a grid traverse algorithm used to shorten the parameter searching zone from a global to a local area. The second one, based on swarm optimization determines the best parameters in that searching area. After the model is built, short-term load forecast is performed for the distribution system. The proposed approach outperforms other methods like ARIMA, genetic algorithm based SVM, and ANN, considering time consumption and forecasting accuracy. With this being said, it is clear that SVR based models have seen some successful applications.

On another note, to solve individual residential households load fluctuation, it was proposed in [75] a LSTM recurrent neural network-based framework for STLF. This technique is one of the latest and most popular in the deep learning field. In this particular case, the dataset included a 3 month-period from the 1st of June to the 31st of August, concerning the year 2013. The data was split in training, validation and testing with a 0.7/0.2/0.1 ratio. The training set is used to train the model, the validating set selects the best models and the testing data returns the result evaluation. Hyper-parameter tuning was considered essential but didn't receive much attention due to the large number of models, therefore, the focus was on identifying the candidate method with the best overall performance.

Again, for a small-scale decentralized energy system, following paper [76] proposes a ML based approach with an integrated feature selection tool. A Gaussian process regression is used for measuring the fitness score of the features while a binary genetic algorithm is applied for feature selection process. Training occurred with a two-year hourly dataset and tested with a different one-year hourly dataset. Additionally, to evaluate the performance of the selected predictor subset a FFNN was implemented. Annual Medium Absolute Percentage Error (MAPE) results showed values around 1.96%, which is a promising value bearing the nature of the issue.

One well known method to answer the STLF matter is the semi-parametric additive model, formulated in [77], where it is estimated the driving force of different variables on the electric demand. Several inputs were considered such as calendar variables, actual lagged demand variables and historical and forecast temperature traces. Prediction intervals forecasts were also addressed through a modified bootstrap method that deals with the complex seasonality evidenced in the electricity demand data. Finally, the forecast was made for up to seven days ahead in a half-hourly interval.

Furthermore, in [78] is presented a model based on deep residual neural networks for STLF, with a two-stage ensemble strategy being used to improve the generalization capability of trained model. A deep convolutional neural network model based on multi-scale convolutions is tackled in [79], while an improved wavelet neural network trained by generalized extreme learning machine is the approach for a probabilistic load forecasting on the Ontario and Australian electricity markets data. A fuzzy logic algorithm for classification is used in a multi-layered FFNN in [80] where the fuzzy set subdivides hourly data into various classes of weather and then the neural network is used for prediction.

2.5.2 – Long-Term Load Forecast review

Literature review for LTLF is not nearly as extensive as the STLF's one. These forecasts are being used for system planning, demand side management and finance. Rob J. Hyndman proposes a new methodology to forecast the density of long-term peak electricity demand [81]

where, once again, a semi-parametric additive model is used to map the relationship between demand and its driving factors. The demand forecasting is then executed by means of a temperature simulation, assumed future economic growth and residual bootstrapping, applied to forecast annual and weekly peak electricity demand for South Australia. Probabilistic forecasting through MLR is applied in [82] where three key elements were modernized, namely predictive modeling, scenario analysis, and weather information. The concept of load normalization was presented and demonstrated in a simulation.

In order to solve long-term retail energy forecasting, the issue was divided into load per customer forecasting using regression analysis, and tenured customer forecasting using survival analysis in [83]. A knowledge based system was implemented in [84] to support the choice of the most suitable forecasting model for medium/long term power system planning. The key variables that affect the electrical demand are firstly identified and then rules concerning these variables are obtained and stored in the knowledge base.

Finally, several ML methods such as FFANN, SVM, RNN, KNN and Gaussian Process Regression are used for a LTLF panel in the following paper [85]. Mean absolute percentage error (MAPE) compares amongst methods the performance results, giving an edge to FFANN, despite the remaining methods performing proficiently.

As a final note for this sub-chapter, it should be highlighted the significance of the loads' nature on the planning and structural development for the methodology proposed for each forecast resolution. The differentiation between workdays and weekends, the variability and unpredictability of the load demand, and the seasonal patterns all bear an impact on the overall strategy that should be constructed. A more thorough explanation on load profiles and their impact on the forecast will be carried out in Chapter 3.

2.6 - Performance measurement

Forecasting is essentially a regression analysis problem and the metrics used to evaluate these models must be able to perform on a set of continuous values. The most popular regression metrics are presented next:

$$MAPE = \frac{1}{N}\sum_{i=0}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100 \tag{2.9}$$

$$MAE = \frac{1}{N}\sum_{i=0}^{N}|y_i - \hat{y}_i| \tag{2.10}$$

$$MSE = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2 \tag{2.11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2} \tag{2.12}$$

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.13}$$

With:

$$\bar{y}_i = \frac{1}{n}\sum_{i=1}^{N} y_i \tag{2.14}$$

And:

$$\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N} \epsilon_i^2 \tag{2.15}$$

Where *N* is the total number of time instants, y_i is the actual target value and $\hat{y}_i$ is the corresponding forecasted values.

In fact, methods such as RMSE/MAE are used to evaluate the variance in the errors, the value itself doesn't have much meaning, unless compared with different models. However, for the R-Squared model who shows how well the independent variables relate with the model's target, the closer the value is to 1, the better performer is the model.

One important metric adopted further in this dissertation, is the Explained Variance Score (EVS). With *Var* being the Variance, which corresponds to the square of the standard deviation, then the explained variance is estimated as:

$$EVS = 1 - \frac{\sum_{i=1}^{N} Var\{y_i - \hat{y}_i\}}{\sum_{i=1}^{N} Var\{y_i\}} \tag{2.16}$$

Best scores are closer to 1, lower values are worse. However, for the purpose of plot interpretability, in this work the EVS is given as 1-EVS, in a way that lower values correspond to the best scores.

Usual procedure for evaluating a model consists in splitting the data set into training set, where it trains the model, and the validating set, for testing assessment. The model's performance is then evaluated on one of the error metrics presented before, to determine its accuracy. These evaluation methods can give a general approximation of the estimator performance; however, one particular test set can reproduce accuracy results somewhat distinct from others.

In a scenario where it is intended to return the hyper-parameters of an estimator, expected results include values who consecutively converge for the same quantifiable results, while a regression estimator might produce different results for each turn of testing, since the instances that are selected to compose the training interval are iteratively different.

With this in mind, *K-Cross Validation* comes as a solution to the aforementioned issue. This method divides the data into specific *k*-folds and ensures that each fold is used as a testing set at some point of the process. For a scenario of *k* = 10, the data is split into 10 folds and assigns the first fold for testing while the rest are used for training. This process is repeated iteratively until each of the 8 folds have been used for testing purposes. Figure 2.7 serves to better visualize the process.

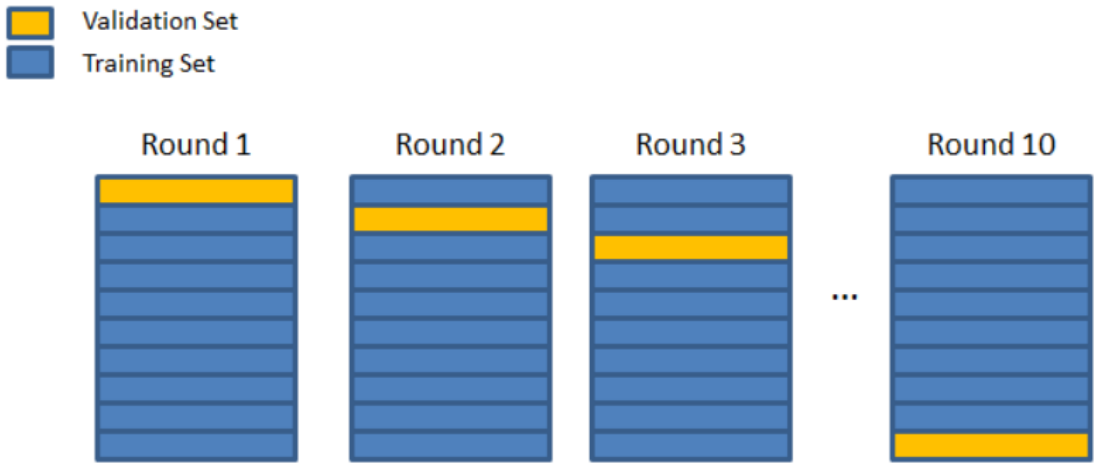

Figure 2.7 - Cross-Validation illustrative process

Cross-validation (CV) is used in machine learning to estimate the generalization capability of the model on unseen data. To reduce variability, most methods perform several rounds of cross-validation using different partitions. The results of the different rounds are combined and produce an overall estimate of the model's predictive performance [86].

Chapter 3

Load profile analysis

The penetration of RES resulting in low-carbon emission technologies and the electrification of transports and services is propelling the necessity to understand the time and amount of electricity which is being used and generated by consumers. Additionally, the external factors which influence the overall electricity demand should also be a target of deep study and analysis.

MG networks use feeders and substations to connect to its end users, therefore encompassing a diverse community stratum. The feeder's size can vary from small sizes which provide energy to a few residential households, to large sizes feeding hundreds of customers. Some MG communities may include hospitals, schools, small or medium size companies while also serving local family neighbourhoods. Additionally, they will also have to feed streetlights and traffic lights.

In both the case studies developed for this dissertation, the data was aggregated, which results in smoother contours and profiles, therefore making it easier to forecast. Case #1 poses a more foreseeable analysis, showing daily load profiles deeply standardized and who follow very repeated load demand behaviours. Case #2 somewhat comes close to represent an individual level demand forecasting case study, despite having a feeder who serves hundreds of customers. This happens due to the origin of demand being very touristic-driven, mainly including restaurants, cafes, hotels, local accommodation and festive fairs.

Furthermore, different households will register electricity demand greatly dependant on the size of the buildings, number of electric installations, occupancy, lifestyle and socio-demographic enclosure, which results in load profiles drastically different. Moreover, load profiles will also rely on whether the clients have electric vehicles, solar panels or overnight storage heating. Therefore, planning the generation and distribution of electrical energy in an effective way is mainly attained through the behaviour of electricity demand in different households and communities [87].

Aligning what was previously mentioned, the first dataset used for this dissertation is characterized in the document 'Caracterização da Procura de Energia Elétrica em 2019'[3] [88]. The load profiles were elaborated from data collected on representative samples of clients covered by Standard Low Voltage energy contracts[4]. It was taken a random sample amongst the contractually connected population on the Low Voltage grid, stratified by sector of activity, class of electrical consumption and class of electrical power.

Quarterly-hour readings were made on locations subjected to an installation campaign of smart meter equipment's that assured the availability of consumption data in real time. These readings relate to a period from the 1st of October 2011 to the 30th of September 2018, counting a total of 84 months. After validation and treatment of the retrieved data, the different load profiles concerning the year 2019 were estimated based on a well-defined methodology that can be found on the document 'Atualização dos perfis de consumo, de produção e de autoconsumo para o ano de 2019' [89].

Load profiles for the 3 different classes of Standard Low Voltage, described in Table 3.1, highlight the different demand characteristics of the clients. While clients from class A belong to a more industrial type of setting due to the high contracted power, clients from class B are majorly constituted by hotels and tourism services buildings, together with residential households with a high yearly electrical consumption. Typical residential households with a contracted power lower to 13.8 kVA with a relatively low total yearly load consumption belong to C class, where the typical load day has its peak during dinner time.

All the values presented for the load were normalized, in order to eliminate the bias and to allow equative comparisons between data samples. Day number one corresponds to a Tuesday, the first day of 2019. Additionally, the original data was presented for a timestep of 15 minutes, however, for an easier understanding of the daily load behaviour, in this chapter only, the data was transformed into an hourly resolution through an average of 4 consecutive values.

This section presents a synthetic analysis of the main results obtained from the construction of the Standard Low Voltage (SLV) and the production and self-consumption units (Prosumer) load profiles. The following table presents the partition adopted to characterize different levels of customers based on the contracted power (kVA) and the yearly energy consumed (kWh).

Table 3.1 - Customers' class level

	Contracted Power (kVA)	Energy (kWh)
Class A	>13,8	any
Class B	≤13,8	>7140
Class C	≤13,8	≤7140

[3] Provided by *Entidade Reguladora dos Serviços Energéticos (ERSE)*

[4] Serving the Electricity Distribution Network Operator in Portugal

The criteria for the determination of the different classes was pre-established by the regulatory entity, not being relevant to explore with meticulous detail the benefits of such contracts, for the scope of this work.

3.1 - Standard Low Voltage 'class A' clients

Clients who belong to class A present a contracted power over 13.8 kVA and have no restraints regarding annual electrical consumption, altough complying with an yearly value of over 7140 kWh.

A simple analysis of the following figures show a clear distinction between workdays and weekends. There is even a magnitude difference in the load curve distinguishing Saturdays and Sundays. These observations are expected considering the nature of given loads, since the majority of clients from this class are companies who pay elevated contracted power tariffs. A more in-depth analysis is carried out next, where in this section the values for electric demand were normalized, in a way that the total sum of values equaled 1000, for each class.

On workdays, consumption is well represented in Figure 3.1, starting from 6h in the morning and reaching the daily peak at around 12h. Afterwards, there is a slight downfall on the load up until 14h, corresponding to the lunch break. During the afternoon, consumption stabilizes, starting to diminish considerably from 18h and in particular at 19h for the coldest months. Also, it should be expected for the load profile to reach maximum values between 15th and the 16th hours of the day, during the summer season. Generally, this load profile behaviour resembles corporate clients who justify a higher demand during work periods, and follow a significative decrease for post-work schedules.

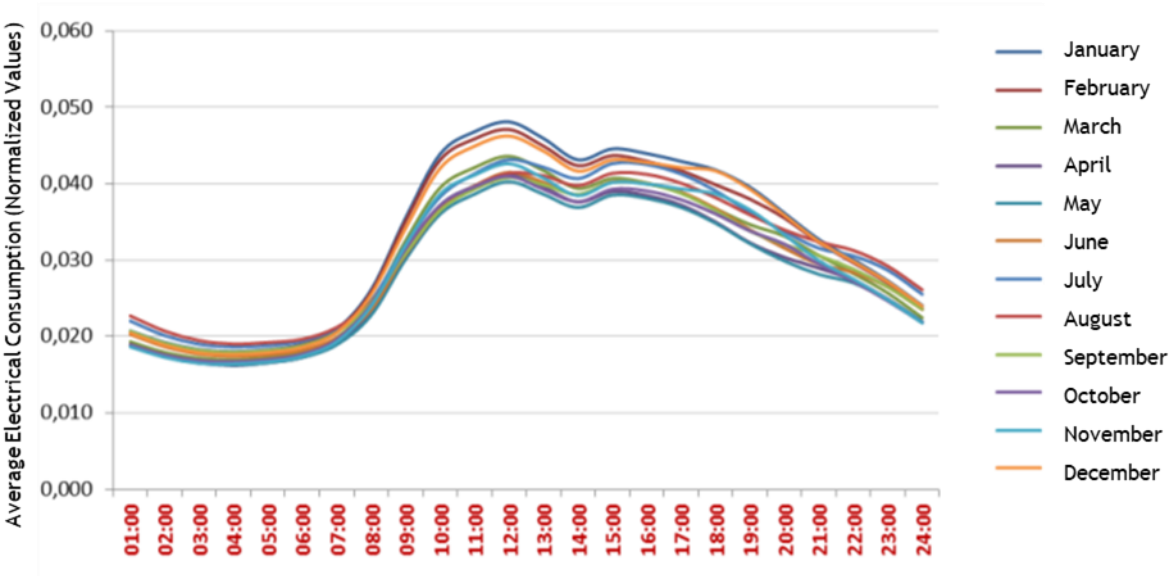

Figure 3.1 - Typical daily load profiles per month for workdays in class A.

For Saturdays, Figure 3.2 shows a load curve similar to working days, but with significantly lower values. Values start to rise around 6th hour reaching the daily maximum around 12h/13h. From there the load decreases until around the 19th hour and increase until the 20th, reaching a local maximum, concerning the coldest months. The curve decreases for the remainder of the hours.

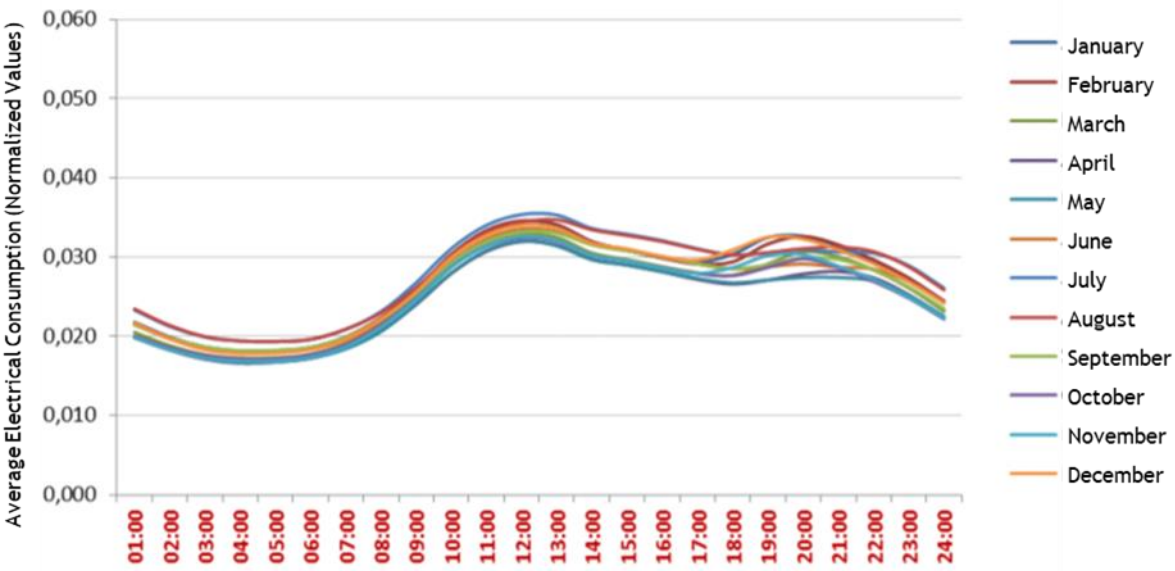

Figure 3.2 - Typical daily load profiles per month for Saturdays in class A.

On Sundays/holidays, it is notable a strong similarity with Saturday, concerning the load curve shape. However, the load weight for this day is considerably lower than Saturdays and especially weekdays. Holidays are also considered as Sundays. The overall maximum peak is reached around the 13th hour, with coldest months having a new peak around 20h, visible in Figure 3.3.

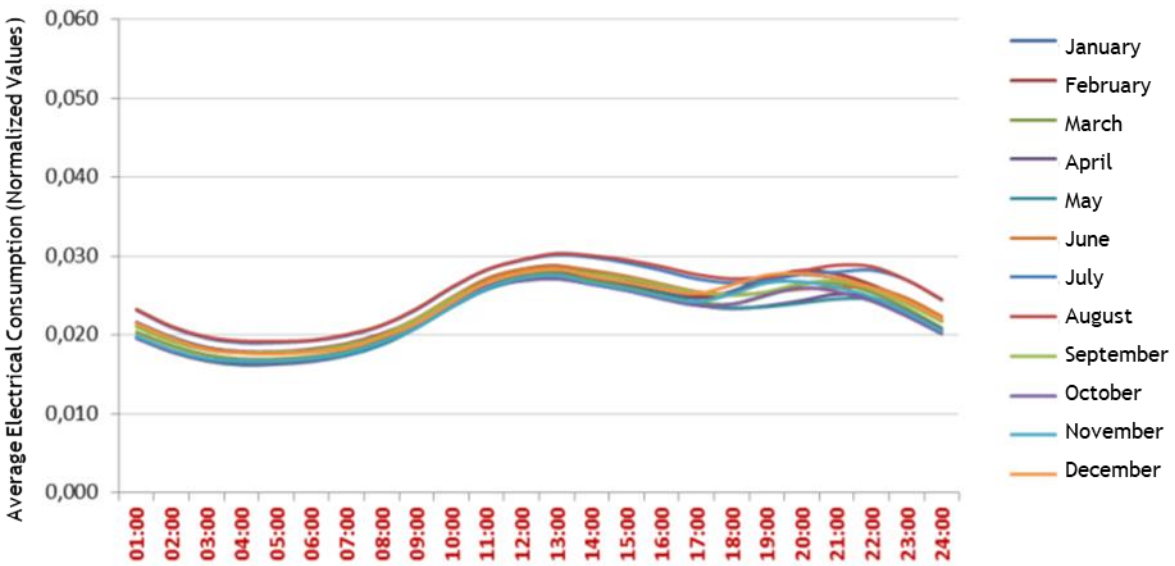

Figure 3.3 - Typical daily load profiles per month for Sundays/holidays in class A.

3.2 - Standard Low Voltage 'class B' clients

Clients who belong to class B present the following characteristics: contracted power equal or below to 13.8 kVA and an yearly consumption over 7140 kWh.

On workdays, represented in Figure 3.4, lowest consumption times are reached between 4h and 6h, depending on the month. From there, load profile rises until hitting a peak around 13h (lunch time) and presents some fluctuations between that hour and 18h. For the winter months, the profile curve increases significatively up until 20h, reaching the daily peak, and diminuishing until the end of the day. For the hotter months, the load profile curve reaches the daily peak at 13h, despite presenting a local maximum between 15h and 16h. It starts decreasing from that time until the end of day.

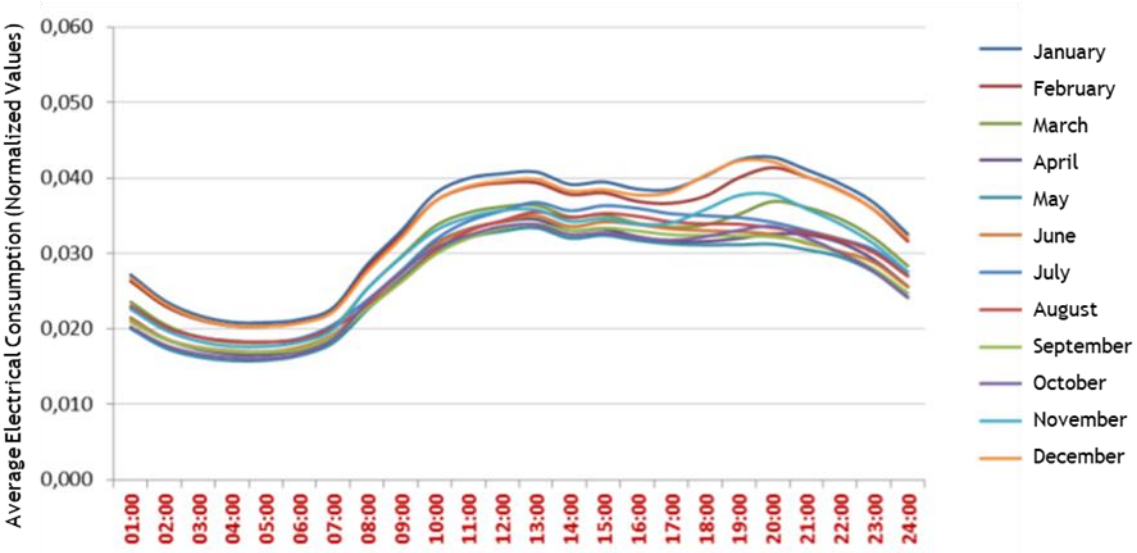

Figure 3.4 - Typical daily load profiles per month for workdays in class B.

On Saturdays, the lower values are registered at the same time as weekdays, and follow the same trends for peak hours, ocurring at 13h and later on, between 19h and 21h, corresponding to dinner time. Between April and September the daily peak occurs at lunch time, while for the remaining months, it occurs both at lunch and dinner times. Characerization of this profile is shown in Figure 3.5.

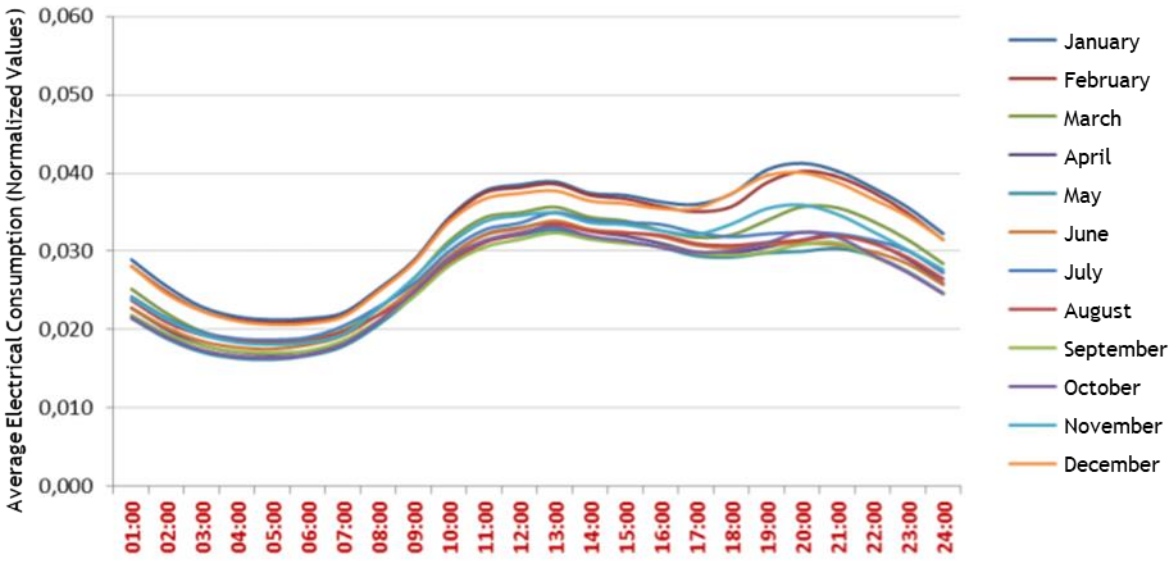

Figure 3.5 - Typical daily load profiles per month for Saturdays in class B.

On Sundays/holidays, the daily consumption peaks at same hours as the previous days, but it registers lower values comparatively to Saturdays and weekdays. The illustration of Figure 3.6 follows.

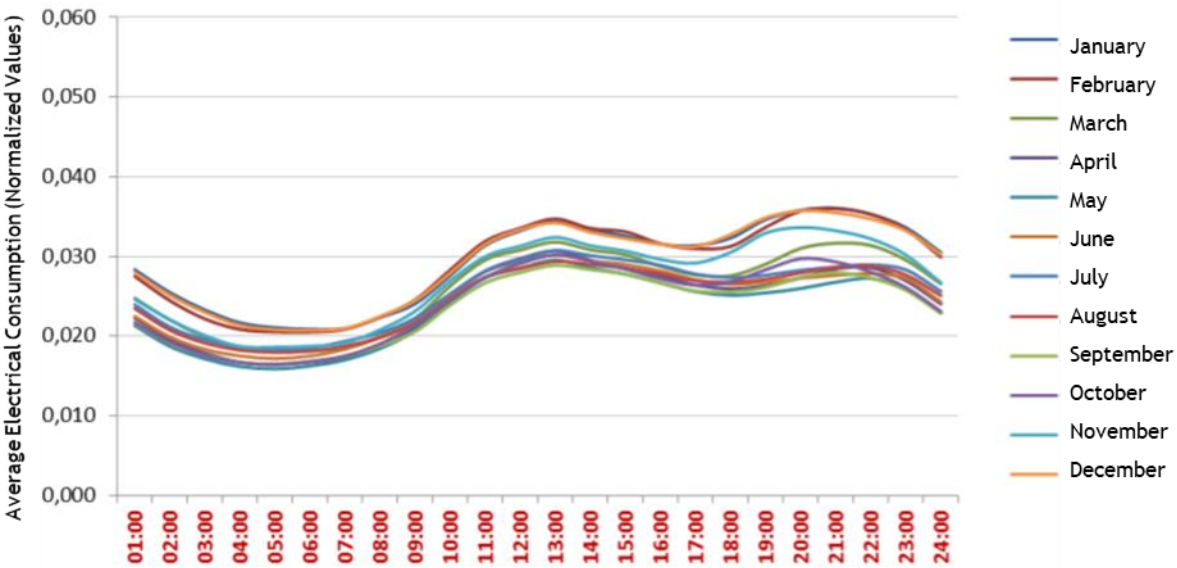

Figure 3.6 - Typical daily load profiles per month for Sundays in class B.

3.3 - Standard Low Voltage 'class C' clients

Clients who belong to class C present the following characteristics: contracted power equal or below to 13.8 kVA and a yearly consumption under 7140 kWh.

For workdays, minimum consumption values are registered between 4h and 6h, considerably increasing up until 8h. Between 8h and 13h, it grows continuously, reaching a peak in the latter. The load curve, referred as Figure 3.7, then diminishes up until 16h/17h, increasing till 21h, where it reaches the global maximum daily peak. Class C clients

differentiate themselves from the rest since they always register the maximum values between 20h and 22h (dinner time), regardless of the month or season.

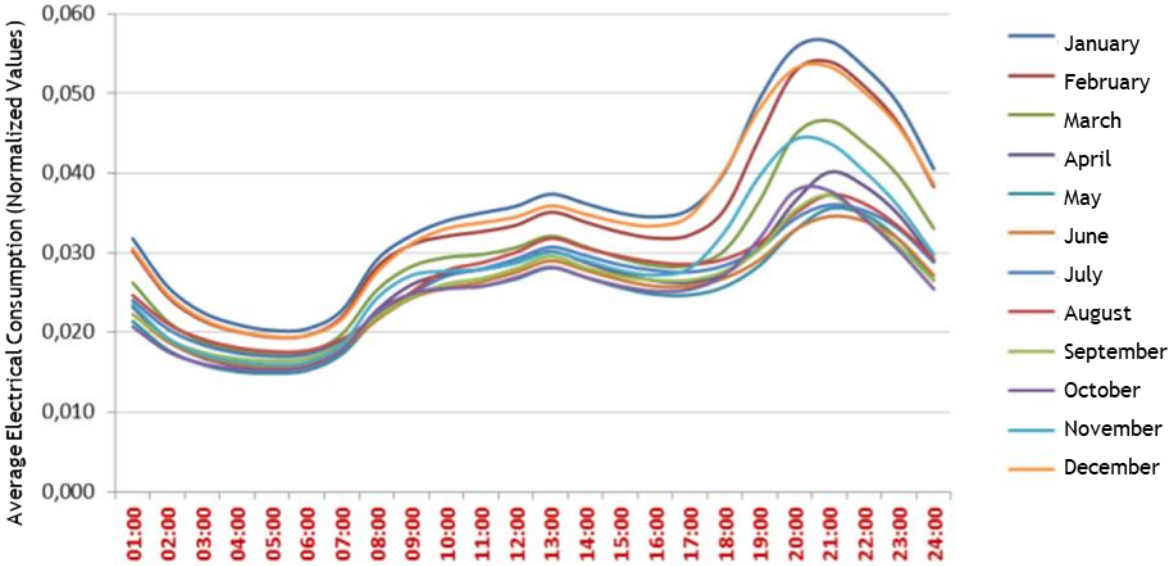

Figure 3.7 - Typical daily load profiles per month for workdays in class C.

On Saturdays, the load profile curve resembles deeply with the one regarding workdays, being Saturdays' electric demand higher for the entirety of days, with exception for the peaks occurring around 8h/9h and the one at 20h/21h, for which they present lower values. Peak demand hours do happen later for summer months, in contrast to colder months, as it is visible in Figure 3.8.

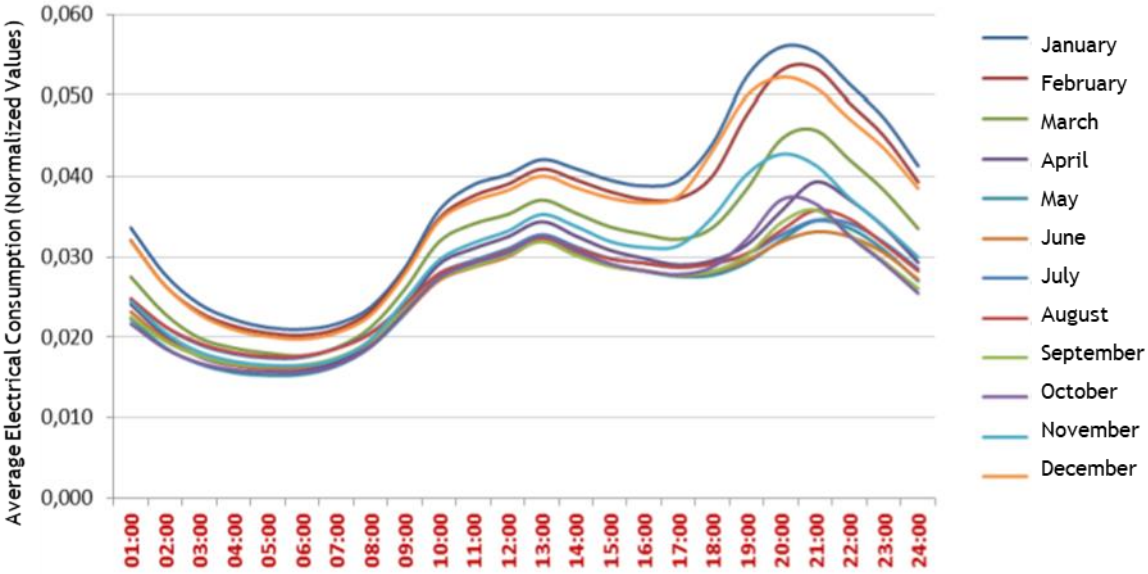

Figure 3.8 - Typical daily load profiles per month for Saturdays in class C.

On Sundays/holidays, the load profile is very similar to Saturday's, with the clear exception of the electrical demand having a greater expression for lunch times, as the peaks are clearly represented in Figure 3.9.

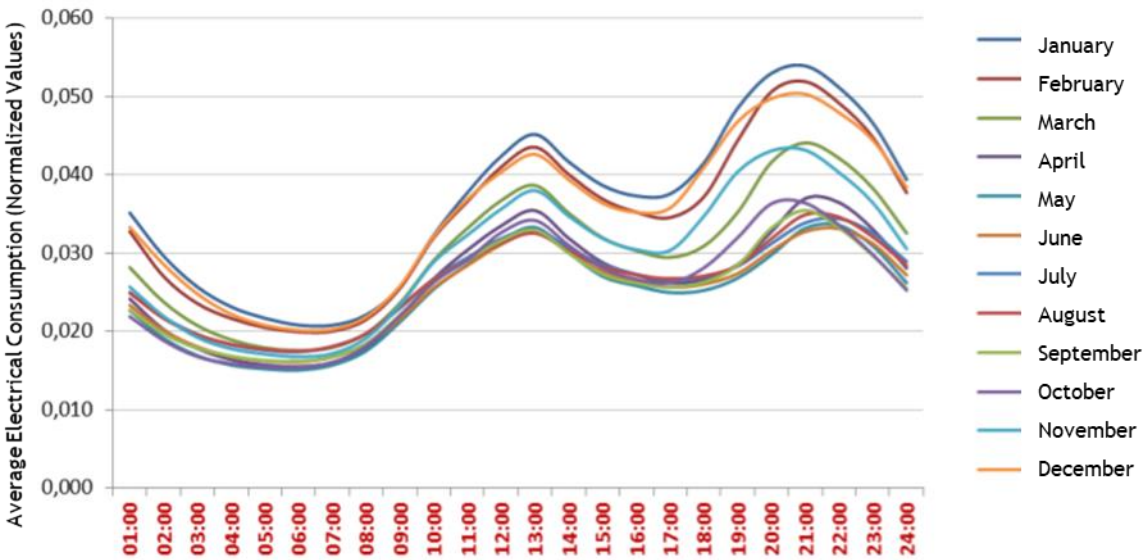

Figure 3.9 - Typical daily load profiles per month for Sundays in class C.

- **<u>Final Notes on SLV classes</u>:**

The comparative analysis between load profiles belonging to different SLV load profiles highlights that the A class is the one who presents less variation in electric consumption between winter months (December, January and February) and summer months (July, August, September). Clients from class B and C, on the other hand, present a more accentuated difference due to higher consumption in Winter instead of Summer.

Furthermore, the average consumption amplitude is higher during workdays for classes A and B when compared to class C. Additionally, while the profiles for classes A and B highlight that the average consumption is considerably higher on workdays than on Saturdays and Sundays, class C profiles showcase lower variations in such regard. These observations are expected, considering the nature and characteristics of the clients belonging to different classes.

3.4 - Production and self-consumption units (Prosumer)

All the classes in this section operate in islanded mode for some time during the day, corresponding to the most intense irradiation hours. The duration of this functioning mode is of course correlated with the period of the year, which controls the intensity and duration of the PV production hours. The PV production profiles will in most cases have the opposite form from the units who present production and self-consumption with a sales contract to the grid, such as in Figure 3.11.

Concerning prosumers[5], the comparative analysis of the yearly load profile amongst different classes highlights that the profile belonging to class A, referred in Figure 3.10, presents a more discrete variation on energetic consumption between winter and summer,

[5] Prosumer is used in the sense of being a client who has alternating periods of demanding and injecting energy into the electric grid, often having a sales contract with the utility services provider

when compared with profiles from classes B and C. The latter load profiles show a more accentuated difference between hotter and colder months.

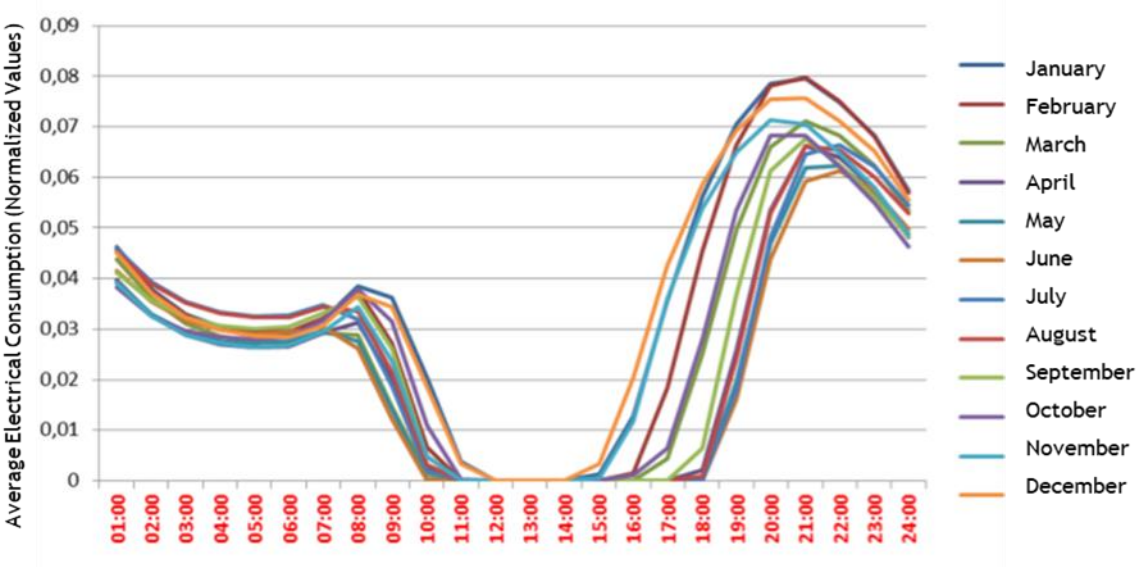

Figure 3.10 - Prosumers' monthly load profiles in class A.

The behavior of these profiles resembles the corresponding classes from SLV clients, which was expected according to the origin of construction of Prosumers' load profiles, built from the SLV different classes. However, Prosumers' installations record lower average consumption amplitude throughout the year, when compared with SLV installations. Prosumer's from class A present are not only the ones who present least amplitude of electric demand throughout the day, but also the ones who request less electrical consumption to the grid for a larger period of time.

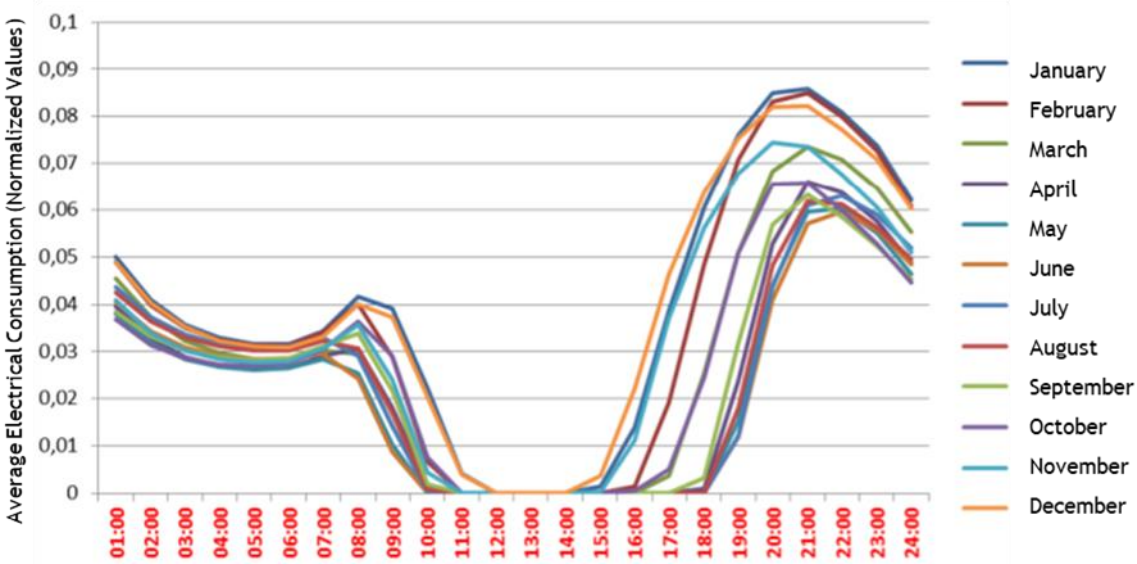

Figure 3.11 - Prosumer's monthly load profile in class B.

From class A, to classes B (Figure 3.11) and C (Figure 3.12), the amplitude of average electric demand throughout the day increases, whilst the period in which is required energy consumption from the grid, decreases.

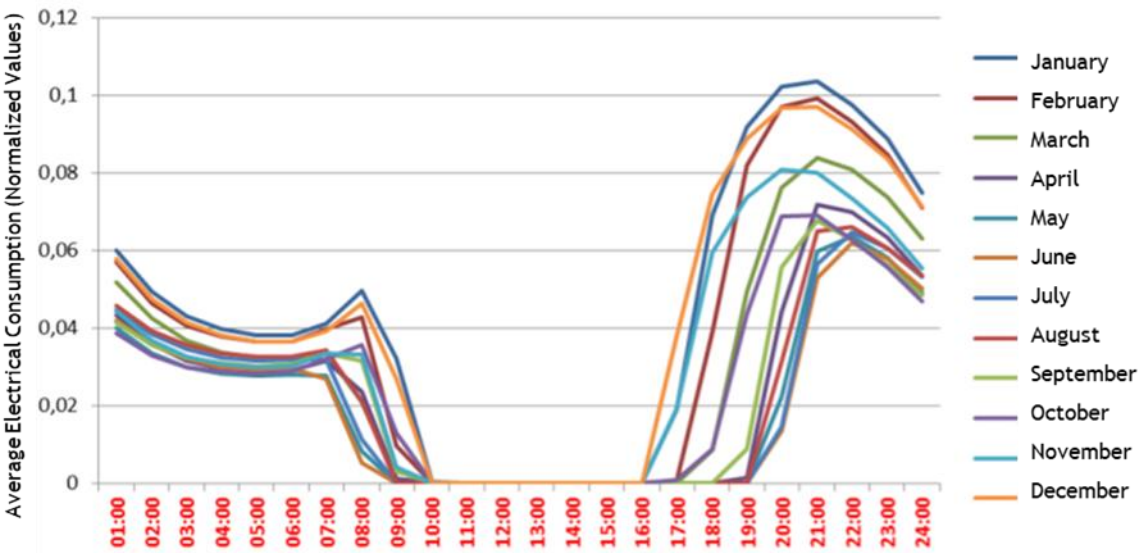

Figure 3.12 - Prosumers' monthly load profiles in class C.

3.5 - Case study #1 'Aggregated Standard Low Voltage and Prosumer Clients'

The first case study was constructed based on an attempt to recreate a realistic MG community scenario, covering the entirety range of clients contractually linked to different classes. Figure 3.13 visually represents the schematic variations and the different possible configurations that are inherent to the MG characteristics.

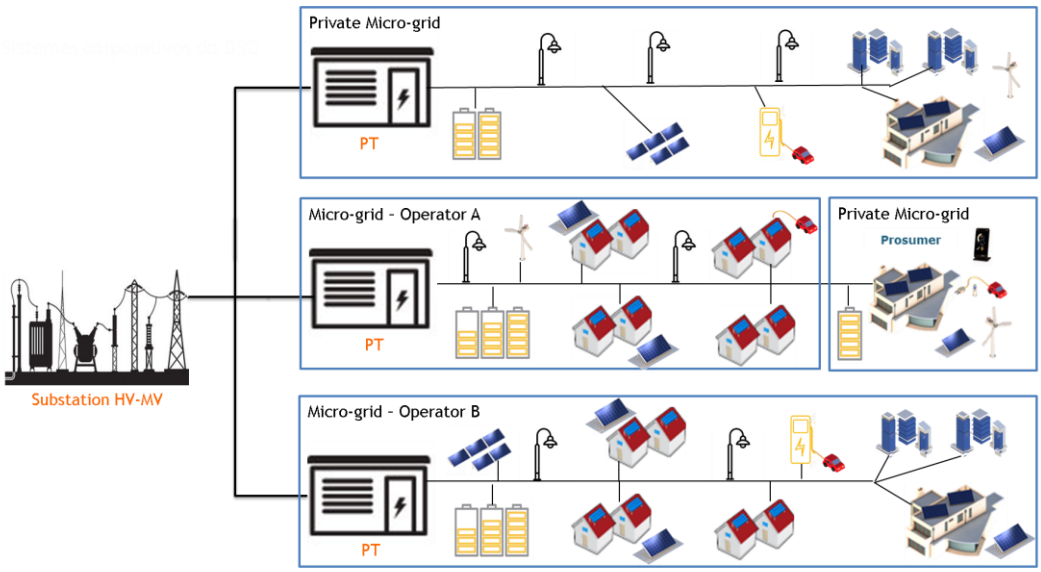

Figure 3.13 - Micro-Grid's structure from the secondary substation to the different flexible assets.

The process in order to get the electric load in kWh from previous normalized values, described in Equation 3.1, was the following:

$$ElectricLoad_{kWh} = \frac{ElectricLoad_{normalized}}{MaxSeriesValue_{normalized}} * ContractedPower_{kVA} \quad (3.1)$$

Which is then multiplied by the desired number of units.
Therefore, the constructed grid is composed by:

- 10 units with 10.35 kVA of contracted power with a yearly consumption over 7140 kWh (SLV class B).
- 15 units with 6.9 kVA of contracted power with a yearly consumption under 7140 kWh (SLV class C).
- 5 units with 17.25 kVA of contracted power with a yearly consumption over 7140 kWh (SLV class A).
- 3 units with 20.7 kVA of contracted power with a yearly consumption over 7140 kWh (SLV class A).
- 4 units of production and self-consumption with 17.25 kVA of contracted power (Prosumer class A).
- 1 unit of production and self-consumption with 20.7 kVA of contracted power (Prosumer class A).

This makes up for a total of 444.65 kVA of contracted power, in which a 500 kVA transformer to feed this community should suffice. The units of production and self-consumption account for approximately 20% of the total contracted power.

To better understand the characteristics of this case study's load profile, it was modeled in MATLAB considering several portions of data. As previously explained, the data was established for the year of 2019, which starts on a Tuesday. First plots shown concern the load profiles for the full year of data, followed by the first month and finally an yearly average of all the days of the week. Figure 3.14 presents a well-defined illustration of the load behavior, regarding the trend for 2019.

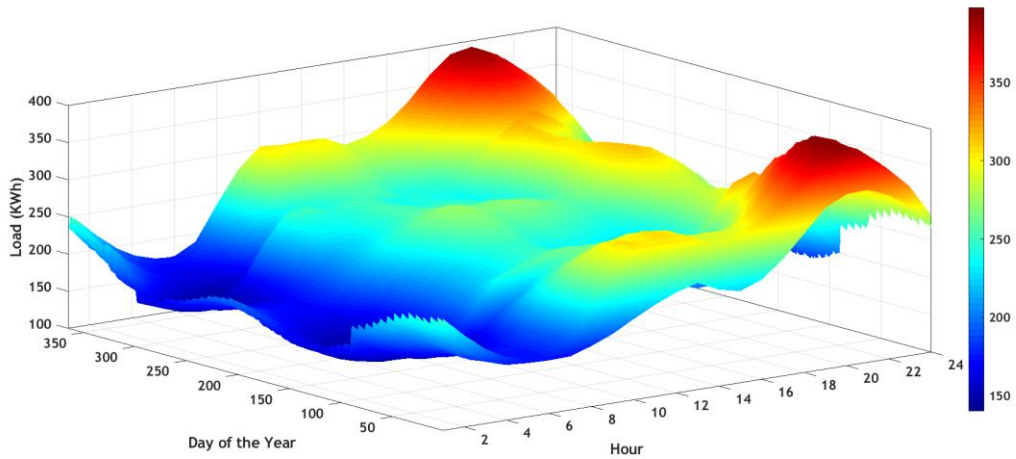

Figure 3.14 - 3D hourly modeling of the yearly load profile in kWh.

Another remark that can be held is the fact that the data is very smooth in behavior, not registering unusual spikes, flaws or missing values. The curves are well defined and do present a low degree of unpredictability. Figure 3.15 shows the same data as the previous Figure 3.14, but in this instance with well-defined timestamps, giving special attention to the load behavior

of a standard week of the year. Figure 3.15 highlights the difference between workdays, Saturdays and Sundays, and how the weekly pattern is repeated.

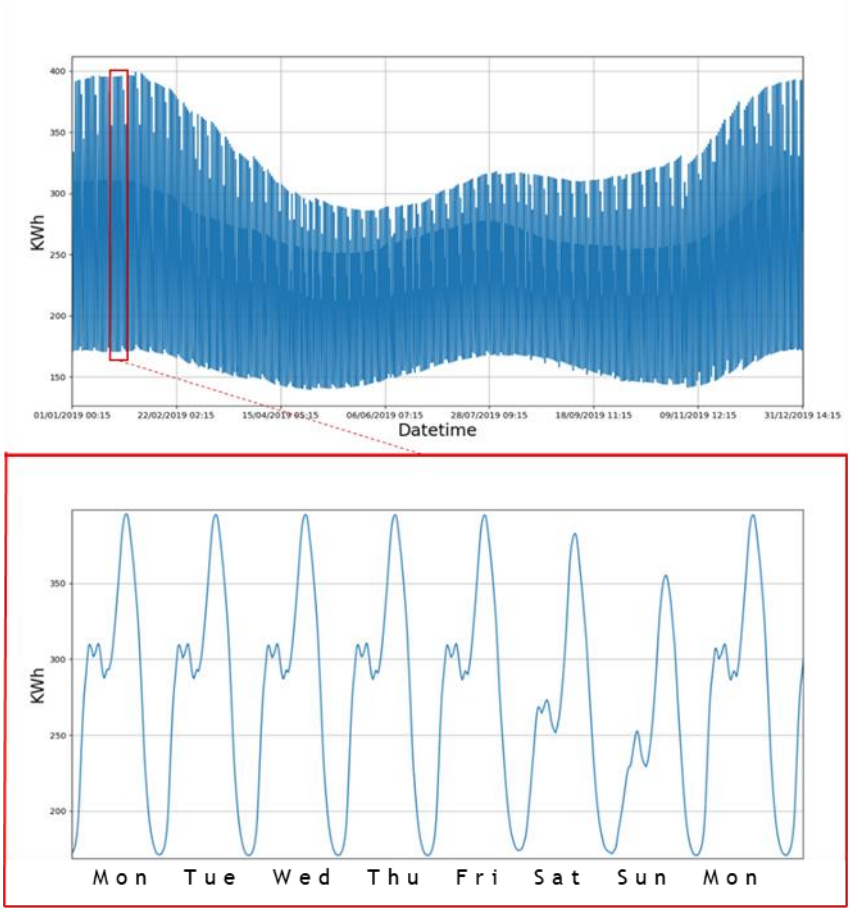

Figure 3.15 - Full year of energy usage and an expanded view of a typical week of the year with date-time stamps for a 15 minute resolution.

Figure 3.16 depicts the load profile for the month of January 2019, in which is clearly notable a strong weekly seasonality throughout the month, which continues for the remaining of the year. This means that the pattern of weekdays having higher electrical demands and weekends showing inferior values, is consecutively repeated. It is also evidently shown the daily seasonality, which leads to the repeated pattern of expected load demand for that particular hour of the day. This conclusions and regards were fundamental in terms of the strategy adopted for the LTLF (24-hour horizon).

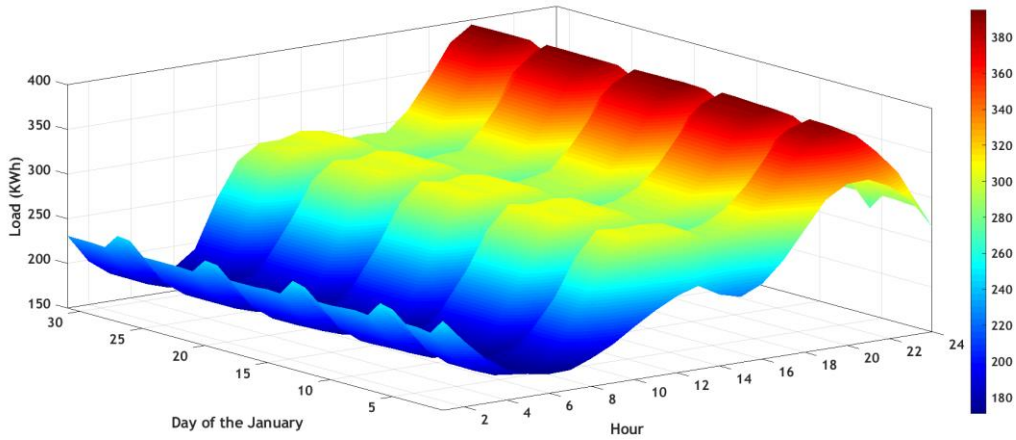

Figure 3.16 - 3D hourly modeling of January 2019.

As expected from the analysis held in previous sub-chapters, there is a clear distinction between weekdays and workdays, with the curves being similar in shape and time-of-the day related behavior, visible in Figure 3.17. This time, it was constructed average load profiles for each day of the week concerning the year 2019.

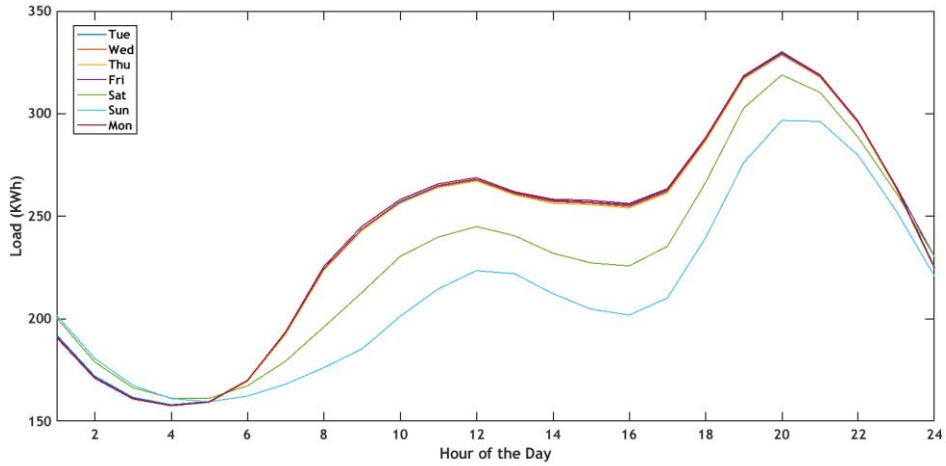

Figure 3.17 - Days of the week yearly average in an hourly resolution.

An important regard considering the nature of load is that for heating purposes, there are multiple scalable sources capable of providing thermal energy, such as gas or oil, and therefore, in many cases, Summer season could be less electrically demanding.

On the contrary, when the purpose is cooling, the energy source is mainly electrical, with rare exceptions of heat exchangers or others passive measures of heat mitigation, leading to an electrically expensive Summer.

This is not the case, however, observed in this case study. Figure 3.14 shows a tendency of higher electrical demand for colder months, instead of the expected trend of it occurring in hotter months. This could be explained through the characteristics of the climate in question, the electrical equipment's nature and the quality of Portuguese standard house construction (colder houses prepared for warm summers).

3.6 - Case study #2 'Touristic Island'

The second case study concerns a touristic island with mostly tropical warm weather. The load history record refers to the year of 2016. It is a whole different dataset, not referring to any of the data from ERSE.

There are several observations that can be held considering the load demand inherent to this particular case. The island's revenue is mostly attained through tourism, and so the local business rides the wave regarding the affluence of people, which is strongly seasonal. Looking at Figure 3.18 and Figure 3.19, it is possible to infer that this seasonality takes place during summer months, particularly July, August and September. With all the local services (restaurants, cafes, hotels and shopping's) being directed to this specific period, it was to be expected this yearly load profile. Not only that, but having a tropical climate mostly dismisses the need for heating during colder months, and accentuates the necessity of cooling during Summer, along with the enormous amount of people checking into the island for that period.

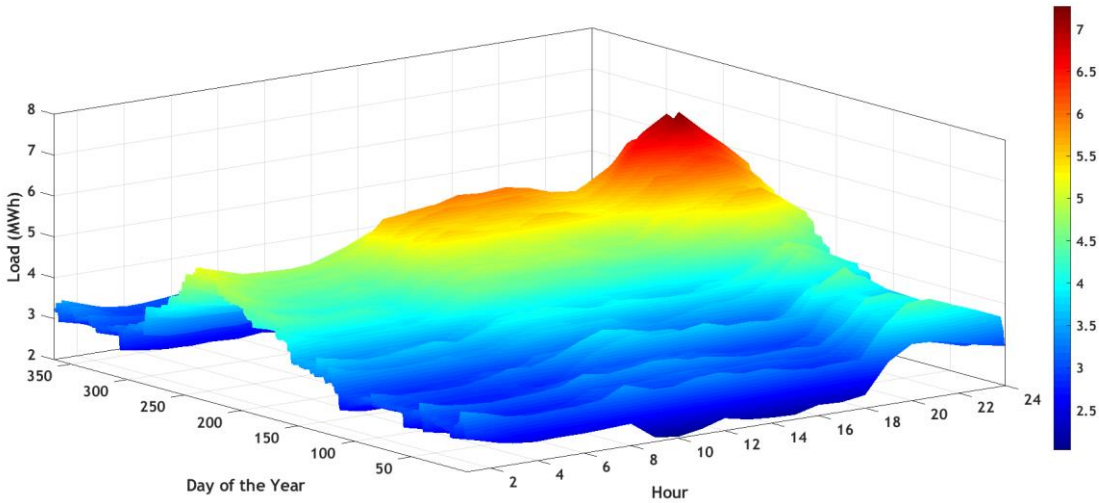

Figure 3.18 - 3D hourly modeling of the yearly load profile in MWh.

A quick look at Figure 3.19 incites that the curves present spikes who are hard to generalize and predict, contrary to the previous case study, where the curves are smooth and well defined for most instances. This shows to be a good proving ground for forecasting applicability, with data hard to generalize or predict.

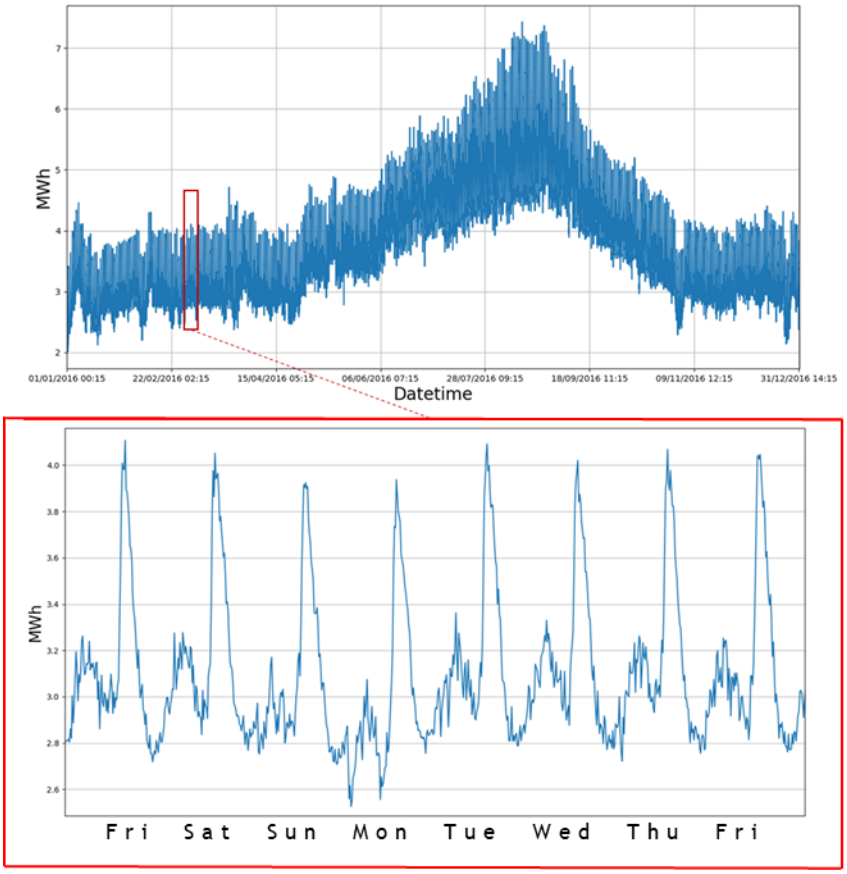

Figure 3.19 - Full year of energy usage and an expanded view of a typical week of the year with date-time stamps for a 15 minute resolution.

A brief analysis on Figure 3.20 indicates a strong variability and unpredictability in terms of day of the week. There are no strict weekly patterns nor differentiation between workdays or weekdays. There is, however, always displayed a high dependency regarding previous day or week same hour loads, thus inciting the use of possible lag variables as features for the building of the regression predictive model.

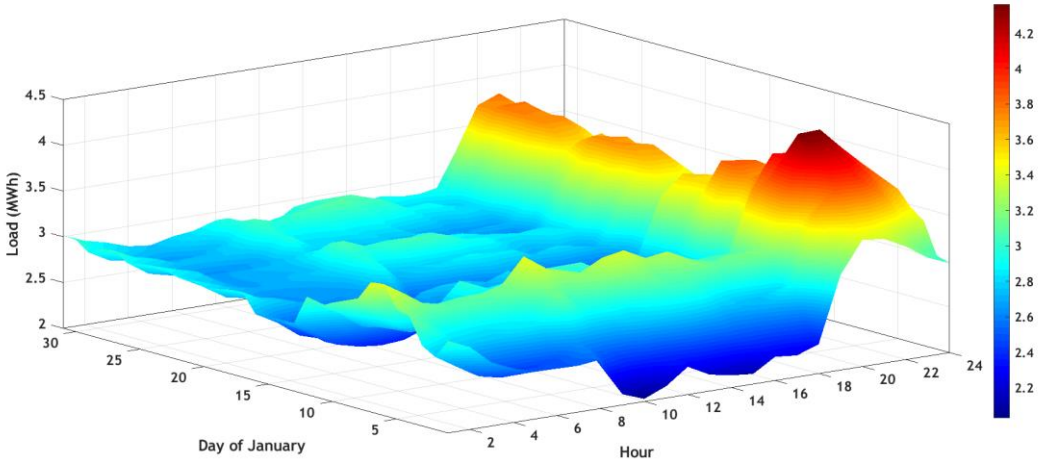

Figure 3.20 - 3D hourly modeling of January 2016

The construction of seven load profiles shown in Figure 3.21, regarding the different days of the week, led to conclude that there is not in fact a clear distinction between workdays or weekdays. Mondays are the ones who register lower load demand values during workhours, since it is probably the weekly day break for local businesses and services. Sundays' follows the same logic as Mondays' and therefore present identical values. Once again, the electrical demand is intensely dependent on the people's mass traveling to this island, being a preferred place for just holidays.

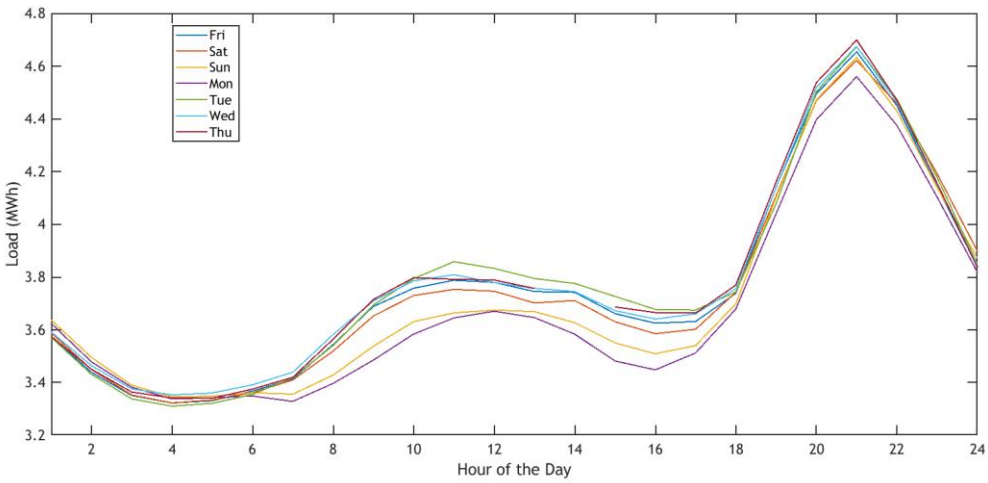

Figure 3.21 – Days of the week yearly average.

The inclusion of this case study under the scope of data analysis was made considering that this dataset better represents the variability and unpredictability of electrical demand inherent to a MG, and poses a deeper challenge concerning 24-hour load forecast, which is ultimately the proposed objective for this dissertation. The results obtained in Chapter 5, who will deserve a more detailed attention and interpretation, give more strength and reasoning to the creation of this new case study.

Chapter 4

HALOFMI Methodology

The developed HALOFMI (Hybrid Approach to LOad Forecast in Micro grIds) methodology seeks to provide a resourceful forecasting tool with real-time application for the management system to plan the operation of generation, performing a forecast for the next 24-hours. The recreated case studies aimed at recreating MG scenarios associated with the presence of uncertainty both in production, due to the increasing penetration of RES, and in electric demand, regarding the variable flexible assets and the electrification of vehicle transportation and heating solutions.

Moreover, it also considered the different load fluctuations in terms of size and nature of electric demand, with the grid having the capacity of expanding and allocating more and diverse customers, or shortening to a lesser amount, meaning that the historic load profiles could be drastically different from within consecutive years, or even months. Having a predictive model that yields good performances is of major importance, along with the system's allocated capacity of storing energy, to decide the optimal strategy for the allocation of controllable resources.

Therefore, the proposed strategy not only selects the optimal subset of features for each day of forecast, but also allows to give insights on the appropriate amount of days given for training, from within the full available dimension of data, considering the characteristics of the target (the day intended to forecast).

Finally, the chapter structure is based on the HALOFMI methodology, with a flowchart illustration guiding though the timeline processes. The subchapters correspond to the detailed explanations composing the various steps of the methodology, and the way the 24-hour forecast is attained.

4.1 - Description of the HALOFMI Methodology

Figure 4.1 represents the designed methodological steps in order to achieve a robust forecast, yielding MG real-time applicability. The strategy aims to mitigate the variability and unpredictability effects of the load's electric demand, by incorporating a multi-stage

operational algorithm and consequently attempts to improve the performance of the forecast. The flowchart from Figure 4.1 demonstrates an ongoing tool that forecasts full days of electric load forecast on demand.

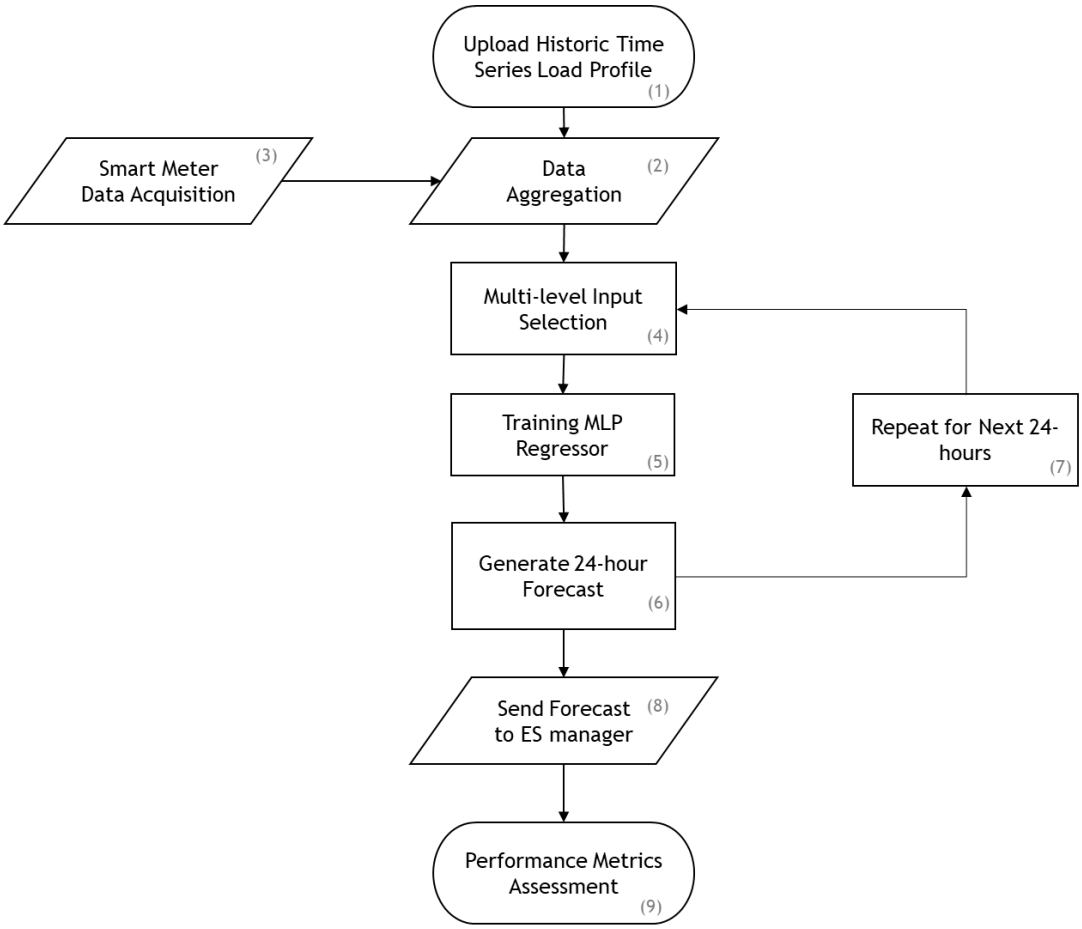

Figure 4.1 - HALOFMI Methodology Flowchart.

The process begins with the data upload, containing an historic time series load profile consisting in step (1). Then, in step (2), it is aggregated with live electric load data information coming from the Smart Meter Data Acquisition, in step (3).

Step (4) is of crucial importance and has a sub-chapter dedicated to a deeper analysis which results in important conclusions. It is in this phase that the inputs are created, and afterwards selected. The first selection is filter-based, using statistical correlation analysis. The second selection is attained through ML techniques.

Step (5) is where the forecast is performed using the Train-Test split defined for the last day of intended data interval. The elaborated function that partitions the training and the testing, starts from preprocessing the input and the output (original target data), by scaling each instance between the range of 0 and 1. Afterwards, it aggregates every 96 instances of 15 minutes (24 hours) in a day, traversing for each day of the year. From here, the array of days is created, and the day of testing is assigned to be the last element of this group. Then, the system receives the number of days to apply the sliding window, reducing the dimensionality of the attributes, avoiding excesses of computational weight and long processing times. It is expected an increase in the size of data over time.

It is also in step (5) that the model is built via supervised learning, which then advances to step (6) to generate the daily forecast.

The management system then receives this information of data and plans the operation control for the next day of electric demand. The results are then assessed, and performance is evaluated using the metrics defined in Chapter 2. This process is repeated for each day of the year, where in step (2) the data is constantly fed with new information by the installed smart meters, updating the data regarding on-demand electric load, depicted in step (3).

4.1.1 - Multi-level Input Selection

Represented in the flowchart as step (4), is one of the most fundamental steps of the methodology, the creation and selection of features for the model training.

Being such an important process, there were some additional data considerations that were made in order to make the best strategic decisions:

1. Whether or not to incorporate external data, and at which extent could these exogenous variables positively impact the model's performance. At first, temperature was included as a feature, due to several published papers referring to it as a crucial factor to boost the accuracy of the model. Despite this fact, after compiling the regression and correlation analysis, results showed insignificant influence on values between the load target and the temperature. Two distinct factors can help explain these results. Firstly, considering temperature as the major contributor to the electric load demand could lead to gross errors, as this could be sometimes the case, but not always. For interior spaces heating for example, irradiation, humidity and especially internal loads (number of people occupying the place, electric utilities working, number of computers) can significantly alter the electric heating demand. Secondly, having reliable and quality historic temperature data profiles regarding the specific geographic locations, showed to be an additional difficult problem to handle. The situation will not endure, however, with the advancements of smart cities, monitoring consistently more reliable atomized data. Therefore, load to temperature correlations are hard to attain and to enable extracting concise conclusions about its influence at the MG overall electric load, which could hurt the training of the model.

2. Whether or not to keep as univariate or multivariate, how many features and which ones. The decision to build a multivariate model relied on the enhancement of information that several features could have on the target, such as data from the previous days at each hour. Additionally, and in regard to adding exogenous variables, it was decided that constructing a model containing solely features composed of lagged variables would contain the implicit influence of external variables such as temperature, irradiance, precipitation levels, humidity or day of the week characterization. Thus, the model is exempted from the necessity of containing these additional features.

3. The necessity to clean the data. The outliers were detected, removed and was created a process for missing values imputation. For the case being, there were

several blackouts in the data, corresponding to periods of electrical shortage. To solve these issues, while the values presented out-of-the box magnitudes (extremely low or extremely high), the outliers were assumed to be an average of the 2 instances happening before and after the occurrence.

After these regards, and the data analysis performed in Chapter 3, there was a strong solidified motivation to use the previous electricity loads to predict future loads. Therefore, in order to better capture the daily and weekly patterns and to extract the useful features, it was applied an autocorrelation analysis to the data.

The lag autocorrelation coefficient r_k indicates the linear correlation of the time series for times t and t-k. With k being the autocorrelation order, X_t being the value of the time series at time t and $\bar{X}$ being the mean value, comes the following Equation 4.1 [90]:

$$r_k = r(X_t, X_{t-k}) = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=k+1}^{n}(X_t - \bar{X})^2} \tag{4.1}$$

Consequently, r1 measures the linear correlation between the time series values that are 1 lag apart, while r2 measures the linear correlation between the time series values that are 2 lags apart, and so forth. The returned values form an autocorrelation function, which bears the purpose of better portraying the cyclic patterns of the time series load data.

Figure 4.2 shows the original series and consecutive differenced series and its respective autocorrelation function for the electric load demand concerning the year of 2019. Each lag represents a 15-minute reading and the data sample accounts for 35040 instances, corresponding to a full year of data. As it is possible to observe, the data is strongly correlated, with the previous days same hour instances being the most relevant ones. Also, values closer to 1 show a strong positive autocorrelation while values close to -1 indicate a strong negative autocorrelation and values close to 0 indicate an absence of autocorrelation.

Additionally, it is important to note, both in Figure 4.2 and Figure 4.3, the increasing trend of the blue cone visible in the original series autocorrelation function, which growth represents the decrease in confidence intervals. This means that the bigger the area of the cone, the lesser the significance of the lag variable. By default, this is set to a 95% confidence interval.

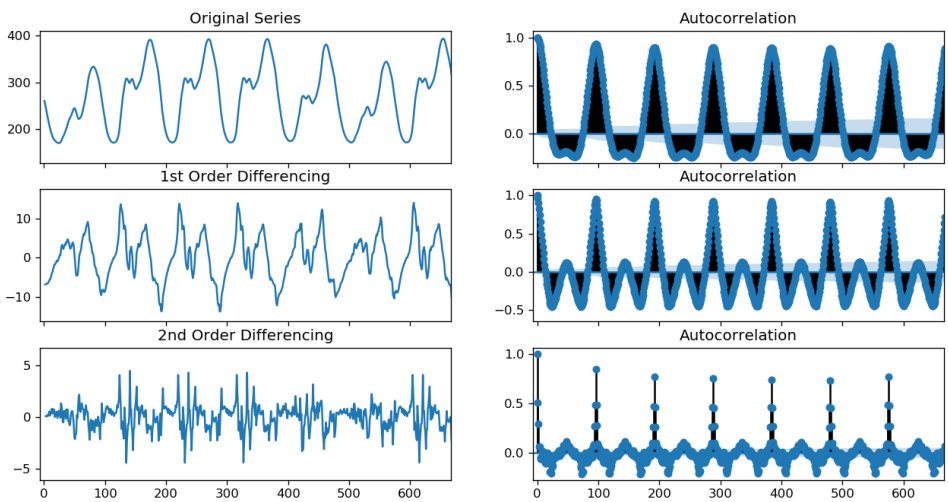

Figure 4.2 - Case Study #1 time series Differencing and respective Auto Correlation Function (ACF).

From the ACF plots it is clear to see the lagged results peak consecutively at a constant interval. This means that relevant lag values occur periodicity every 96 time-steps, which corresponds to a full day (96 intervals of 15 minutes). It is also possible to observe what would apparently be the strongest correlation with the original time series. This peak happens after 672 entries, corresponding exactly to the same day, same hour, previous week load. These results are also very satisfactory considering the expected strong seasonal pattern of the load, with the weekly period being the most relevant one. The load profile is therefore considered to be a cycle where characterization of the day of the week can lead to better results in terms of feature selection to the predicative model.

Also, applying a 2nd order differencing to the original time series from case study #1 opens way for a clearer interpretability in terms of strongest autocorrelation values, amongst the strongest ones shown. A very significant correlation, apart from the first peak which relates with itself, is observed concerning the same hour previous day load. This matches the expected results since Friday's same hour load relates with Thursday's, and so forth, except for Monday's, who have a lower correlation with Sunday's, but still follow somewhat the same curve characteristics. Sunday's same hour load also relates with Saturday's for the same period, whilst Saturday's differs in some way to Friday's daily load, due to the distinction of workdays/weekdays.

From this analysis it seems clear why methods who use persistence technique are based on observations of previous day loads to forecast the hours ahead, which means they can either provide reliable forecasts, depending on if the previous loads are similar to the present ones.

A similar analysis was conducted for the case study #2, which portrays similar results in Figure 4.3. It is also clear to observe that a 2nd order differencing was unnecessary, since the 1st order differencing successfully eliminated the trend and cyclic patterns, as well as highlighted the most significate spikes. For this case, there are no visible distinctions between the strongest autocorrelation lags, which reinforces the use of the second part of this Multi-Level Input Selection: a feature selection reduction method.

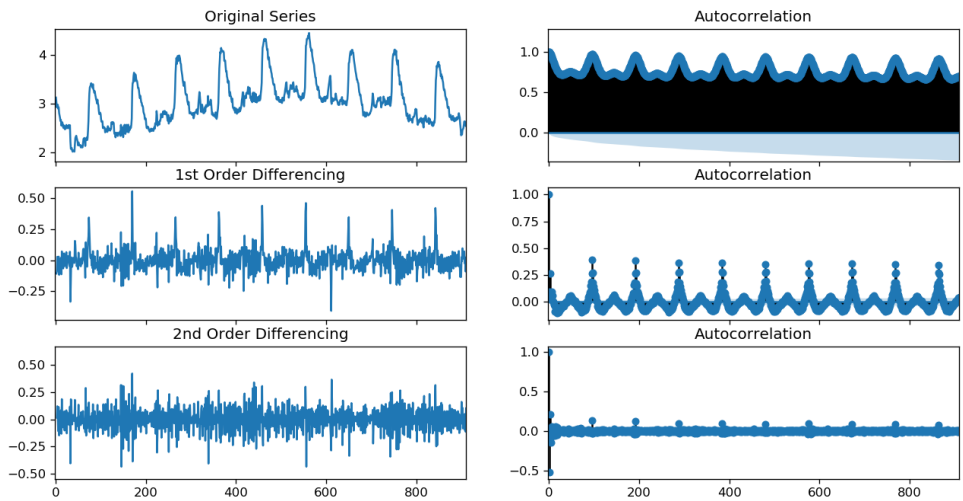

Figure 4.3 - Case Study #2 time series Differencing and respective Auto Correlation Function (ACF).

As part of the adopted strategy, it was created a new file with the possible features to be selected, only considering the lags that happen at the same hour up until two weeks of data.

More than two weeks was considered to have little significance on the target data. Such process implies deleting data equivalent to 2 weeks, 14 days, 336 hours or 1344 cells (considering a time step of 15 minutes for a time span of 1 year), so that all the lagged features derived from timestamps have the same dimension. This amount comprises 3.8% of the total data analyzed, an insignificant amount and therefore unimportant for the result of the forecast. This process is illustrated in Figure 4.4 and Figure 4.5. As a result, 14 features are selected, corresponding to 14 days.

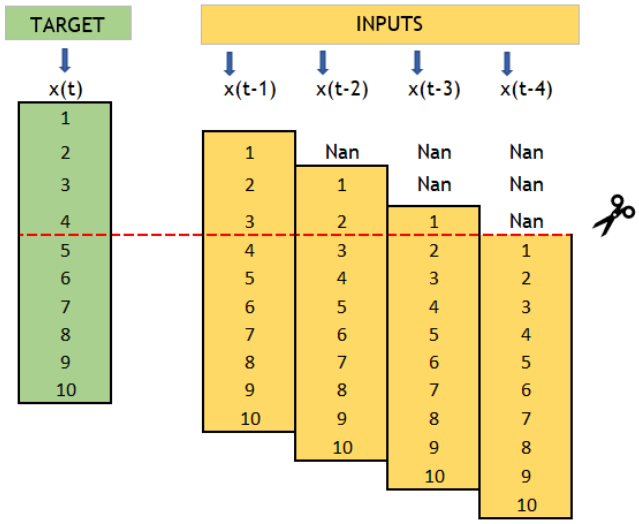

Figure 4.4 - Adopted strategy to adjust data size.

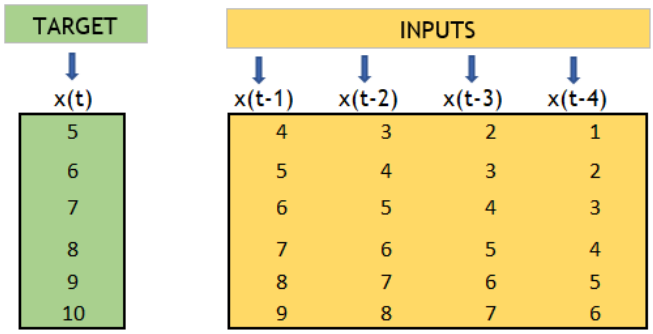

Figure 4.5 - Visualization of data size adjustment results.

After selecting the subset of 14 features, it is important to retrieve meaningful entries to the model, for both enhanced efficiency, error minimization through maximizing the chance of subsequent validation and to lighten the computational weight.

Therefore, arises the need to perform some method of feature selection. Feature selection methods reduce the number of features by eliminating the ones that present redundant information or selecting only the ones who show the highest weight vector value.

Typically, there are three distinct approaches that use three categories of methodologies: filters, wrappers and embedded solutions. Figure 4.6 illustrates the working mechanism behind each one. Filter methods perform the filter selection regardless of construction of the regression model. Wrapper methods iteratively select or eliminate a set of features using the

prediction accuracy of the prediction model. For embedded methods, as the name suggests, feature selection is an integral part of the regression model [91].

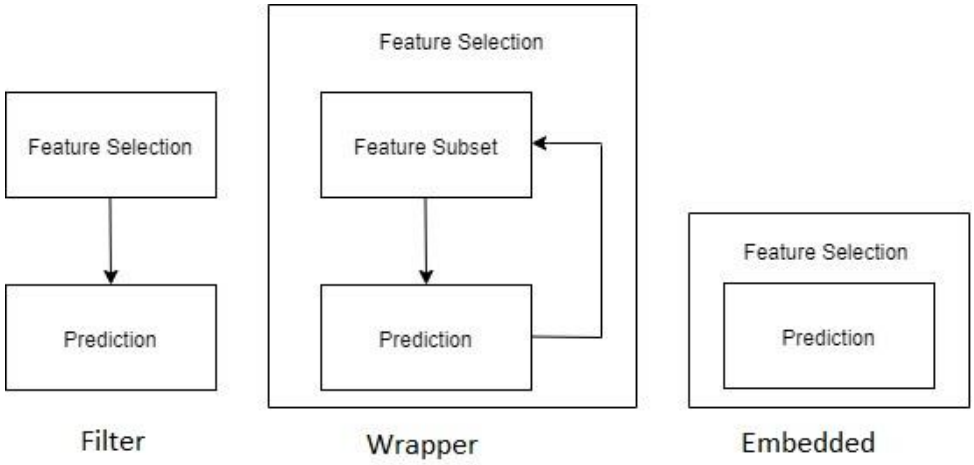

Figure 4.6 - Feature Selection categories.

Therefore, for the scope of this work, it was decided to adopt a hybrid approach consisting of a 2-phase process: at first, a gross filtering of features based on the statistical analysis through autocorrelation-based feature selection (ACFS), performed previously in this chapter.

The good features subset contained data with a time step of 24 hours for 2 weeks. This autocorrelation analysis falls into the filter method of feature selection. Afterwards, this subset goes through a second selection-based wrapper method, namely the Recursive Feature Elimination (RFE) who eliminates worst performing features on a particular model one after the other, until the best subset of features is known.

For this particular data, 14 features (attributes who are selected to better explain the forecast target) are selected. In each round, '14 minus the number of the round' models are created with combination of all features except one. The least performing feature, assessed by the lowest importance score, is removed. This process goes on until only one feature is left for testing. With this, the method returns the optimal amount, and which features to use for training. In Figure 4.7, it is confirmed that for the case study #1, the same day previous week feature is the one with the highest weight vector, followed by previous two weeks and finally the day before. Gradient Boosting Regression was the estimator used for this ranking.

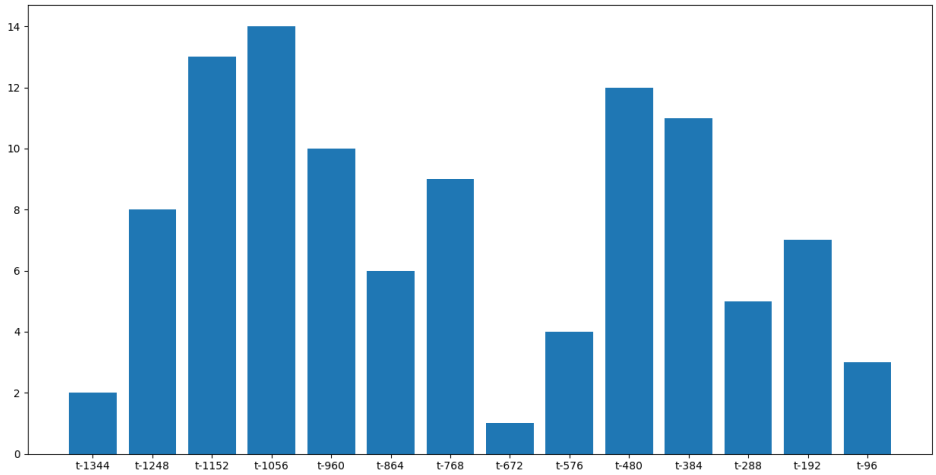

Figure 4.7 – Illustrative result of the process of ranking features from the importance score.

Wrapper methods such as RFE deliver an optimal set of features through a greedy search algorithm, who operates by making the locally optimal choice at each stage, with the goal of finding the global optimum.

For this segment, three estimators were applied to sequentially evaluate the best performing features subset: Gradient Boosting Regression (GBR), Random Forest Regression (RFR) and the Linear Regression (LR). The decision to eventually choose the Linear Regression as an estimator for the RFE process was made considering that the results did not varied significantly, while the computational time decreased immensely, compared with the other two. Wrapper methods can be computationally heavy, like the SVR, who returned the results in approximately 18 hours. In conclusion, for a simulation standpoint, the LR model benefits more with the use of RFE compared with the remaining subjected to testing. Also, it is expected a significant decrease in computational processing times after the application of the method exposed in Chapter 4.2, regarding the dimensionality reduction. With this enhancement, the use of LR as an estimator for the feature selection model is excluded, and for the practical case, ML algorithms are used instead, such as the GBR and the RFR.

Moreover, RFE was implemented using a CV technique, resulting in training several ML models on subsets of the available input data and evaluating them on the complementary subset of data. This prevents the data to over fit, thus maintaining its ability to generalize.

This method was implemented in Python, importing the RFECV (Recursive Feature Elimination with Cross Validation) function in the framework of *Scikit-Learn*. A stratified K-fold was also used, with the attributed number of folds to be K = 10.

Regarding the case study #1, the illustration of the RFECV process with different estimators is presented in Figure 4.8, where GB optimally selects 3 features, RF selects 11 features and LR selects 14 features. From more than 6 features, the differences in each cross-validation score is almost negligible, as it can be visibly seen.

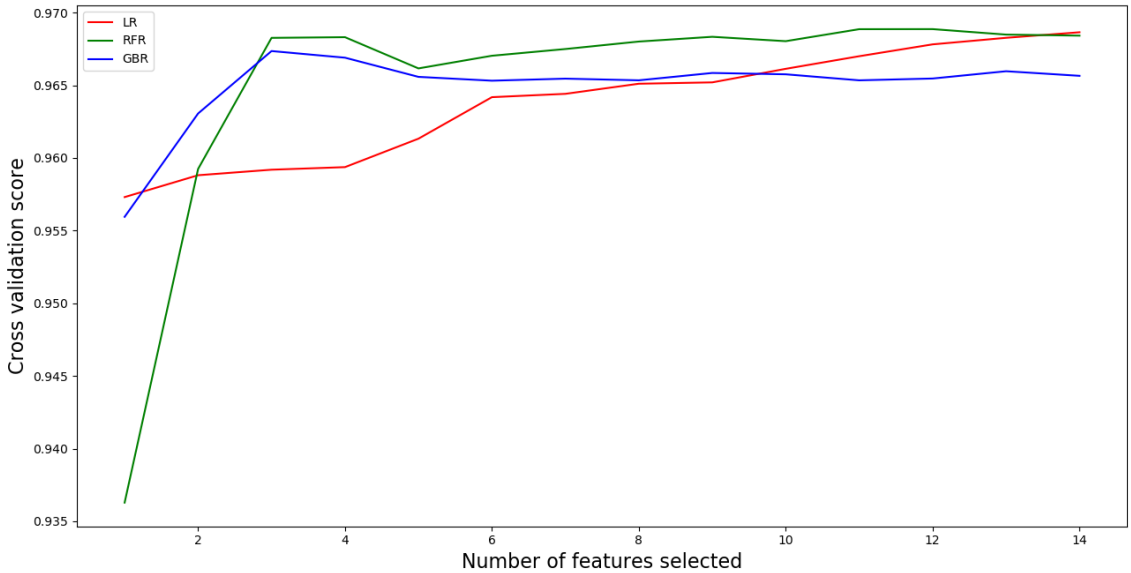

Figure 4.8 - Recursive Feature Elimination results with 3 different estimators for case study #1

Additionally, these results show that LR has a diligent behaviour in terms of computational speed, but it does not show to be very 'smart' in terms of learning from the data, since it still didn't clearly stabilized, even with 14 features. The other two estimators learn much faster, and with this wit, can quickly adapt to new scenarios, requiring less features lo learn with the same efficiency. This shows the power and insight of ML models. Again, the choice for LR fell due to the lack of time given the necessity of running an immense number of tests. Overall, GBR and the RFR can lead to better performances when having access to less information.

4.1.2 - Training MLP Regression

The adopted ANN architecture model, used as part of the developed methodology, was the Multi-Layer Perceptron (MLP), whose structure and processual operations was already dissected in Chapter 2. This model was implemented once again through *Scikit-Learn*, using the *MLPRegressor* function, which trains using backpropagation with no activation function in the output layer.

MLP uses supervised learning to train a function $f : R^m \rightarrow R^o$ on a dataset that contains *m* number of dimensions for inputs and *o* number of dimensions for outputs. The attractive attribute of this algorithm is that given a set of features $X = x_1, x_2, \dots, x_m$ and a target *y*, it can approximate a non-linear function for regression through the non-linear layers, called hidden layers.

In regression, MLP uses the Square Error loss function, expressed as in Equation 4.2, whereas the output activation function is just the identity function.

$$Loss(\hat{y}, y, W) = \frac{1}{2}\|\hat{y} - y\|_2^2 + \frac{\alpha}{2}\|W\|_2^2 \tag{4.2}$$

where $\hat{y}$ is the forecast value, y is the real value, $\frac{\alpha}{2}\|W\|_2^2$ is an L2-regularization penalty term that is responsible for penalizing complex models and $\alpha > 0$ is a positive hyper-parameter that controls the penalty's magnitude.

Additionally, these weights are repeatedly update, thus minimizing the loss function. After this loss is computed, it is back propagated from the output layer to the previous layers, feeding individual weight parameters containing a value that aims to decrease this loss function.

For the gradient descent, the gradient of the loss $\nabla Loss_W$ concerning the weights is calculated from W. The formulation is presented in Equation 4.3:

$$W^{i+1} = W^i + \epsilon \nabla Loss_W^i \tag{4.3}$$

where i is the iterative step, and ϵ is the learning rate (always larger than 0).

The algorithm either stops when it reaches the maximum number of iterations, defined from the Grid Search explained below, or when the loss improvement is insignificant.

Succinctly, the advantages of the MLP include the capability of learning non-linear models, and a very useful tool that enable models to learn in real-time (on-line learning) using the *partial fit* function.

However, MLP also bears some disadvantages, such as being sensitive to feature scaling, and requiring 'manual' adjustments in terms of tuning some hyper-parameters, i.e. number of hidden neurons, layers and maximum number of iterations.

Therefore, in order to solve these limitations, the dataset values were normalized from a 0 to 1 scale in the Tran Test Split designed algorithm and it was implemented a Grid Search with Cross Validation to find the best hyper-parameters, amongst the ones given. Grid search performs an exhaustive search over a specified parameter grid containing values for an estimator. Also, this search is optimized using cross-validated grid-search over the parameter grid.

For the case being, it was created a parameter grid composed of 3 options for the number of hidden layers and neurons consisting in 2 layers of 30 and 20 hidden neurons, 2 layers of 20 and 10 hidden neurons and 3 layers of 20,10 and 5 hidden neurons. The number of hidden layers and neurons were set according to the rules-of-thumb studied in Chapter 2. Moreover, 3 options were set for the activation function and 3 options for the number of maximum iterations (500, 750 and 1000). *Adam* was the solver used for training and the learning rate was set to be adaptive.

Before initiating the training, it was created a routine of data segmentation, resulting in a walking-forward and a sliding window method comparison. The designed flowchart is presented in Figure 4.9, and a more exact description for the strategy and reasoning behind the implementation of this method follows:

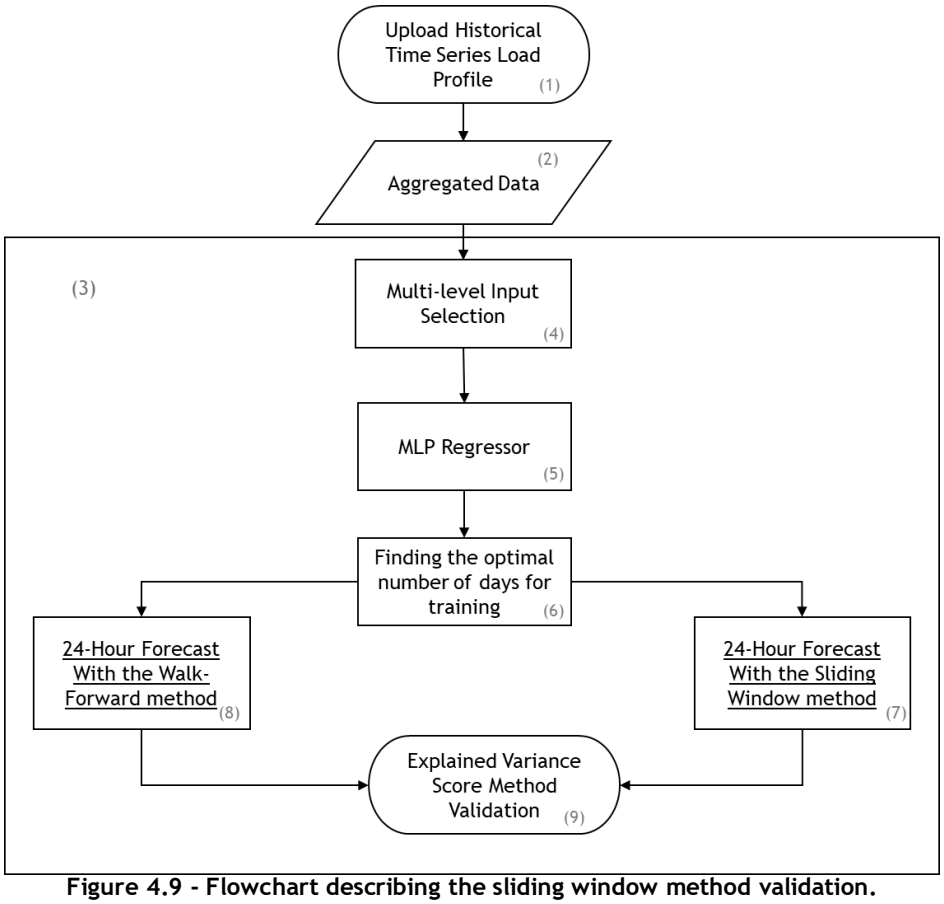

Figure 4.9 - Flowchart describing the sliding window method validation.

To find the best possible set of training days to implement the sliding windows, it was created an iterative process which decreases the amount of training days, starting with the maximum amount of data available. The testing day, 30th of December, remains unchanged throughout this process, in order to rightfully validate the best 'goodness of fit' for each dimension of the training set. Afterwards, the Explained Variance Score (EVS) resulting from each different set of training is used as a comparing metric to decide amongst the different training sets. Moreover, this process will always be conducted for the last available day of the dataset, the one who is intended to forecast. The process corresponds to step (6) and is illustrated next in Figure 4.10:

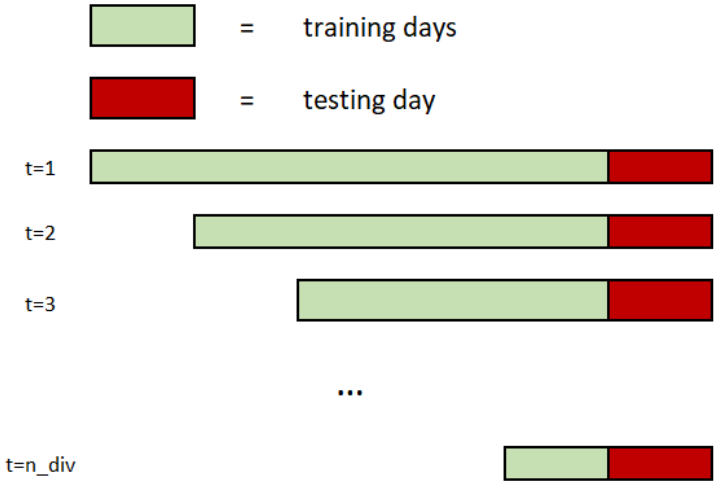

Figure 4.10 - Illustrative demonstration on the strategy implemented to assess the optimal number of days for training

The next phase, step (8) consisted in accessing the results of the EVS on different testing days throughout the year, considering a real applicable solution for a control and resource management platform of a MG. For this, the elaborated routine consisted in sequentially adding more days of training to the model, testing on different days throughout, which can be seen in Figure 4.11.

This process is called the walk-forward method and it simulates the working behavior of a MG forecasting system, making it both realistic and allowing it to make use of the best available data. For the case being, the model is required to make a one day prediction, then the actual data for that day is made available to the model so that it can be used an additional training day in order to make a new prediction on the subsequent testing day.

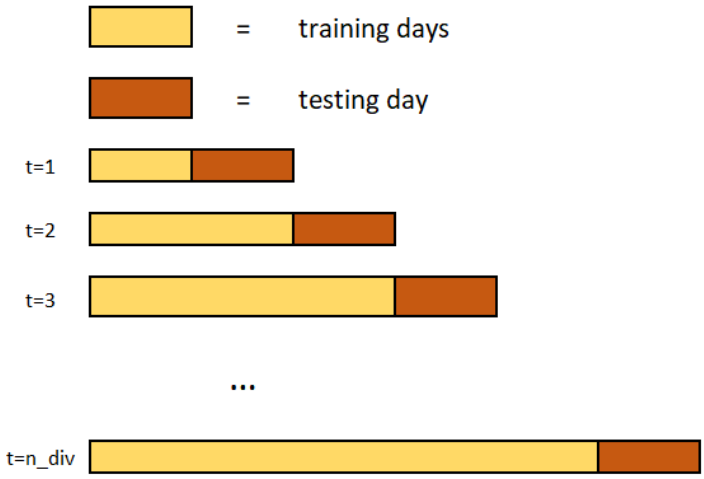

Figure 4.11 - Walk-forward model illustration.

The comparison is made from the walk-forward mode of operation applied to the MG case, against the one which implements the sliding window in step (7), when reaching the previously defined number of days required for training, illustrated in Figure 4.12. The sliding window method allows the system to always train with an optimal amount of days, dispensing unnecessary additional data and therefore alleviating the system of memory storage and

enabling lower computational processing times. The results are compared using the Explained Variance Score to validate the method in step (9).

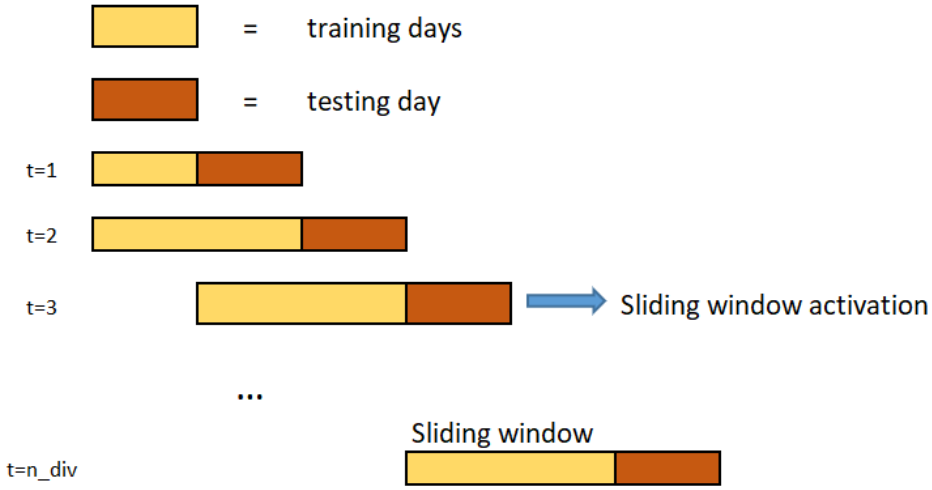

Figure 4.12 - Sliding window model illustration

Finally, step (3) represents the repeated process of updating the amount of days composing the sliding window, in which the comparison of the method application is always made from within the same testing days. The results are presented in Chapter 5.

4.2 - Final Remarks

In this chapter the HALOFMI (Hybrid Approach to Load Forecast in Micro Grids) multi-stage methodology is developed. The implemented architecture initializes with the created inputs, which consist on lagged variables of the original data. These features are then selected according to the characteristics of the target forecast day, with a calculated amount of days for training.

The number of days allocated for training is attained through a sliding window routine, which reduces dimensionality and selects the data from within a growing amount of historic load profiles, until it deletes old and irrelevant data. It was concluded that the optimal number was around the length of each season of the year. Further results, illustrations and conclusions will be held in Chapter 5.

In addition, the methodology's ability to adapt to variability of data in terms of size and nature are an important predicate, given the uncertainty associated to an increasing penetration of RES and the MG's electric demand.

Chapter 5

Method Validation and Final Results Assessment

This chapter aims to validate the proposed method and consequently assess the results obtained from it. Chapter 5 is divided in two sub-chapters. In the first one, it is validated the feature selection process, afterwards the sliding window application and finally the choice of the adopted estimator. The second sub-chapter aims at illustrating the 24-hour load forecast on different days, and the final results of the method. As an important note, the load values, originally in kWh and MWh for the case study #1 and #2 respectively, were normalized to be between 0 and 1.

5.1 - HALOFMI Method Validation using Case Study #2

Following the presentation of the multi-stage methodology in Chapter 4, sub-chapter seeks to validate the developed HALOFMI method. The methodology is assessed and validated only for the case study #2, a touristic island who presents granularity, uncertainty and a high amount of difficult days to forecast. The data refers to the year of 2016 with 15-minute time steps.

Regarding the RFECV validation, it is made a comparison prior and before of its application. For the sliding window, the objective aims at alleviating the system of unnecessary memory allocation capacity and associated costs, while maintaining the performance of the forecast. Also, it takes into consideration the possibility of adding new sources of electrical load demand to the grid, such as new offices, buildings or even the addition of electrical vehicle charging solutions. The results are compared with the walk-forward method. Finally, an estimator comparison is also carried out, testing the capability of the MLP, the AdaBooster (a decision tree improvement algorithm) and the classic persistence method.

5.1.1 - Recursive Feature Elimination with Cross-Validation (RFECV)

All the metrics which were not previously stated are now presented, where the Root Mean Squared Logarithmic Error (RMSLE) is defined in Equation 5.1. This function calculates a risk metric corresponding to the expected value of the squared logarithmic quadratic error loss.

The Median Absolute Error (MedAE) is defined in Equation 5.2, where the loss is computed by gathering the median of all absolute differences between real values and the forecasted ones. Also, it has the ability to be robust to outliers. Finally, the metric that captures the worst-case residual error between the predicted value and the true value (MaxError) is shown in Equation 5.3:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\left(\log_e(1+y_i) - \log_e(1+\hat{y}_i)\right)^2} \tag{5.1}$$

In which $\log_e(x)$ means the natural logarithm of x. Although this metric is best used in cases of data having an exponential growth, it can be successfully applied to time series which contain seasonal trends.

$$MedAE = median(|y_1 - \hat{y}_1|, \dots \dots \dots, |y_n - \hat{y}_n|) \tag{5.2}$$

$$Max\ Error = \max(|y_i - \hat{y}_i|) \tag{5.3}$$

In order to validate the application of a recursive feature elimination process, three distinct days were tested, where the only varying parameter was the application of this wrapper method. The performance metrics used to assess the results were the RMAE, the RMSE, the MedAE and the MaxError. Figure 5.1 and Figure 5.2 represent a good day of forecast while in Figure 5.3 the model struggled to capture the variability of the data. The 3 cases highlight the improvements of the RFE application.

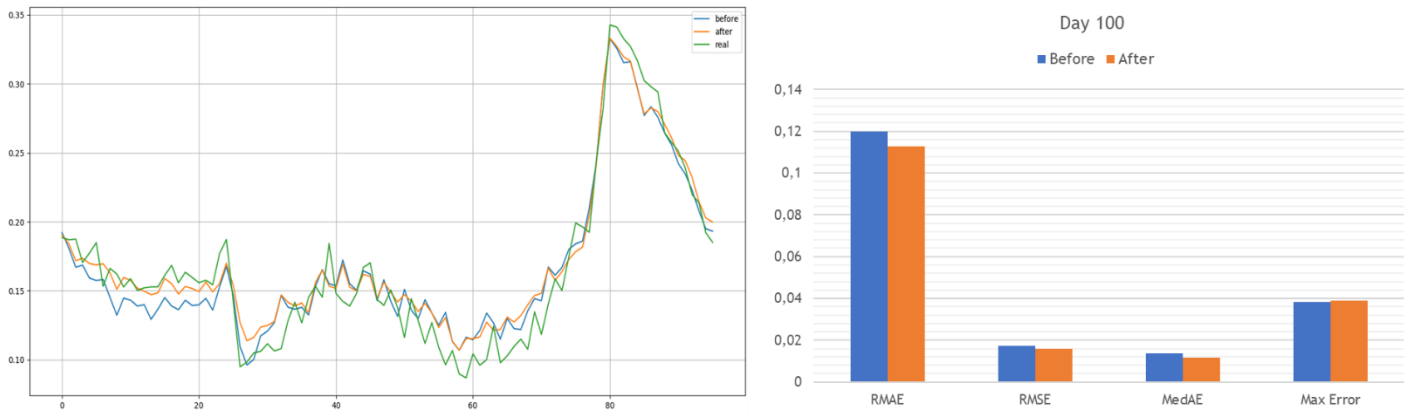

Figure 5.1 - Results on the application of RFE for the day 100 of the dataset.

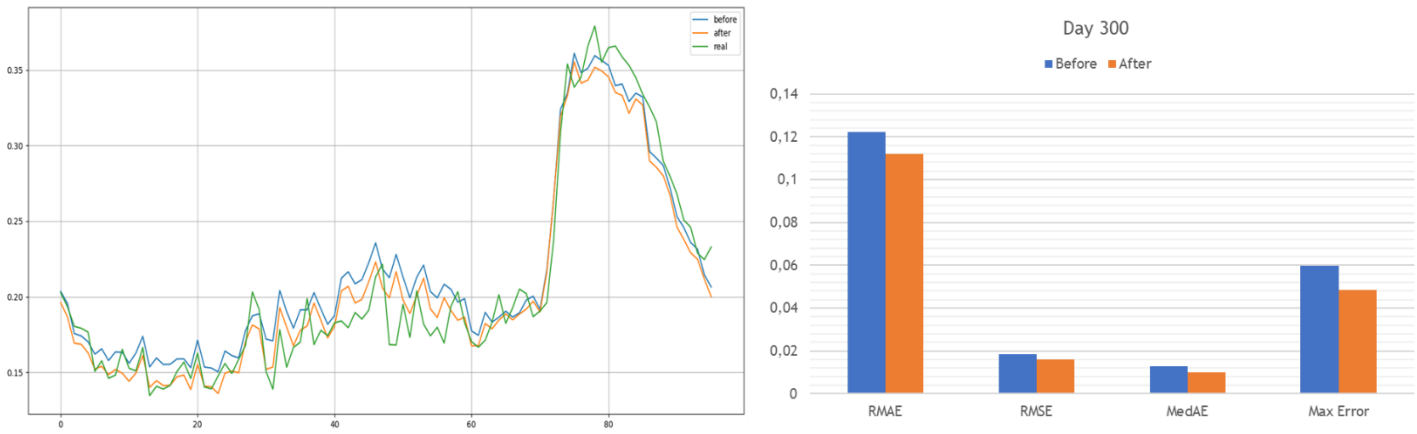

Figure 5.2 - Results on the application of RFE for the day 300 of the dataset.

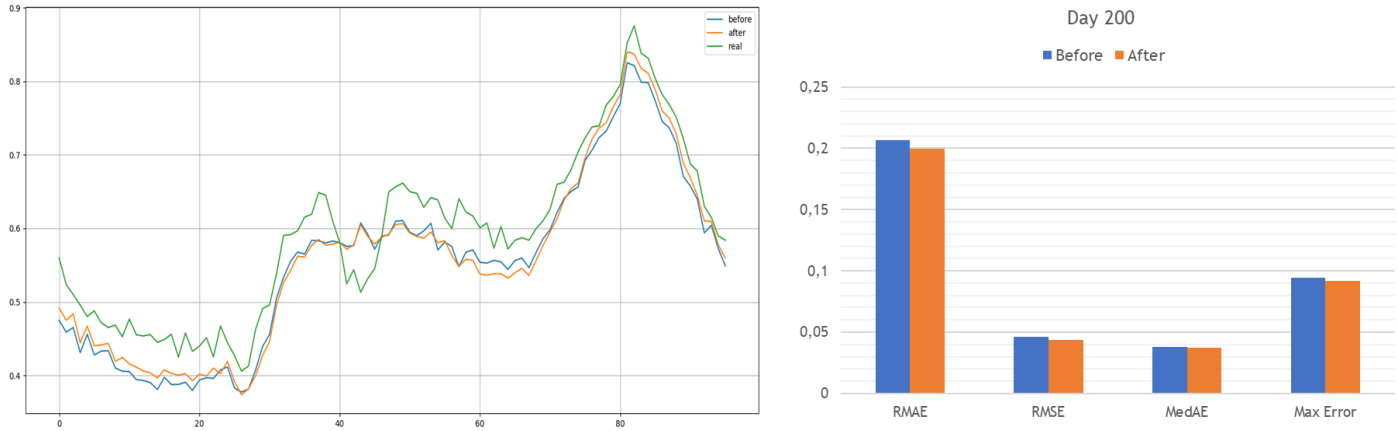

Figure 5.3 - Results on the application of RFE for the day 200 of the dataset.

The time stamps represent every 15-minute mark of the day, counting for a total of 96 (24 hours). Moreover, the application of a feature elimination based on the fit score between the input and the output leads to, in the worst-case scenario (selecting the entirety of features), to the same effect as not applying any feature selection method. Relying on an automated intelligent model who optimizes the choice of the model's attributes showed to be an important addition to the overall strategy.

5.1.2 - Sliding window vs Walking-Forward Validation

This study is performed considering the perspective of the MG's management platform, creating a functional and reliable forecasting tool, i.e. which initiates its functions from having access to barely any information to having to select the most appropriate amount. Furthermore, it is also created the opportunity for a *trade-off* scenario, where the accuracy can be sacrificed in favor of a lighter and faster forecasting mechanism.

The creation of a proof case which increases data every 8 days bears the purpose of proving the consistency of the adopted method, in a way that the testing day is always 8 days ahead of the previous one, assuring that it is also the day-of-the-week ahead. For this reason, the testing days are always multiple of 8, as the remainders are used for training. For this reason, 43 partitions of data were created resulting for each one in 43 test days. The results show not only that this method is successfully applicable for different days of the week, but also allow to explore and to prove the seasonality of the data.

The Figure 5.4 illustrated the process of extracting the appropriate number of training days for the model. Each x-axis tick corresponds to a day that was attributed for testing, out of all the available days (350 days since it was deleted two weeks' worth of data). The blue line corresponds to the Explained Variance Regression Score obtained for a decreasing number of training days, tested in the same day.

Once again, it is important to note that the EVS measures the variance fluctuation in the target data that is explained by the input data. The choice of using the EVS as a metric to validate the method was made considering that it offers a qualitative insight on how well the variance of the targeted data is being explained by the inputs that are fed to the model. For the entirety of this work, the EVS presented is in fact (1-EVS), thus the lower the result, the better. Best possible score is 0, higher values are worse. Furthermore, the red dotted line is a polynomial function of order 5 designed to fit the consecutive EVS, creating a visible trend for the adequate dimension of data for training.

In this process, the testing day remains the same, corresponding to the day 336 of the year 2016. The first 24-hour forecast is then executed with 335, followed by 327 and so forth, until training with only 7 days. These results varied depending on the chosen testing day, but the trend showed to consistently determine a minimum point for around 140 to 110 days of training, corresponding to around 4 months of data.

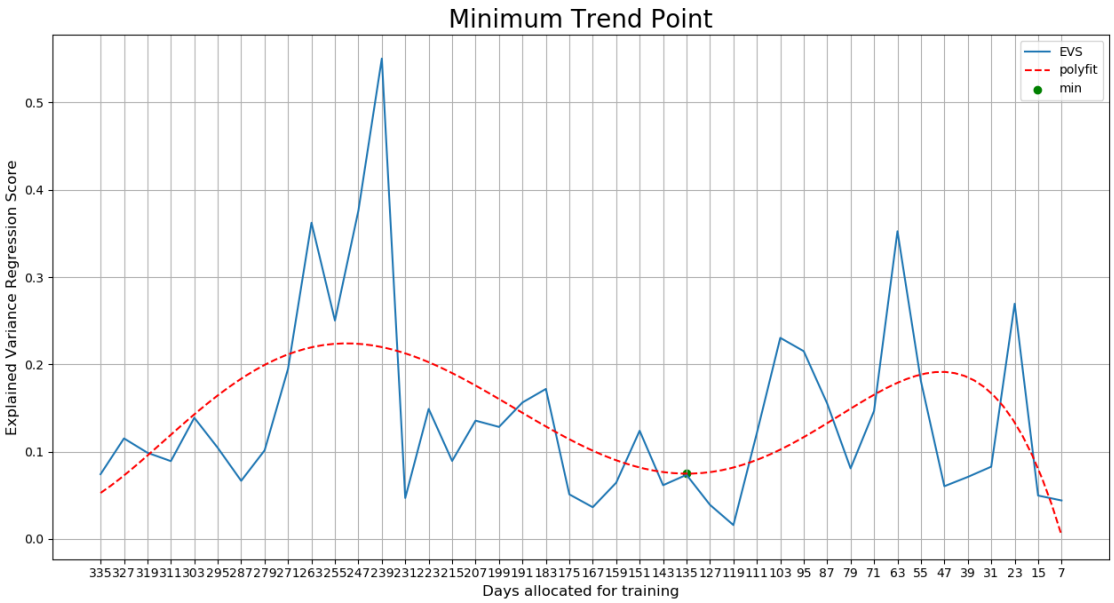

Figure 5.4 - Adopted method to discover the number of training days.

Therefore, the adopted ideal size of the 'sliding window' was 119 days, which rounds about 4 months, corresponding to the different seasons of the year. This translates and confirms expected results, from which it is possible to find the most relevant information for training in the 4 months prior to the testing day. Additionally, it can also be concluded that the methodology adopted could have been segmented into 4 different models, regarding the seasons of the year, where the load profile follows similar patterns and behaviors. The biggest advantage coming from this strategy, similar to the sliding window method, would be to

maintain the quality of the forecast with less weight of data, where it is selected the model (Winter, Spring, Autumn and Summer) according to the desired forecast day of the year.

For the next step, the results from the system operating via a walk-forward process were compared from the ones obtained when applying a sliding window to the data. This routine was implemented in a way that the testing days for both processes would match, in order to properly validate the results. The results are illustrated in Figure 5.5.

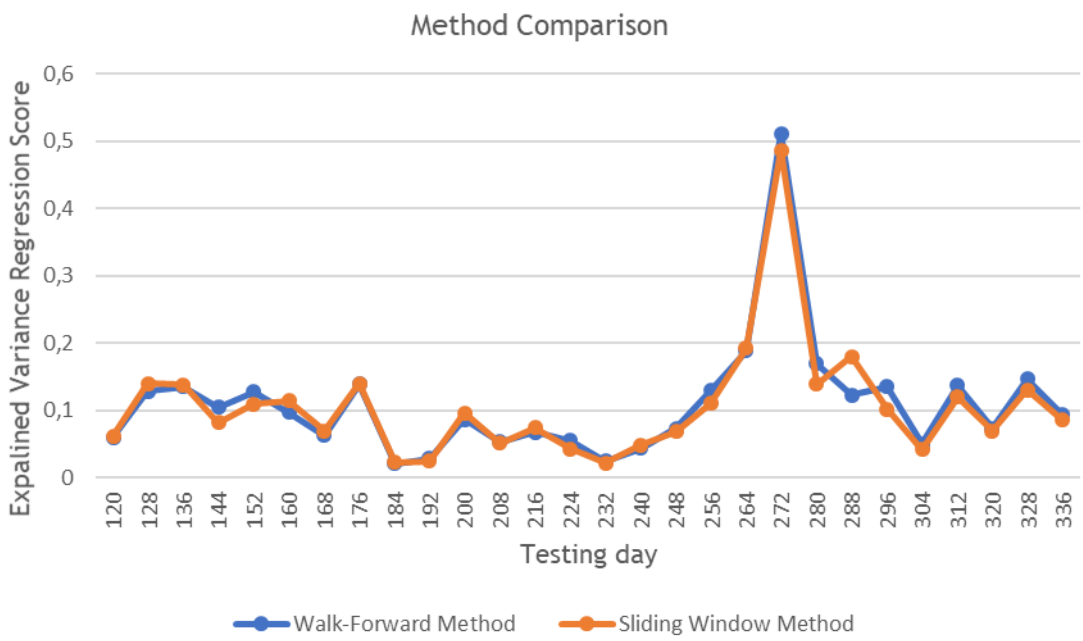

Figure 5.5 - Explained Variance Regression Score method comparison.

The results are compared from the beginning of the sliding window application up until the entirety of the available data. It is possible to observe that the results barely change considering the scenario of a growing amount of training days against a fixed 4 month-period for training. Additionally, in the 28 tested days prior to the application of a sliding window, Table 5.1 shows that for 16 of these days, the results actually improved, altough not signficantly enough to conclude that it brings an overall EVS improvement.

Table 5.1 - EVS results and improvement assessment.

	Testing Days
Improved	16
Worsened	12
Improve %	0,571428571

However, regarding computational processing time, with the sliding window application, the method is completed more than 2 minutes faster, as shown in Table 5.2. The average of the two method's EVS throughout all the testing days doesn't present any considerable changes.

Table 5.2 - EVS averages and computational time comparison.

	EVS Average	Computational time (s)
Walk-Forward	0,108992083	502,45
Sliding Window	0,105516096	371,23
Difference	0,003475988	131,22

Furthermore, it is important to consider the continuously evolving scenario of the MG. These considerations regard the flexible assets inherent to the MG, such as the total number of clients, the number of active EV and the characteristics of the load environment (residential, industrial or touristic sites) who can notably influence and alter the electric demand, especially within consecutive years. Additionally, this means that load history from one year before testing day, can contain almost redundant information, which reinforces using some type of sliding window method, or data dimensionality selection and reduction, as adopted for this dissertation.

In conclusion, the system dispenses additional data, not requiring no more than the amount that the sliding window offers. This means that a 120 amount of days can successfully maintain the fluctuation of explained variance compared to a full year of training data.

5.1.3 - Estimator's comparison

In this section, it is intended to show and compare the performance between three distinct estimators: the MLP, used for this dissertation, the AdaBoost regression and the classic persistence method.

The AdaBoost Regression belongs to the *booster* family, as the previous Gradient Boosting Regression. They both act on improving other algorithms by modifying the data at each iteration through application of weights to each of the training samples. Its essential principal is to fit a sequence of weak learners, i.e. decision trees, on repeatedly modified versions of the data. The combination of all the predictions is obtained by the sum of the weighted majority [92].

On the other hand, persistence technique is used in this work as a trivial reference to compare the skill of the MLP as a predictive model [93]. The persistence method works well when the predictive variable changes very little. It assumes that future values in the time series are determined under the assumption that the conditions remain unchanged between the present time t and the future $t + T_k$ [94].

To prove the overall consistency of the MLP, the last 3 days of the year were held for training, one at a time. On top of that, it was created a routine that splits the training and testing days in a random way but with a specific segmentation. For this, three cases were tested with the train/test split being 0.9, 0.8 and 0.7 of the data days being target for training and the remaining for testing. The performance errors were aggregated, and the average results are presented and illustrated in Figure 5.6. The improvements are clear across all the tested metrics, being the maximum error the most significate one.

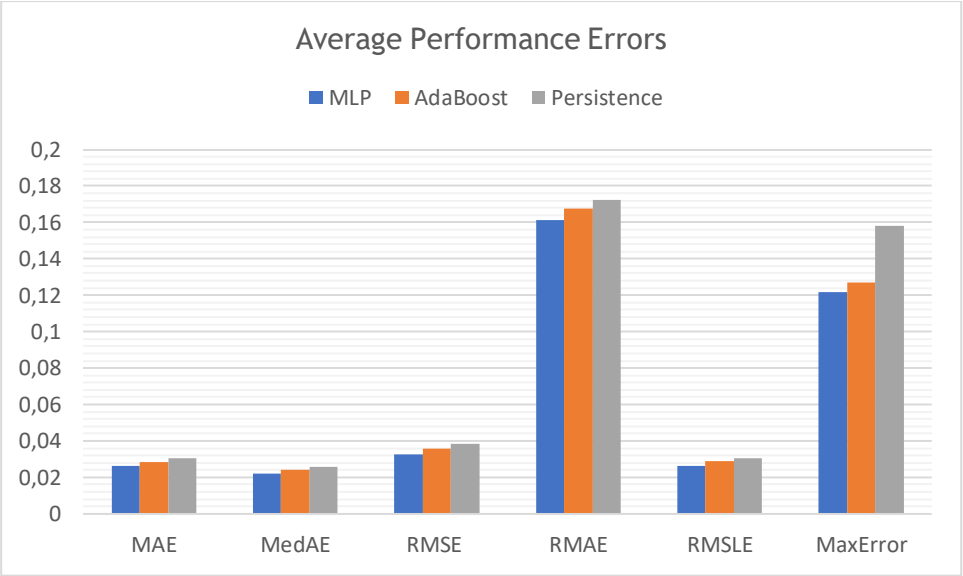

Figure 5.6 - Illustration of the average errors of 6 different tests.

5.2 - HALOFMI Results Analysis and Discussion

This section intends to show the final results coming from the developed strategy. At first, it was created a routine that aggregates the 96 instances of data into a day, iteratively repeating this process for the remainder of instances. Once the vector of days is created, it randomly chooses the testing days, based on the value of train/test segmentation that is provided. The following section concerns the results from a 24-hour forecast curve, testing the last element regarding the vector of days.

5.2.1 - Random Days Train-Test Split

The creation of a routine which randomly splits the data into a defined training/testing ratio allows the model to forecast several days in a row. This section illustrated and reinforces the necessity of creating the case study #2, since the results obtained for the case study #1 were so good, that they could not highlight the positive impact that the method's adjustment tools brings to the model. For both tests, the split was set to 0.99, which means 0.99 * 350 = 346.5 rounding to 347 days for training, leaving 3 days for testing. Case study #1 is represented in Figure 5.7 and case study #2 in Figure 5.8.

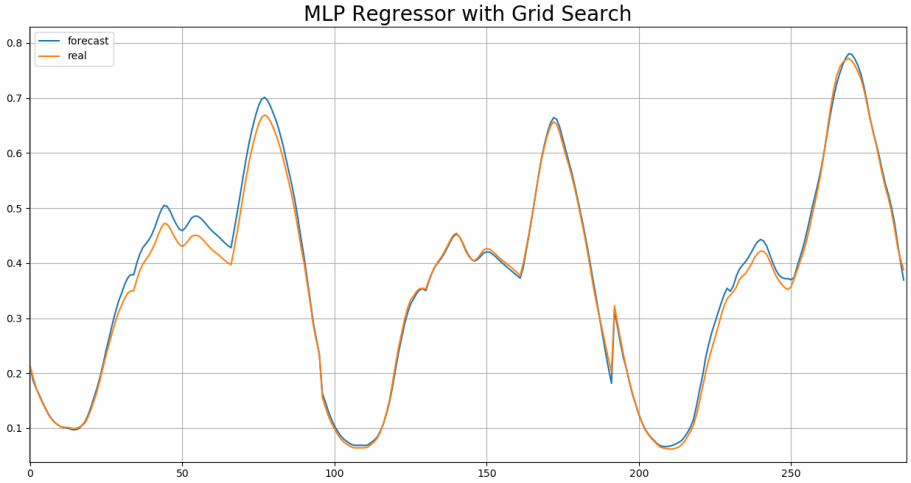

Figure 5.7 - Random Split = 0.99 for Case Study #1.

The Figure 5.7 clearly shows how smooth the load profile curve is, and how easy it is for the algorithm to understand its daily seasonality, as well as the weekly tendencies. The cyclic pattern of weekdays/weekends is well understood by the model, which clearly adapts and forecasts with a coefficient of determination (R^2) reaching 99.23% and an EVS of 99.48%. These results were considered to have little to no room for improvement, considering the lack of unexpected spikes and the well-defined load profile. This reinforces the need to study a more unpredictable electric load demand site, hence the creation of the case study #2, whose load profile characteristics were previously established.

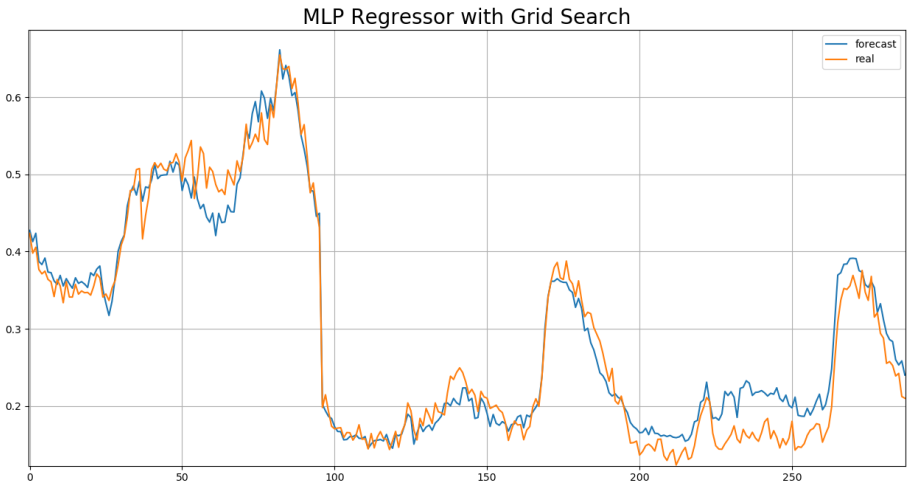

Figure 5.8 - Random Split = 0.99 for Case Study #2.

For this case, the algorithm selected one day with an average/high electric load demand and two days with a low one, corresponding to winter months. The coefficient of determination (R^2) was of 93%, same as the EVS. Results were considered to be satisfactory but show room for

further improvements. Again, this case study represents a touristic site, and showed no supportive differentiation regarding weekdays/weekends.

5.2.2 - 24-Hour Load Forecast

This section illustrated the final 24-hour predictions of a day who highlighted high percentage errors against a day who showed to be more 'standard'. They are both represented in case study #2. Figure 5.9 shows the forecast for the last day of the year, typically a very hard day to predict, with an R^2 of 83%.

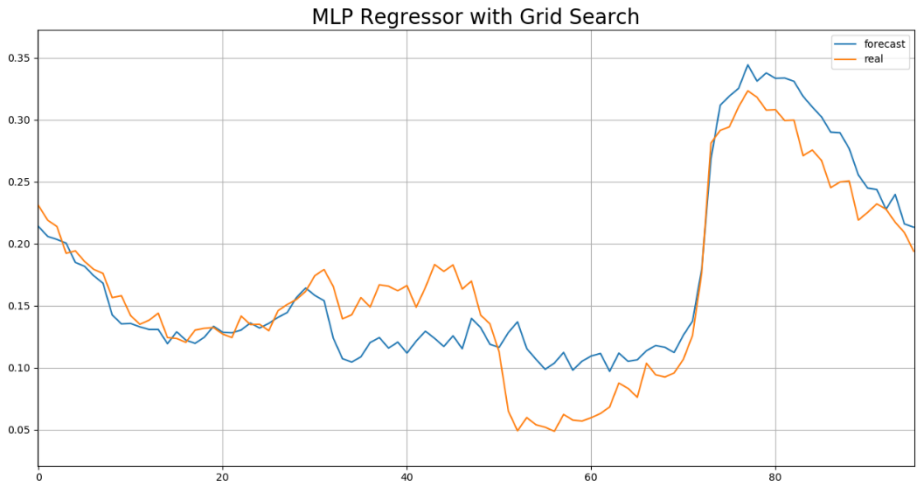

Figure 5.9 - Forecast of the 31st of December 2016.

On the other hand, on the 21st of December 2016 is a day in which the model was able to successfully follow the trend of the load curve, apart from an abrupt descent occurring shortly after the hour 05:00, although missing shortly the right magnitude values of the daily spikes. The Figure 5.10 visually illustrates this day, which returned an R^2 of 94%.

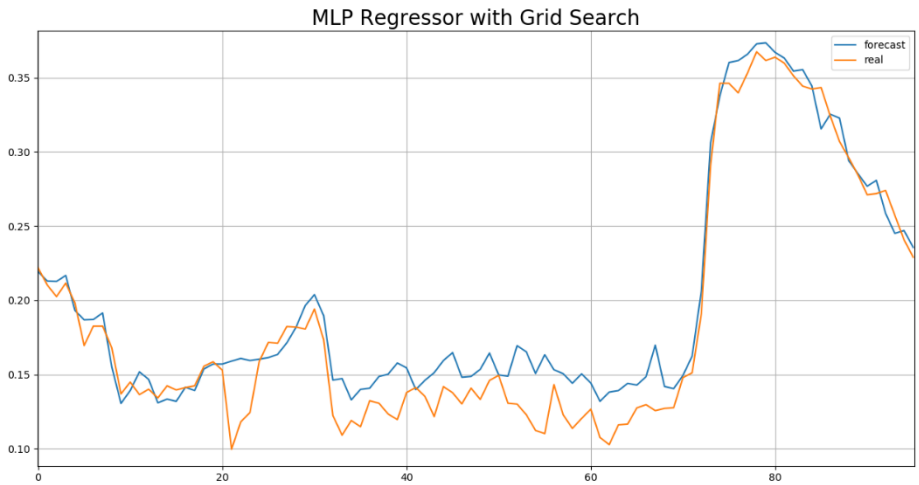

Figure 5.10 - Forecast of the 21st of December 2016.

5.3 - Final Remarks

In Chapter 5 it was showcased the enhancements coming from the performance enablers that were adopted for the 24-hour forecast model developed for this dissertation. In a first instance, highlighted the positive impact of creating an intelligent feature selection algorithm. Secondly, assessed the viability of reducing the dimensionality of data without compromising the EVS results. Finally, confirmed the MLP's expected superior capability of generalization when applied to the case study #2. It is important to note that this not imply that the adopted ANN is the best overall estimator, as the AdaBoost regression also returned good results. Once again, it all depends on the nature of the data and other potential impacting factors.

Chapter 6

Conclusions

6.1 - Essential remarks considering dimensionality adjustments

There were several considerations made regarding the dimension of the data: the first instance addresses the issue in terms of a simulation standpoint. The second instance focus more on validating the process/methodology adopted and studying the results obtained. The third instance is about creating a solution that is suitable for a MG application for real time demand solutions.

1. It was established a precedent and further demonstrated, the influence of the nature of the data set on the forecast errors obtained. Under the scope of this dissertation, it was intended to recreate the most realistic MG scenario, with all its inherent characteristics, such as variability of the load profile and an intermittent renewable production. These considerations lead to choosing a dataset that best represents and illustrates a MG community, with case study #2 bearing these characteristics.

2. To better study the dimensionality influence, it was implemented a routine that determines the optimal operational point for the MG load forecast. These considerations concern the required quality (error minimization), speed (time to compute the entire process) and allocated computational weight of the forecast. This operational point is illustrated in a trend that shows the forecast error associated to a decrease in dimensionality of data, starting from a full year up until training a few days prior to the testing day.

3. The next step of the process was aimed towards real applicable scenarios. In this instance, it is studied the error evolution considering a growing amount of data available, similarly to a MG management solution, where the initial amount of available data is very limited in the first days but cumulates load history profiles throughout time. The MG is simulated to work in two distinct ways of managing the data: the walk-forward method and the sliding window method. With the results obtained from this comparison from different testing days, it is expected an

autonomous error mitigation process by an acquired capability of adapting to the available data dimensions. A 'trade-off' analysis between error minimization and allocated data (memory storage and processing time), could take place and produce interesting decisions in terms of operational planning. The system can choose between the two operating modes.

6.2 - Conclusions and Possible Future Work

The constructed methodology for this dissertation was developed with the scope of forecasting for a time horizon of 24-hours, attributed as Long-Term Load Forecast. Additionally, forecasting for a 3-hour ahead time was an early objective, and would imply the design, creation and a new approach to the problem, hence developing a whole new methodology. The strategy for this case could include adding more features considering not only the load data from previous days at the same hour, but also from hours different than the hour intended to forecast, in a way that *h-1, h-2* or *h-3* could contain useful information. This time horizon was adopted as Short-Term Load Forecast and will be targeted for future work. The expected results would include reduced errors for the first predicted hours, compared to the LTLF (24-hour Load Forecast).

Another possible alternative route would be to attribute numerical values in order to differentiate weekdays from weekends. That way, features used to predict days who have the same categorical values, belong to the same class of days. Such method would lead to the creation of two separate models and could produce effective results depending on if the data has a clear distinct daily load profile for working days or weekends. Holidays would be considered as Sundays. Regarding the case study #1, a valid strategy would've been to model the Saturdays, Sundays and Weekdays, resulting in three different models, in a way that the attained forecast errors could be associated to each respective class of day. From there, different measures and procedures would be studied in order to improve every one of them.

One other possible way to solve the forecasting issue in hands would be to create seasonal models, instead of yearly ones, giving the cyclic nature of the electricity load. In this case, the sliding window would be defined by the length of each seasonal period.

Furthermore, the decision of selecting two weeks of data as features was made from a sensitive guess. The features could include the most relevant days up to three weeks, or even one month of data. This would create an opportunity for 'older data' that proved to be more relevant than more recent instances, to be included as features. For example, daily load profiles of Saturdays occurring 3 or 4 weeks before the target day could get higher importance scores than the features from 2 or 3 days before.

More advanced models can be used to select the most suitable and influencing factors that better describe electricity load forecasting on a MG level, e.g. more in-depth information of the consumer, relating to an incentive based demand response model, covering the diversity of customers by provisioning the different incentive rates. Also, including the uncertainty from the distributed RES generation to the developed hybrid model for future research can have interesting results. These additional inputs can result in the construction of models with a high degree of complexity, opening a way to build deep learning models, with deep neural networks and reinforcement learning to better process the non-linear nature of the data.

Regarding the estimators used and the adopted architecture for the ANN, the MLP proved to be more consistent overall, showing a batter capability of adapting to data with a high

degree of spikes and variability. For the RFECV, the estimator used for the MG' two modes of operation regarding the data management was the LR, due to inferior computational time, in spite of the other two, the GBR and the RF, who showcased a more suitable adaptability to new data and a more diligent training force, meaning they need less features to build the model in an effective way. Additionally, to prove the benefits that come from the RFECV, it was compared for same testing days, the differences before and after its application

Finally, the strategy developed for this dissertation can provide practical short-term improvements for the MG data management operations, scheduling and maintenance of the grid network, offering an opportunity for cost savings.

References

[1] Robin John Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. OTexts, 2018.

[2] G. M. U. Din and A. K. Marnerides, "Short term power load forecasting using Deep Neural Networks," *2017 Int. Conf. Comput. Netw. Commun. ICNC 2017*, pp. 594–598, 2017.

[3] A. Gonz, A. F. Skarmeta, and S. Miˇ, "Machine Learning based Electric Load Forecasting for Short and Long-term Period," pp. 511–516, 2016.

[4] J. Wang and S. Wang, *Business intelligence in economic forecasting : technologies and techniques*. Information Science Reference, 2010.

[5] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecast.*, 2016.

[6] T. Hong, "HSC Research Report Past , present and future."

[7] B. Islam, "Comparison of Conventional and Modern Load Forecasting Techniques and Expert Systems," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, pp. 504–513, 2011.

[8] H. Marzooghi, K. Emami, P. J. Wolfs, and B. Holcombe, "Short-term Electric Load Forecasting in Microgrids: Issues and Challenges," 2019, pp. 1–6.

[9] M. A. Mat Daut, M. Y. Hassan, H. Abdullah, H. A. Rahman, M. P. Abdullah, and F. Hussin, "Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review," *Renewable and Sustainable Energy Reviews*, vol. 70. Elsevier Ltd, pp. 1108–1118, 2017.

[10] X. Zhu and M. Shen, "Based on the ARIMA model with grey theory for short term load forecasting model," in *2012 International Conference on Systems and Informatics, ICSAI 2012*, 2012, pp. 564–567.

[11] M. A. Al Amin and M. A. Hoque, "Comparison of ARIMA and SVM for short-term load forecasting," in *IEMECON 2019 - 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference*, 2019, pp. 205–210.

[12] V. Radziukynas and A. Klementavicius, "Short-term wind speed forecasting with ARIMA model," in *2014 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University, RTUCON 2014*, 2014, pp. 145–149.

[13] M. Akpinar and N. Yumusak, "Forecasting household natural gas consumption with ARIMA model: A case study of removing cycle," in *AICT 2013 - 7th International Conference on Application of Information and Communication Technologies, Conference Proceedings*, 2013.

[14] N. Elamin and M. Fukushige, "Modeling and forecasting hourly electricity demand by SARIMAX with interactions," *Energy*, pp. 257–268, Dec. 2018.

[15] Ö. Ö. Bozkurt, G. Biricik, and Z. C. Tayşi, "Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market," *PLoS One*, vol. 12, no. 4, p. e0175915, Apr. 2017.

[16] W. Li and Z. G. Zhang, "Based on time sequence of ARIMA model in the application of short-term electricity load forecasting," in *ICRCCS 2009 - 2009 International Conference on Research Challenges in Computer Science*, 2009, pp. 11–14.

[17] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning forecasting methods: Concerns and ways forward," *PLoS One*, vol. 13, no. 3, Mar. 2018.

[18] N. Hodžić, T. Konjić, and V. Miranda, "Artificial neural networks applied to short term

load diagram prediction," in *8th Seminar on Neural Network Applications in Electrical Engineering, Neurel-2006 Proceedings*, 2006, pp. 219–223.

[19] D. Scott, T. Simpson, N. Dervilis, T. Rogers, and K. Worden, "Machine Learning for Energy Load Forecasting," in *Journal of Physics: Conference Series*, 2018, vol. 1106, no. 1.

[20] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting," in *2017 6th International Youth Conference on Energy, IYCE 2017*, 2017.

[21] Q. W. Luthuli and K. A. Folly, "Short term load forecasting using artificial intelligence," in *IEEE PES PowerAfrica Conference, PowerAfrica 2016*, 2016, pp. 129–133.

[22] L. Limei and H. Xuan, "Study of electricity load forecasting based on multiple kernels learning and weighted support vector regression machine," in *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, 2017, pp. 1421–1424.

[23] G. Bontempi, S. Ben Taieb, and Y. A. Le Borgne, "Machine learning strategies for time series forecasting," in *Lecture Notes in Business Information Processing*, 2013, vol. 138 LNBIP, pp. 62–77.

[24] Z. C. Sun *et al.*, "Study on load forecasting of an actual micro-grid system in Taiwan," in *IEEE International Symposium on Industrial Electronics*, 2013.

[25] J. Chan Phooi M'ng and M. Mehralizadeh, "Forecasting East Asian Indices Futures via a Novel Hybrid of Wavelet-PCA Denoising and Artificial Neural Network Models," *PLoS One*, vol. 11, no. 6, p. e0156338, Jun. 2016.

[26] C. Deb, L. S. Eang, J. Yang, and M. Santamouris, "Forecasting diurnal cooling energy load for institutional buildings using Artificial Neural Networks," *Energy Build.*, vol. 121, pp. 284–297, Jun. 2016.

[27] M. Puig-Arnavat and C. B. Joan, "Artifical Neural Networks for Termochemical Conversion of Biomass," 2015, pp. 133–156.

[28] F. Rosenblatt, "ARMED SERVICES TECHNICAL INFORMATM AGENCY ARLINGTON HALL STATION ARLINGTON 12, VIRGINIA."

[29] D. E. Ruineihart, G. E. Hint, and R. J. Williams, "LEARNING INTERNAL REPRESENTATIONS BERROR PROPAGATION two," 1985.

[30] N. J. D. Nagelkerke, "A note on a general definition of the coefficient of determination," *Biometrika*, vol. 78, no. 3. pp. 691–692, Sep-1991.

[31] L. Hernandez, C. Baladrón, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, and J. Lloret, "Short-term load forecasting for microgrids based on artificial neural networks," *Energies*, vol. 6, no. 3, pp. 1385–1408, 2013.

[32] F. Rodrigues, C. Cardeira, and J. M. F. Calado, "The daily and hourly energy consumption and load forecasting using artificial neural network method: A case study using a set of 93 households in Portugal," in *Energy Procedia*, 2014, vol. 62, pp. 220–229.

[33] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Q. Appl. Math.*, vol. 2, no. 2, pp. 164–168, Jul. 1944.

[34] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.

[35] P. Zeng, H. Li, H. He, and S. Li, "Dynamic Energy Management of a Microgrid Using Approximate Dynamic Programming and Deep Recurrent Neural Network Learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4435–4445, Jul. 2019.

[36] N. Cruz, L. G. Marin, and D. Saez, "Prediction Intervals With LSTM Networks Trained By Joint Supervision," 2019, pp. 1–8.

[37] F. Maria Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, "An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting."

[38] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Networks*, vol. 116, pp. 237–245, Aug. 2019.

[39] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, pp. 1078–1088, Feb. 2019.

[40] D. L. Vu, "A combination model based on a neural network autoregression and

Bayesian network to forecast for avoiding brown plant hopper," in *International Conference on Advanced Technologies for Communications*, 2016, vol. 2016-January, pp. 220–225.

[41] D. Kim, S. W. Hwang, and J. Kim, "Very Short-Term Photovoltaic Power Generation Forecasting with Convolutional Neural Networks," in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018, pp. 1310–1312.

[42] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network," *Energies*, vol. 11, no. 12, Dec. 2018.

[43] A. Elvers, M. Vos, and S. Albayrak, "Short-Term Probabilistic Load Forecasting at Low Aggregation Levels Using Convolutional Neural Networks," 2019, pp. 1–6.

[44] M. Vos, C. Bender-Saebelkampf, and S. Albayrak, "Residential Short-Term Load Forecasting Using Convolutional Neural Networks," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018*, 2018.

[45] I. Koprinska, D. Wu, and Z. Wang, "Convolutional Neural Networks for Energy Time Series Forecasting," in *Proceedings of the International Joint Conference on Neural Networks*, 2018, vol. 2018-July.

[46] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional Time Series Forecasting with Convolutional Neural Networks," Mar. 2017.

[47] M. Bińkowski, G. Marti, and P. Donnat, "Autoregressive Convolutional Neural Networks for Asynchronous Time Series," Mar. 2017.

[48] Y. J. Chang, S. H. Wang, and H. Y. Sun, "Research of short-term load forecasting algorithm based on wavelet analysis and Radial Basis Function Neural Network," in *PEITS 2009 - 2009 2nd Conference on Power Electronics and Intelligent Transportation System*, 2009, vol. 1, pp. 81–84.

[49] L. Zhuang, H. Liu, J. Zhu, S. Wang, and Y. Song, "Comparison of forecasting methods for power system short-Term load forecasting based on neural networks," in *2016 IEEE International Conference on Information and Automation, IEEE ICIA 2016*, 2017, pp. 114–119.

[50] H. J. Lu and G. W. Chang, "Wind Power Forecast by Using Improved Radial Basis Function Neural Network," in *IEEE Power and Energy Society General Meeting*, 2018, vol. 2018-August.

[51] W. Wu, S. Zhong, and G. Zhou, "A study on PID intelligent optimization based on radial basis function neural networks," in *2013 3rd International Conference on Consumer Electronics, Communications and Networks, CECNet 2013 - Proceedings*, 2013, pp. 57–60.

[52] D. Niu, L. Ji, and J. Tian, "Wavelet neural network embedded expert system used in short-term load forecasting," in *ICEMMS 2011 - Proceedings: 2011 2nd IEEE International Conference on Emergency Management and Management Sciences*, 2011, pp. 190–193.

[53] H. Chitsaz, H. Shaker, H. Zareipour, D. Wood, and N. Amjady, "Short-term electricity load forecasting of buildings in microgrids," *Energy Build.*, vol. 99, pp. 50–60, Apr. 2015.

[54] T. Liu, "Research on the electric load forecasting and risk assessment based on wavelet neural network," in *3rd International Symposium on Intelligent Information Technology Application, IITA 2009*, 2009, vol. 3, pp. 568–571.

[55] Q. Zhang and T. Liu, "Research on Mid-long Term Load Forecasting Base on Wavelet Neural Network," in *2010 Second International Conference on Computer Engineering and Applications*, 2010, pp. 217–220.

[56] T. Liu, "Research on the short-term electric load forecasting based on wavelet neural network," in *2009 International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2009*, 2009, vol. 3, pp. 20–23.

[57] N. Liu and H. Wang, "Ensemble based extreme learning machine," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 754–757, 2010.

[58] G.-B. Huang, S. Member, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *CYBERNETICS*, vol. 42, no. 2, p. 513,

2012.
[59] G.-B. Huang, “What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt’s Dream and John von Neumann’s Puzzle,” *Cognit. Comput.*, vol. 7.
[60] G.-B. Huang, “An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels.”
[61] M. Braun, T. Bernard, O. Piller, and F. Sedehizade, “24-hours demand forecasting based on SARIMA and support vector machines,” in *Procedia Engineering*, 2014, vol. 89, pp. 926–933.
[62] “Support Vector Machine Regression.” [Online]. Available: http://kernelsvm.tripod.com/. [Accessed: 28-Nov-2019].
[63] V. Vapnik and S. E. Golowich, “Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing·.”
[64] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3. pp. 199–222, Aug-2004.
[65] H. Jiang, Y. Zhang, E. Muljadi, J. J. Zhang, and D. W. Gao, “A Short-Term and High-Resolution Distribution System Load Forecasting Approach Using Support Vector Regression with Hybrid Parameters Optimization,” *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3331–3350, Jul. 2018.
[66] N. Amjady, “Short-term bus load forecasting of power systems by a new hybrid method,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 333–341, Feb. 2007.
[67] N. Amjady and A. Daraeepour, “Midterm demand prediction of electrical power systems using a new hybrid forecast technique,” *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 755–765, May 2011.
[68] S. Fan and L. Chen, “Short-term load forecasting based on an adaptive hybrid method,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 392–401, Feb. 2006.
[69] A. S. Pandey, D. Singh, and S. K. Sinha, “Intelligent hybrid wavelet models for short-term load forecasting,” *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1266–1273, Aug. 2010.
[70] S. Li, P. Wang, and L. Goel, “A novel wavelet-based ensemble method for short-term load forecasting with hybrid neural networks and feature selection,” *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1788–1798, May 2016.
[71] M. Hanmandlu and B. K. Chauhan, “Load forecasting using hybrid models,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 20–29, Feb. 2011.
[72] S. Motepe, A. N. Hasan, and R. Stopforth, “Improving Load Forecasting Process for a Power Distribution Network Using Hybrid AI and Deep Learning Algorithms,” *IEEE Access*, vol. 7, pp. 82584–82598, 2019.
[73] C. Cecati, J. Kolbusz, P. Rózycki, P. Siano, and B. M. Wilamowski, “A Novel RBF Training Algorithm for Short-Term Electric Load Forecasting and Comparative Studies,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6519–6529, Oct. 2015.
[74] E. Ceperic, V. Ceperic, and A. Baric, “A strategy for short-term load forecasting by support vector regression machines,” *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, 2013.
[75] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
[76] A. T. Eseye, M. Lehtonen, T. Tukia, S. Uimonen, and R. John Millar, “Machine learning based integrated feature selection approach for improved electricity demand forecasting in decentralized energy systems,” *IEEE Access*, vol. 7, pp. 91463–91475, 2019.
[77] S. Fan and R. J. Hyndman, “Short-term load forecasting based on a semi-parametric additive model,” *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 134–141, Feb. 2012.
[78] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, “Short-Term Load Forecasting with Deep Residual Networks,” *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019.
[79] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, and Z. Zhu, “Multi-scale convolutional neural network with time-cognition for multi-step short-Term load forecasting,” *IEEE Access*, vol. 7, pp. 88058–88071, 2019.
[80] M. Daneshdoost, “Neural network with fuzzy set-based classification for short-term

load forecasting," *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1386–1391, 1998.
[81] R. J. Hyndman and S. Fan, "Density forecasting for long-term peak electricity demand," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1142–1153, May 2010.
[82] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 456–462, Jan. 2014.
[83] J. Xie, T. Hong, and J. Stroud, "Long-term retail energy forecasting with consideration of residential customer attrition," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2245–2252, Sep. 2015.
[84] M. S. Kandil, S. M. El-Debeiky, and N. E. Hasanien, "Long-term load forecasting for fast developing utility using a knowledge-based expert system," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 491–496, May 2002.
[85] H. Sangrody, N. Zhou, S. Tutun, B. Khorramdel, M. Motalleb, and M. Sarailoo, "Long term forecasting using machine learning methods," in *2018 IEEE Power and Energy Conference at Illinois, PECI 2018*, 2018, vol. 2018-Janua, pp. 1–5.
[86] G. Seni and J. F. Elder, "Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions," *Synth. Lect. Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 1–126, Jan. 2010.
[87] M. Jacob, C. Neves, and D. Vukadinović Greetham, "Short Term Load Forecasting," in *Forecasting and Assessing Risk of Individual Electricity Peaks*, Springer, Cham, 2020, pp. 15–37.
[88] "(No Title)." [Online]. Available: https://www.erse.pt/media/wk3py2zf/caracterização-procura-ee-2019-dez2018.pdf. [Accessed: 20-Feb-2020].
[89] "(No Title)." [Online]. Available: https://www.edpdistribuicao.pt/sites/edd/files/2019-03/DocMetodologico_Perfis2019_20190111v2.pdf. [Accessed: 20-Feb-2020].
[90] I. Koprinska, M. Rana, and V. G. Agelidis, "Yearly and seasonal models for electricity load forecasting," in *Proceedings of the International Joint Conference on Neural Networks*, 2011, pp. 1474–1481.
[91] A. Suppers, A. J. van Gool, and H. J. C. T. Wessels, "Integrated chemometrics and statistics to drive successful proteomics biomarker discovery," *Proteomes*, vol. 6, no. 2. MDPI AG, 27-Mar-2018.
[92] "1.11. Ensemble methods — scikit-learn 0.22.2 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html#adaboost. [Accessed: 05-Mar-2020].
[93] G. Notton and C. Voyant, "Forecasting of Intermittent Solar Energy Resource," in *Advances in Renewable Energies and Power Technologies*, vol. 1, Elsevier, 2018, pp. 77–114.
[94] C. F. M. Coimbra and H. T. C. Pedro, "Stochastic-Learning Methods," in *Solar Energy Forecasting and Resource Assessment*, Elsevier Inc., 2013, pp. 383–406.